

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**SERVIÇO DE COLABORAÇÃO PARA A
ARQUITETURA CLINICSPACE**

DISSERTAÇÃO DE MESTRADO

Marcelo Lopes Kroth

Santa Maria, RS, Brasil

2011

SERVIÇO DE COLABORAÇÃO PARA A ARQUITETURA CLINICSPACE

por

Marcelo Lopes Kroth

Dissertação apresentada ao Programa de Pós-Graduação em Informática da
Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para
a obtenção do grau de
Mestre em Computação

Orientador: Prof^a. Dr^a. Iara Augustin (UFSM)

Santa Maria, RS, Brasil

2011

K93s Kroth, Marcelo Lopes

Serviço de Colaboração para a Arquitetura ClinicSpace / por Marcelo Lopes Kroth. – 2011.

63 p.: il.; 31 cm.

Orientador: Iara Augustin (UFSM)

Dissertação (Mestrado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Informática, RS, 2011.

1. Informática 2. Computação 3. Computação Ubíqua 4. Computação Pervasiva 5. Gerenciamento de tarefas 6. *Middleware* 7. Atividades Clínicas I. Augustin (UFSM), Iara. II. Título.

CDU 004.75.057.5

Ficha catalográfica elaborada por
Simone G. Maisonave - CRB-10/1733
Biblioteca Central da UFSM

© 2011

Todos os direitos autorais reservados a Marcelo Lopes Kroth. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

E-mail: marcelo.tuco@ufsm.br

**Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Informática**

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

**SERVIÇO DE COLABORAÇÃO PARA A ARQUITETURA
CLINICSPACE**

elaborada por
Marcelo Lopes Kroth

como requisito parcial para obtenção do grau de
Mestre em Computação

COMISSÃO EXAMINADORA:

Iara Augustin (UFSM), Dr^a.
(Presidente/Orientador)

Leandro Buss Becker, Prof. Dr. (UFSC)

Deise de Brum Saccol, Prof^a. Dr^a. (UFSM)

Santa Maria, 18 de Julho de 2011.

DEDICATÓRIA

Para meu filho Pedro.

AGRADECIMENTOS

Uma das melhores partes de realizar um trabalho é chegar ao final e ter agradecimentos a fazer.

Agradeço aos colegas da UFSM, em especial ao Fernando Bordin da Rocha, Diretor do CPD, que sempre incentivou a sua equipe a se qualificar, Giuliano Ferreira pelas aulas de EXEHDA, ClinicSpace e Java, Sérgio Limberger pela atenciosa revisão do documento e Douglas Pasqualin pela consultoria no \LaTeX .

Agradeço à minha esposa Giana, que sempre está ao meu lado me ajudando a superar os desafios, e ao meu filho Pedro, minha família é o principal motivo para eu tentar, a cada dia, me tornar uma pessoa melhor.

Agradeço à Prof^a. Iara Augustin que já havia me orientado no trabalho final de graduação e, em um gesto de quase insanidade, aceitou novamente me orientar agora no mestrado, é uma honra ter a sua orientação.

Agradeço aos colegas do GMob da UFSM pela troca de ideias, muito importante em um projeto tão complexo e desafiador quanto o ClinicSpace.

Agradeço aos professores Leandro Buss Becker, Deise de Brum Saccol e Andrea Schwertner Charão (suplente) por terem aceitado o convite de fazer parte da comissão examinadora do trabalho.

Muito obrigado!

*“Se as coisas são inatingíveis... ora!
Não é motivo para não querê-las...
Que tristes os caminhos, se não fora
A presença distante das estrelas!”*

— MÁRIO QUINTANA

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Informática
Universidade Federal de Santa Maria

SERVIÇO DE COLABORAÇÃO PARA A ARQUITETURA CLINICSPACE

AUTOR: MARCELO LOPES KROTH

ORIENTADOR: IARA AUGUSTIN (UFSM)

Local da Defesa e Data: Santa Maria, 18 de Julho de 2011.

A Computação Ubíqua (*Ubiquitous Computing*) ou Pervasiva (*Pervasive Computing*) é um novo paradigma computacional que objetiva prover tecnologia de informação e comunicação em qualquer lugar, acessível por qualquer pessoa, disponível o tempo todo, onde os recursos computacionais devem estar integrados ao ambiente físico de forma transparente. Uma das áreas de pesquisa sobre infraestrutura computacional pervasiva está relacionada a ambientes clínicos, devido às características de mobilidade, interrupção e colaboração. Dentro deste contexto, o projeto ClinicSpace, em desenvolvimento no GMob / PPGI / UFSM, utiliza os conceitos de Computação Pervasiva para auxiliar os médicos na execução de suas tarefas em um ambiente hospitalar. O objetivo é permitir aos médicos a personalização da execução de suas tarefas, que são gerenciadas por uma infraestrutura computacional em um ambiente pervasivo. O trabalho clínico é altamente colaborativo devido à natureza especialista do tratamento: médicos de diferentes especialidades precisam colaborar através do tempo e espaço. Outro aspecto importante a ser observado é a característica assíncrona da comunicação entre esses profissionais, como por exemplo, em uma troca de turno. Este trabalho apresenta o Serviço de Colaboração criado para dar suporte à colaboração assíncrona entre os profissionais da área da saúde através da delegação de tarefas que ainda não foram concluídas, de forma integrada à arquitetura do projeto ClinicSpace. Atualmente, a maioria dos trabalhos que abordam os aspectos colaborativos de ambientes clínicos com suporte de uma infraestrutura computacional pervasiva tem o foco na colaboração síncrona, com vários esforços para a criação de mecanismos para auxiliar, principalmente, os diagnósticos em equipes fisicamente distribuídas. Diferente destas abordagens, este trabalho observa as características assíncronas da colaboração em ambientes hospitalares, sempre com a visão central no clínico e sua forma particular de executar a atividade médica. Foi feita uma análise de desempenho das aplicações após as modificações realizadas na arquitetura do projeto ClinicSpace e a conclusão foi que, após a introdução de um mecanismo de cache para as tarefas, as alterações na arquitetura mantiveram um desempenho na execução das aplicações no ambiente pervasivo, semelhante ao que se tinha antes das modificações.

Palavras-chave: Computação Ubíqua, Computação Pervasiva, *Middleware*, Atividades Clínicas, Colaboração.

ABSTRACT

Master's Dissertation
Program in Computer Science
Universidade Federal de Santa Maria

COLLABORATION SERVICE TO CLINICSPACE ARCHITETURE

AUTHOR: MARCELO LOPES KROTH

ADVISOR: IARA AUGUSTIN (UFSM)

Defense Place and Date: Santa Maria, July 18th, 2011.

Ubiquitous Computing or Pervasive Computing is a new paradigm that aims to provide information and communication technology accessible anywhere, by anyone, available anytime, where computational resources should be integrated into the physical environment in a transparent manner. One of the areas of research on pervasive computing infrastructure is related to hospital settings, because of the characteristics of mobility, collaboration and interruption. In this context, the ClinicSpace project, under development at GMob / PPGI / UFSM, uses the concepts of Pervasive Computing to help physicians while performing their tasks in an hospital settings. The goal is to allow doctors to customize the execution of their tasks, which are managed by a middleware in a pervasive environment. Medical work is highly collaborative because of specialist nature of treatments: physicians from different specialties need to collaborate across time and space. Another important aspect to note is characteristic of the asynchronous communication between these professionals, for example, when changing shift. This thesis presents a Collaboration Service created to support asynchronous collaboration among professionals through the delegation of tasks that are not yet complete, integrated to architecture of the ClinicSpace project. Currently, most studies that focus the collaborative aspects in hospital settings with a base of a pervasive middleware have focused on synchronous collaboration, with several efforts to create mechanisms to help, especially in diagnostics, physically distributed teams. Unlike these approaches, this thesis focuses on the characteristics of asynchronous collaboration in hospital settings, always with central vision in the doctor and the particular way of executing their medical activity. An analysis of the applications performance was made after the modifications to the architecture of ClinicSpace project and the conclusion was that after the introduction of a caching mechanism for the tasks, changes in the architecture remained a performance in the execution of applications in pervasive environment, similar than what they had before the modifications.

Keywords: Ubiquitous Computing, Pervasive Computing, Middleware, Clinical Activities, Collaboration.

LISTA DE FIGURAS

2.1	Arquitetura Aura (GARLAN et al., 2002)	22
2.2	Componentes do <i>Prism</i> (GARLAN et al., 2002)	23
2.3	Arquitetura ABC (BARDRAM, 2009)	25
3.1	Conceito de tarefas na arquitetura ClinicSpace	30
3.2	Máquina de Estados para as Tarefas (FERREIRA, 2009)	31
3.3	Arquitetura para programação e gerenciamento de tarefas (FERREIRA, 2009)	33
3.4	Subsistema de Gerenciamento Distribuído de Tarefas (FERREIRA, 2009)...	35
3.5	Interface de Edição de Tarefas e Contexto (MACHADO, 2010).....	37
4.1	Serviço de Colaboração	41
4.2	Modificações na Arquitetura ClinicSpace	42
4.3	Diagrama de Sequência da Delegação de Tarefa para uma Especialidade Médica	43
4.4	API do Serviço de Colaboração	44
4.5	Diagrama de Classes das Tarefas Ativas e Delegadas	46
5.1	Tela do Gerenciador de Tarefas	50
5.2	Tela da Delegação de Tarefa	51
5.3	Tela para Recusar uma Tarefa Delegada	52
5.4	Comparativo de desempenho com e sem o banco de dados	53
5.5	API das Caches de Tarefas Ativas e Tarefas Delegadas	54
5.6	Comparativo de desempenho com banco de dados e com as caches	55
5.7	Comparativo de desempenho sem banco de dados e com as caches.....	56

LISTA DE CÓDIGOS FONTES

4.1	Tratamento das Requisições do Serviço de Colaboração	45
-----	--	----

LISTA DE TABELAS

5.1	Comparativo entre as infraestruturas computacionais pervasivas	49
-----	--	----

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
AVU	Ambiente Virtual do Usuário
BDA	Base de Dados da Aplicação
EHS	<i>Eletronic Health System</i>
EXEHDA	<i>Execution Environment for Highly Distributed Applications</i>
IETC	Interface de Edição de Tarefas e Contexto
pEHS	<i>pervasive Eletronic Health System</i>
SAT	Serviço de Acesso a Tarefas
SC	Serviço de Colaboração
SCT	Serviço de Contexto de Tarefas
SGDT	Subsistema de Gerenciamento Distribuído de Tarefas
SGT	Serviço de Gerenciamento de Tarefas
SI	Serviço de Interceptação
SHE	Serviço de Histórico de Execução
SPT	Serviço de Predição/Inferência de Tarefas
STA	Serviço de Tarefas Ativas
UbiComp	<i>Ubiquitous Computing</i>
URL	<i>Uniform Resource Locator</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Problema e Objetivos	16
1.2	Abordagem Proposta no Trabalho	17
1.3	Apresentação do Documento	18
2	COLABORAÇÃO EM AMBIENTES CLÍNICOS E PERVASIVOS	19
2.1	Computação Ubíqua ou Pervasiva na Área da Saúde	19
2.2	Colaboração em Ambientes Clínicos	20
2.3	Projetos Existentes	21
2.3.1	Projeto Aura	21
2.3.2	Projeto <i>Activity-Based Computing</i> (ABC)	24
3	PROJETO CLINICSPACE	28
3.1	Motivação e Problemas Abordados	28
3.2	Tarefas no ClinicSpace	29
3.3	Funcionamento do Serviço de Gerenciamento de Tarefas (SGT)	31
3.4	<i>Middleware</i> de Gerenciamento para Ambientes Pervasivos - EXEHDA	32
3.5	Arquitetura ClinicSpace	33
3.5.1	Serviço de Interceptação (SI)	35
3.5.2	Sistema de Informação Pervasivo em Saúde (pEHS - <i>pervasive Eletronic Health System</i>)	35
3.5.3	Interface de Edição de Tarefas e Contexto (IETC)	36
3.5.4	Serviço de Contextos de Tarefas (SCT)	38
3.5.5	Serviço de Gerenciamento de Tarefas (SGT)	38
3.5.6	Serviço de Tarefas Ativas (STA)	39
3.5.7	Serviço de Histórico de Execução (SHE)	39
3.5.8	Serviço de Inferência ou Serviço de Predição de Tarefas (SPT)	39
4	COLABORAÇÃO NA ARQUITETURA CLINICSPACE	40
4.1	Colaboração no ClinicSpace	40
4.2	Arquitetura do Serviço de Colaboração	41
4.3	Funcionamento do Serviço de Colaboração	43
4.4	Classes do Serviço de Colaboração	46
5	DISCUSSÃO E RESULTADOS	48
5.1	Análise dos Trabalhos Relacionados	48
5.2	Cenário de Uso	50
5.3	Análise de Desempenho do Serviço de Colaboração	53
6	CONCLUSÕES	57
6.1	Contribuições	57
6.2	Trabalhos Futuros	58
6.3	Publicações	59
	REFERÊNCIAS	60

1 INTRODUÇÃO

As principais características do trabalho em um hospital são o alto grau de mobilidade, interrupções frequentes das atividades que estão em andamento e a colaboração entre os clínicos de diferentes especialidades médicas (BARDRAM; CHRISTENSEN, 2004). Porém, grande parte dos sistemas informatizados foi projetada em um cenário de utilização onde os usuários estão sentados em suas mesas em frente a um computador. Este é um dos principais motivos da grande rejeição que os profissionais da área da saúde têm com tais sistemas (JHA et al., 2009) (MILLER; SIM, 2004).

Atualmente, pesquisas na área de tecnologia de informação têm demonstrado grande interesse em entender melhor como os profissionais de saúde trabalham (BARSOTTINI, 2007). Se, de um lado, a utilização de sistemas de informação durante o atendimento pode ser um obstáculo para o médico, entender melhor esse trabalho pode oferecer ao profissional melhor qualidade no cuidado ao paciente a partir da introdução de novas tecnologias (BARSOTTINI, 2007).

Por outro lado, a Computação Ubíqua (*Ubiquitous Computing*) ou Computação Pervasiva (*Pervasive Computing*) é um novo paradigma computacional com tecnologia de informação e comunicação em qualquer lugar, acessível por qualquer pessoa, disponível o tempo todo, onde os recursos computacionais devem estar integrados ao ambiente físico de forma transparente (WEISER, 1991). A Computação Pervasiva não pode ser interpretada somente como um ambiente impregnado por dispositivos, servindo e se adaptando para um único usuário, mas também como um lugar onde vários usuários agem e interagem uns com os outros para a realização das atividades (CABITZA et al., 2006). Assim, os conceitos e tecnologias da Computação Pervasiva podem ser explorados para construção de sistemas integrados ao cotidiano dos profissionais da área da saúde.

O desenvolvimento de um sistema integrado a esse cenário representa um desafio significativo, razão pela qual vêm sendo realizadas pesquisas com o objetivo de criar uma infraestrutura computacional que dê suporte a esses requisitos (BARDRAM, 2005) (VICENTINI et al., 2010). O projeto ClinicSpace, em desenvolvimento no Grupo de Sistemas de Computação Móvel (GMob) da Universidade Federal de Santa Maria (UFSM), tem como objetivo o desenvolvimento de uma ferramenta-piloto que permita aos médicos modelarem o auxílio às suas atividades, como profissional de saúde, utilizando tecnologias e conceitos de Computação

Ubíqua (FERREIRA, 2009). A arquitetura prototipada atende, inicialmente, a modelagem e gerenciamento das atividades clínicas, e previu serviços adicionais, dentre eles, o serviço de colaboração.

1.1 Problema e Objetivos

Como possibilitar o suporte à colaboração entre os médicos de forma integrada à arquitetura do projeto ClinicSpace?

A introdução de mecanismos para dar suporte à colaboração no projeto ClinicSpace requer a criação de um novo serviço integrado à arquitetura, chamado de Serviço de Colaboração, responsável por gerenciar a transferência (delegação) de tarefas que ainda não foram concluídas entre os usuários.

O objetivo deste trabalho é estudar os requisitos de ambientes clínicos pervasivos, modelar e implementar um serviço que dê suporte à colaboração entre os profissionais da saúde de forma integrada à arquitetura do projeto ClinicSpace.

A arquitetura do projeto ClinicSpace foi criada a partir de novos serviços que são extensões do *middleware* pervasivo EXEHDA (*Execution Environment for Highly Distributed Applications*) (YAMIN et al., 2005); portanto, para a elaboração deste trabalho foram necessários estudos sobre o *middleware* EXEHDA, bem como sobre os serviços desenvolvidos para a criação da arquitetura ClinicSpace.

Para que fosse possível atingir o objetivo do trabalho, a execução das seguintes etapas foram necessárias:

- Instalação e estudo do *middleware* EXEHDA;
- Análise da forma de modelagem dos serviços do EXEHDA, bem como o modo de interação entre eles;
- Estudo dos subsistemas da arquitetura ClinicSpace;
- Projeto da API do Serviço de Colaboração;
- Modelagem das classes utilizadas no Serviço de Colaboração, bem como de outras classes necessárias para a simulação da delegação de tarefas;
- Modelagem do banco de dados das tarefas ativas e tarefas delegadas;

- Implementação dos métodos do Serviço de Colaboração;
- Criação de um protótipo para validação do modelo, incluindo as novas funcionalidades implementadas para a realização da delegação de tarefas;
- Documentação dos novos componentes da arquitetura, bem como, das alterações nas funcionalidades já existentes;
- Elaboração de um cenário de uso, como prova de conceito para demonstrar a usabilidade e as funcionalidades do Serviço de Colaboração e sua integração com os demais serviços da arquitetura;
- Análise do desempenho das aplicações após as modificações realizadas na arquitetura do ClinicSpace.

1.2 Abordagem Proposta no Trabalho

Atualmente, grande parte dos projetos de infraestrutura computacional pervasiva que possuem suporte à colaboração dão ênfase à colaboração síncrona, ou seja, mais de uma pessoa interagindo ao mesmo instante para a realização de uma mesma atividade. Tais mecanismos são muito interessantes para auxiliar os profissionais em diagnósticos envolvendo uma equipe que não se encontra na mesma localização física.

Diferente dessa abordagem, este trabalho concentra esforços na colaboração assíncrona, ou seja, mais de uma pessoa interagindo para a realização de uma atividade, mas não ao mesmo tempo. Tal abordagem foi escolhida devido às características do projeto ClinicSpace, que tem o foco na personalização da execução das tarefas pelo clínico, respeitando a forma particular com que cada médico trabalha.

A colaboração assíncrona entre os clínicos é feita através da transferência de tarefas que ainda não foram concluídas pra outro profissional. Esta situação pode acontecer em uma troca de turno no hospital, para solicitar o parecer de um médico de outra especialidade ou simplesmente para ter uma segunda opinião de um colega.

Este trabalho descreve a criação do Serviço de Colaboração que tem como objetivo dar suporte à colaboração de forma integrada à arquitetura do projeto ClinicSpace, apresenta um cenário de uso e demonstra a manutenção do desempenho das aplicações após as modificações realizadas.

1.3 Apresentação do Documento

O restante do documento está organizado da seguinte forma. No capítulo 2 são detalhados os conceitos de Computação Ubíqua ou Pervasiva e os aspectos envolvendo colaboração em ambientes clínicos. São mostrados, também, os projetos de infraestrutura computacional pervasiva que utilizam o conceito de Computação Orientada a Atividades ou Tarefas (*Task-Oriented Computing*) de forma similar ao projeto ClinicSpace.

No capítulo 3 são detalhados os componentes da arquitetura do ClinicSpace e apresentado o conceito e o funcionamento das tarefas clínicas no âmbito do projeto. São detalhados, também, os serviços que fazem parte do Subsistema de Gerenciamento Distribuído de Tarefas, que é responsável pelo controle dos usuários (clínicos) e suas tarefas ativas, devido a sua ligação direta com o Serviço de Colaboração.

No capítulo 4 é discutida a modelagem do novo Serviço de Colaboração integrado à arquitetura ClinicSpace. São apresentados alguns pontos importantes acerca da implementação e das modificações necessárias em outros serviços do projeto para permitir o suporte à colaboração na arquitetura.

No capítulo 5 é feita uma comparação do suporte à colaboração alcançado no ClinicSpace com o existente em projetos similares. Também é apresentado um cenário de uso e são discutidos os resultados obtidos com testes de desempenho após as modificações realizadas na arquitetura.

Por fim, no capítulo 6, são registradas as conclusões e apresentadas as contribuições desta dissertação. São sugeridos, também, trabalhos futuros e listadas as publicações aceitas e as submetidas até o momento.

2 COLABORAÇÃO EM AMBIENTES CLÍNICOS E PERVASIVOS

A ideia da Computação Ubíqua ou Pervasiva surgiu em 1991 quando Mark Weiser escreveu um artigo descrevendo a sua visão de como seria a computação no século 21: *“As tecnologias mais profundas são aquelas que desaparecem. Elas se entrelaçam na vida cotidiana até que sejam indistinguíveis dela”* (WEISER, 1991).

Computação Pervasiva é um novo paradigma computacional com tecnologia de informação e comunicação em qualquer lugar, acessível por qualquer pessoa, disponível todo o tempo. Os recursos computacionais devem estar integrados ao ambiente físico da forma mais transparente possível (WEISER, 1991).

Em um cenário ideal, o usuário pode realizar as suas tarefas usufruindo ao máximo dos recursos disponíveis no ambiente onde ele se encontra, de forma transparente. Por outro lado, a Computação Pervasiva não pode ser interpretada somente como um ambiente impregnado por dispositivos que servem e se adaptam a um único usuário, mas também um lugar onde vários usuários agem e interagem uns com os outros para realizar tarefas de forma colaborativa (CABITZA et al., 2006). Esse cenário representa um desafio significativo para o projeto de aplicações pervasivas.

2.1 Computação Ubíqua ou Pervasiva na Área da Saúde

Atualmente, pesquisas na área de tecnologia de informação têm demonstrado grande interesse em entender melhor como os profissionais de saúde trabalham (BARSOTTINI, 2007). Isto é, como esses profissionais registram e utilizam as informações coletadas no cotidiano para o atendimento ao paciente.

Se, de um lado, usar sistemas de informação durante o atendimento pode ser um obstáculo para o médico, interrompendo a sinergia entre o trabalho e o registro de informações, entender melhor esse trabalho pode oferecer ao profissional, entre outras coisas, melhor qualidade no cuidado ao paciente, a partir da introdução de novas tecnologias (BARSOTTINI, 2007). Assim, nesta seção, abordam-se as características dos ambientes clínicos, bem como os aspectos colaborativos relacionados aos profissionais da saúde.

As atividades realizadas em um ambiente clínico podem ser previsíveis e planejadas ou completamente aleatórias. Algumas atividades têm prioridade enquanto outras podem ser feitas quando houver tempo. Uma determinada atividade pode ser executada somente por uma

especialidade médica específica, enquanto alguns procedimentos podem ser realizados por um conjunto maior de profissionais (KROTH; AUGUSTIN, 2010).

O trabalho dos profissionais da área da saúde é extremamente dinâmico no que se refere à mobilidade e o deslocamento no ambiente clínico com equipamentos pesados é inviável. Uma alternativa seria ter dispositivos espalhados pelo ambiente de trabalho onde os profissionais pudessem interagir a qualquer momento. Além disso, o ambiente clínico também requer o acesso a um conjunto de informações variadas e atualizadas, que podem ser requeridas por vários profissionais ao mesmo tempo (BARDRAM, 2005).

2.2 Colaboração em Ambientes Clínicos

Devido à natureza e ao número de profissionais das mais diversas áreas envolvidos no ambiente clínico, o compartilhamento de informações entre os profissionais é um ponto chave. O trabalho médico é altamente colaborativo devido à natureza especialista do tratamento: médicos de diferentes especialidades precisam colaborar através do tempo e espaço. Outro aspecto importante a ser observado é a característica assíncrona da comunicação entre esses profissionais, por exemplo, em uma troca de turno. Segundo Bardram e Christensen (2007), a colaboração é um aspecto fundamental no trabalho de profissionais da área da saúde.

Há a necessidade de mobilidade causada pela necessidade de estar em diferentes lugares físicos. Segundo Bardram e Bossen (2005), em um hospital existem locais que geralmente são específicos para um determinado propósito, ou seja, certos tipos de atividades somente podem se realizar neles, exigindo um deslocamento dos profissionais.

Uma característica destacada por Yngve (2007) é a temporalidade, onde o ambiente de trabalho é altamente distribuído não só no sentido de que os profissionais estão muitas vezes separados fisicamente (devido à constante mobilidade), mas também no sentido de que o trabalho no hospital é distribuído no tempo.

A distribuição do trabalho no tempo é um resultado da rotatividade nos turnos e também uma consequência da mudança de prioridades, causando a interrupção de tarefas de menor prioridade. Isto significa que as informações relevantes devem ser passadas de um turno para os turnos subsequentes, e que as tarefas inacabadas devem ser concluídas até o final (possivelmente por outros clínicos).

2.3 Projetos Existentes

Nesta seção são abordados alguns projetos que fornecem infraestrutura computacional (*middleware*) para sistemas pervasivos e são analisadas suas características relativas ao suporte à colaboração.

2.3.1 Projeto Aura

O projeto Aura (SOUSA; GARLAN, 2002) (GARLAN et al., 2002), desenvolvido na *Carnegie Mellon University*, tem por objetivo a construção de uma arquitetura de gerência para um sistema pervasivo, baseado nas atividades realizadas pelo usuário. O sistema busca uma aproximação com o usuário através da disponibilização, busca e adaptação de recursos, a fim de corresponder às necessidades das tarefas que estão sendo executadas pelo usuário.

Com esse foco, foi projetada uma arquitetura que dê suporte à maximização da disponibilidade de recursos através da comunicação entre dispositivos de forma inteligente, fornecendo um ambiente pervasivo. Com isso, busca-se automatizar o processo de migração e reconfiguração do ambiente computacional do usuário, minimizando a distração outrora ocasionada por esse processo.

O conceito principal da arquitetura é a “aura” computacional do usuário, a qual se constitui no mecanismo de identificação automática do usuário e de suas tarefas, bem como os requisitos necessários para sua realização. Baseado nas informações obtidas da aura computacional do usuário, a infraestrutura computacional do Aura age proativamente para buscar os serviços e recursos disponíveis e mais adequados à realização da tarefa.

O projeto Aura possui uma arquitetura baseada em serviços, onde o usuário pode mover-se entre diferentes ambientes, sempre levando consigo sua “aura” computacional, a qual é responsável por identificar os serviços necessários em um novo ambiente. Dessa forma, isenta-se o usuário da configuração de recursos computacionais, bem como das ações explícitas de migração da tarefa.

2.3.1.1 Visão Geral da Arquitetura Aura

A Figura 2.1 mostra a arquitetura utilizada no projeto Aura, incluindo os componentes clientes e seus relacionamentos:

- *Odyssey* - dá o suporte ao monitoramento e adaptação das aplicações aos recursos disponíveis;

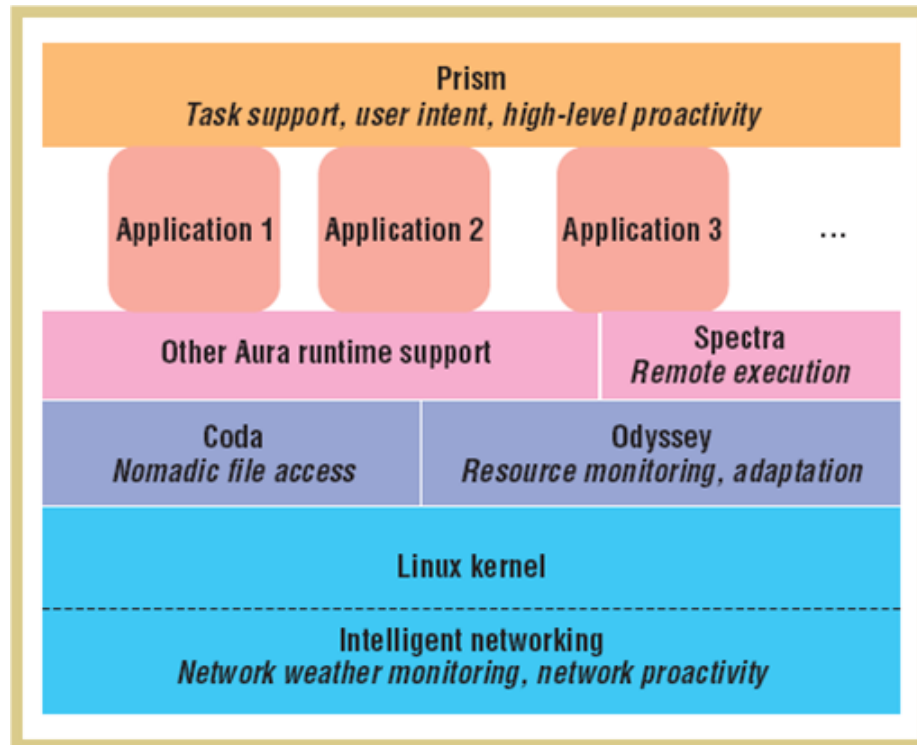


Figura 2.1: Arquitetura Aura (GARLAN et al., 2002)

- *Coda* - fornece o suporte para acesso a arquivos de forma nômade, desconectada e adaptativa à largura de banda. A implementação existente foi adaptada (SATYANARAYANAN, 2002) para atender aos requisitos da Computação Pervasiva;
- *Spectra* - é o mecanismo de execução remota adaptativa que utiliza as informações de contexto para decidir qual é a melhor forma de realizar a chamada remota. A implementação existente foi adaptada (SATYANARAYANAN, 1996) para atender aos requisitos de Computação Pervasiva;
- *Prism* - principal camada do projeto que personifica o conceito de Aura pessoal; é o gerenciador de tarefas, responsável por capturar e gerenciar a intenção do usuário dando suporte à proatividade e auto ajuste das aplicações (Figura 2.2).

A arquitetura do *Prism*, o gerenciador de tarefas, foi modelada na forma de quatro componentes principais:

- Gerenciador de Tarefas (*Task Manager*) - responsável por minimizar a distração do usuário, atuando (i) na migração das informações, quando o usuário se desloca para outro ambiente, (ii) na descoberta de serviços compatíveis com a tarefa do usuário, (iii) na coordenação do salvamento da situação atual, (iv) na inicialização de uma nova tarefa

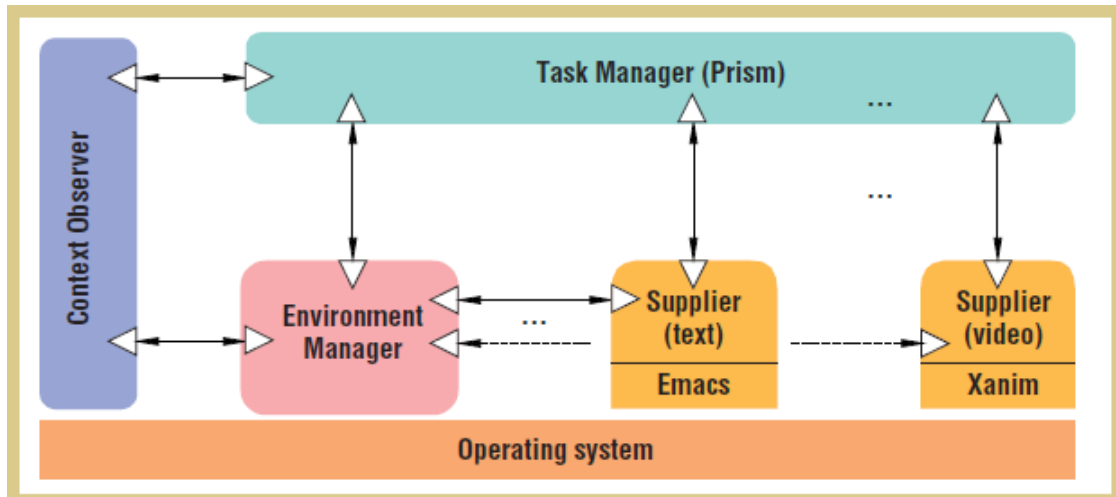


Figura 2.2: Componentes do *Prism*(GARLAN et al., 2002)

requerida pelo usuário ou pelo contexto e (v) na suspensão da execução ou no ajuste das partes afetadas da tarefa quando as restrições (como por exemplo privacidade e atividade do usuário) não são satisfeitas;

- Provedores de Serviço (*Service Suppliers*) - fornecem serviços abstratos que serão utilizados para compor as tarefas. Na prática, esses serviços abstratos são implementados através de um *wrapping* das aplicações e serviços existentes para a API do Aura;
- Gerenciador de Ambiente (*Environment Manager*) - faz o mapeamento dos recursos e serviços disponíveis no ambiente; também encapsula o mecanismo de acesso aos arquivos distribuídos;
- Observador de Contexto (*Context Observer*) - responsável por fornecer informações sobre o contexto físico e retornar eventos do mesmo contexto para o Gerenciador de Tarefas e o Gerenciador de Ambiente.

2.3.1.2 Colaboração no Projeto Aura

O projeto Aura não possui a característica de colaboração às tarefas de forma nativa em sua infraestrutura computacional. Um caso de colaboração desenvolvido no projeto foi o *Idealink* (GARLAN et al., 2002), aplicação desenvolvida utilizando a infraestrutura Aura. O *Idealink* é uma espécie de quadro negro distribuído e compartilhado utilizando colaboração síncrona, ou seja, suporte a múltiplas sessões onde os usuários podem escrever ou desenhar simultaneamente. A infraestrutura Aura possui suporte a alguns requisitos que podem ser utilizados para colaboração, porém deixa a cargo das aplicações essa responsabilidade.

Conclui-se que Aura disponibiliza alguns conceitos importantes para a colaboração como: atividade (tarefa), parada-reativação (*suspend-resume*), e deslocamento (*roaming*). Entretanto, não foi publicado nenhum resultado explorando os aspectos colaborativos, fundamentais em ambientes clínicos.

2.3.2 Projeto *Activity-Based Computing* (ABC)

O projeto ABC (BARDRAM, 2003) (BARDRAM, 2005) (BARDRAM; CHRISTENSEN, 2007) (BARDRAM, 2009) foi criado em 2001, em uma iniciativa conjunta entre a principal empresa dinamarquesa de sistemas EPR (*Electronic Patient Record*), um grande hospital universitário e o Centro de Computação Pervasiva para a saúde (*Centre for Pervasive Healthcare*). O foco principal do projeto é a criação de um suporte computacional para as atividades humanas, em especial, o trabalho médico em hospitais.

A filosofia do projeto ABC é a mobilidade e a colaboração, inerente a ambientes clínicos. Para tanto, foi desenvolvido um *framework*, que é a infraestrutura de suporte para a computação pervasiva que tem os seguintes objetivos:

- dar suporte a atividade humana;
- dar suporte a mobilidade através de ambientes computacionais heterogêneos;
- dar suporte a colaboração assíncrona, ou seja, várias pessoas participando de uma atividade;
- dar suporte a colaboração síncrona, ou seja, colaboração em tempo real, como por exemplo, vídeo conferência.

A abordagem utilizada pelo projeto foi a da computação baseada em atividades, onde a atividade é a entidade principal e que pode ser gerenciada pela infraestrutura computacional ou *middleware*.

Dentro dos conceitos do projeto ABC, a infraestrutura computacional permite que as atividades possam ser inicializadas, suspensas, retomadas e finalizadas, assim é possível que haja interrupções e trabalho em múltiplas tarefas fazendo com que as atividades sejam suspensas e depois retomadas. A característica de suporte à persistência implementada na infraestrutura computacional permite que uma atividade seja suspensa em um ambiente e retomada em outro, atendendo aos requisitos de mobilidade do usuário de ambientes clínicos.

2.3.2.1 Visão Geral da Arquitetura ABC

A arquitetura possui uma plataforma para o desenvolvimento e distribuição de aplicações que utiliza os conceitos de ABC e está dividida em três camadas, conforme a Figura 2.3: a camada de infraestrutura, responsável por gerenciar as atividades colaborativas e distribuídas através da adaptação dos recursos ou serviços disponíveis em um ambiente específico; a camada cliente, que gerencia a atividade em um dispositivo específico; e a camada de aplicação, que possui uma API para o desenvolvimento de aplicações ABC.

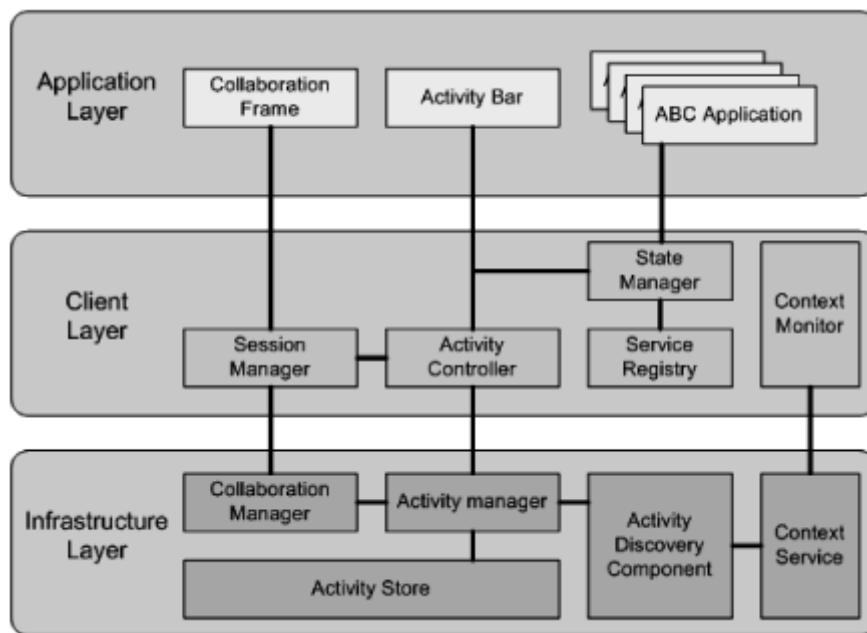


Figura 2.3: Arquitetura ABC (BARDRAM, 2009)

Camada de Infraestrutura:

- Banco de Dados de Atividade (*Activity Store*) - responsável por manipular a persistência das atividades através de uma interface para criar, apagar e obter uma atividade, mantendo um histórico de utilização de cada usuário;
- Gerenciador de Atividade (*Activity Manager*) - gerencia o comportamento em tempo de execução das atividades, possibilitando que elas sejam criadas, inicializadas, suspensas, retomadas e finalizadas pelo cliente;
- Gerenciador de Colaboração (*Collaboration Manager*) - responsável pelas requisições para colaboração síncrona em uma atividade; para tal finalidade é criado um objeto de sessão para cada atividade colaborativa em andamento;

- Serviço de Contexto (*Context Service*) - responsável por coletar as informações de contexto que serão utilizadas pelo Monitor de Contexto;
- Componente de Descoberta de Atividade (*Activity Discovery Component*) - trabalha em conjunto com o Serviço de Contexto, possui um conjunto de regras com o qual busca uma atividade existente ou então cria uma nova atividade que seja relevante em um contexto específico;

Camada Cliente:

- Gerenciador de Sessão (*Session Manager*) - responsável por gerenciar (pelo lado cliente) uma sessão de atividade colaborativa que está em andamento;
- Controlador de Atividade (*Activity Controller*) - é a ligação entre o cliente (dispositivo) e a infraestrutura (*Activity Manager*);
- Registro de Serviço (*Service Registry*) - componente responsável por descobrir e registrar os serviços locais disponíveis;
- Gerenciador de Estado (*State Manager*) - responsável por manipular o estado da atividade, que são as informações de estado de cada um dos serviços que participam da atividade;
- Monitor de Contexto (*Context Monitor*) - responsável pelo monitoramento do contexto do dispositivo e a sua utilização;

2.3.2.2 Colaboração no Projeto ABC

O projeto *Activity-Based Computing* possui algumas características importantes para o trabalho colaborativo:

Interrupção e retomada de atividades (*suspende-resume*) - o trabalho em hospitais é altamente colaborativo e, por isso, interrupções são frequentes (BARDRAM; CHRISTENSEN, 2007). Este princípio permite que os usuários mudem rapidamente de atividade, como por exemplo: um clínico está trabalhando em uma atividade A e ele inicia a atividade B, automaticamente a atividade A fica com situação “interrompida” até que ele posteriormente possa retomar a atividade.

Compartilhamento de atividades (*activity sharing*) - o trabalho médico, para o atendimento a um paciente, pode envolver várias especialidades e o tratamento e os cuidados distribuídos a múltiplos profissionais. O princípio do compartilhamento de atividades permite que vários usuários trabalhem em uma mesma atividade, podendo ser ao mesmo tempo (colaboração síncrona) ou em tempos diferentes (colaboração assíncrona). Nos dois casos a abordagem do projeto é a mesma, através do princípio de interrupção e retomada da atividade; os usuários que tiverem acesso podem retomar a atividade em qualquer tempo. A colaboração assíncrona ocorre quando usuários diferentes em tempos diferentes retomam a mesma atividade (depois que um usuário interrompe o outro retoma). No outro caso, quando vários usuários retomam uma atividade ao mesmo tempo ocorre a colaboração síncrona (um usuário retoma a atividade enquanto outro ainda não a interrompeu).

Através de funcionalidades direcionadas aos usuários, podem-se utilizar mecanismos síncronos ou assíncronos de comunicação entre os profissionais de diferentes especialidades, permitindo o compartilhamento de informações e mecanismos colaborativos.

3 PROJETO CLINICSPACE

O projeto ClinicSpace (FERREIRA et al., 2009) (FERREIRA, 2009) (MACHADO, 2010) (RIZZETTI, 2009) (SILVA, 2009) (SOUZA, 2010) (VICENTINI, 2010), em desenvolvimento pelo Grupo de Sistemas de Computação Móvel (GMob) da Universidade Federal de Santa Maria (UFSM), tem como objetivo utilizar tecnologias de Computação Pervasiva para auxiliar os profissionais da área da saúde na execução de suas tarefas clínicas dentro de um ambiente hospitalar.

A ideia central do projeto é a criação de uma ferramenta-piloto que permita aos clínicos a personalização da execução de suas tarefas, as quais são gerenciadas por uma infraestrutura computacional (*middleware*) específica para um ambiente pervasivo.

3.1 Motivação e Problemas Abordados

Os atuais Sistemas Eletrônicos de Saúde (EHS - *Electronic Health System*) são projetados e desenvolvidos com a visão hospitalar corporativa, a qual busca prover um gerenciamento em termos de negócios da instituição. Pesquisas demonstram (BARDRAM, 2005) que existe uma grande rejeição por parte dos clínicos aos EHS devido a essa estruturação de negócio vinculado aos sistemas de saúde. Este boicote ao sistema traz inúmeros problemas à realização das atividades diárias, como a falta ou minimização de informações históricas de consultas dos pacientes, prejudicando diagnósticos que necessitam de maiores informações históricas para serem providos.

A construção de EHS, com a visão centrada no usuário clínico (médico), está se tornando requisito para diminuição da rejeição atual aos sistemas de saúde. Devido às características dinâmicas encontradas na área clínica, sua construção se torna um desafio.

Como visto, algumas características do meio clínico são mobilidade e trabalho colaborativo. No cotidiano de um médico, as atividades podem sofrer interrupções para realização de atividades emergenciais e o retorno a ela, quando possível. A atividade pode ser delegada a outro profissional para sua conclusão (BARDRAM; CHRISTENSEN, 2007). Profissionais precisam deslocar-se pela instituição, e os sistemas devem garantir ao profissional rápido acesso aos dados do paciente, em qualquer ponto do hospital. Logo, o EHS deve ter uma semântica “siga-me” (*follow-me*) (AUGUSTIN; YAMIN; GEYER, 2005), deslocar-se junto com o médico (migração) e adaptar-se às situações onde se encontra (contexto), além de necessitar prover

informações históricas dos pacientes implementadas em uma base de dados pervasiva: sempre disponível a qualquer momento (*anytime*), em qualquer lugar (*anywhere*) e em qualquer dispositivo (*any device*).

As características encontradas nos ambientes clínicos podem ser gerenciadas por sistemas ubíquos (BARDRAM, 2003). Neste sentido, o projeto ClinicSpace busca prover uma infraestrutura computacional para auxiliar o médico em suas atividades diárias, permitindo que este programe a sua forma individual que deseja realizar uma determinada atividade (personalização). Com isso, não são necessários conhecimentos sobre sistemas gerados por analistas, que podem modificar o fluxo de trabalho dos usuários pela falta ou má definição dos requisitos do sistema.

Devido à complexidade computacional necessária para suprir as características do ambiente clínico, o projeto ClinicSpace procura realizar rotinas proativas, como a entrada de dados através da captura de contexto, minimizando a complexidade para o usuário clínico (RIZZETTI, 2009).

Sendo assim, o projeto ClinicSpace usa conceitos da Computação Ubíqua para gerenciamento e adaptabilidade dos sistemas ao meio clínico, buscando equilibrar a proatividade (agir em nome do usuário) com a personalização (forma individual de cada usuário ao realizar uma atividade), visando diminuir a rejeição encontrada nos sistemas EHS atuais.

3.2 Tarefas no ClinicSpace

Procurando prover o máximo de integração com o usuário, a arquitetura do ClinicSpace foi projetada através da concepção da Teoria da Atividade (RANGANATHAN; CAMPBELL, 2005). O termo atividade refere-se ao processo realizado por humanos de forma colaborativa, coordenada, distribuída, em um espaço determinado. A Teoria da Atividade humana usa seis componentes para modelar a atividade: sujeito, objeto, ferramentas, regras, comunidade e colaboração.

Aplicando ao ambiente clínico o modelo genérico proposto pela Teoria da Atividade de Ranganathan e Campbell (2005), tem-se o médico atuando como sujeito, que tem por objetivo diagnosticar e/ou tratar o paciente que, por sua vez, atua como objeto da atividade. Para realizar a atividade, o médico utiliza ferramentas de mediação, as quais consistem em registros, procedimentos, equipamentos e recursos. Essa mediação, geralmente, segue regras especificadas através de guias clínicos. A atividade é executada no ambiente clínico, sendo esta a sua

comunidade, onde há uma diversidade de profissionais que, normalmente, fazem uma divisão do trabalho para a sua realização.

Ao modelar essas noções na arquitetura do projeto ClinicSpace, a atividade clínica foi decomposta em um conjunto de tarefas - ações que têm o auxílio de recursos computacionais, e seguem a forma particular de cada indivíduo de realizá-la (personalização) (SILVA, 2009).

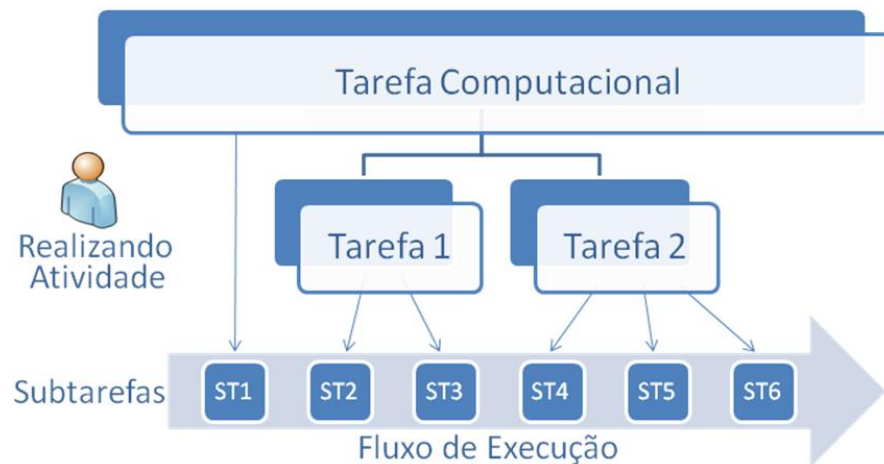


Figura 3.1: Conceito de tarefas na arquitetura ClinicSpace

As tarefas simples são compostas por subtarefas, que são operações básicas do sistema, aplicações Java disponibilizadas para composição das tarefas pelo clínico. Quando agrupadas, as tarefas simples formam uma tarefa composta, que segue um fluxo de execução (semelhante a um sistema de workflow) ilustrado na Figura 3.1. Essa modelagem dá às tarefas as características de decomposição, recombinação e reuso (FERREIRA, 2009) (VICENTINI, 2010).

As subtarefas foram categorizadas da seguinte forma (SILVA, 2009):

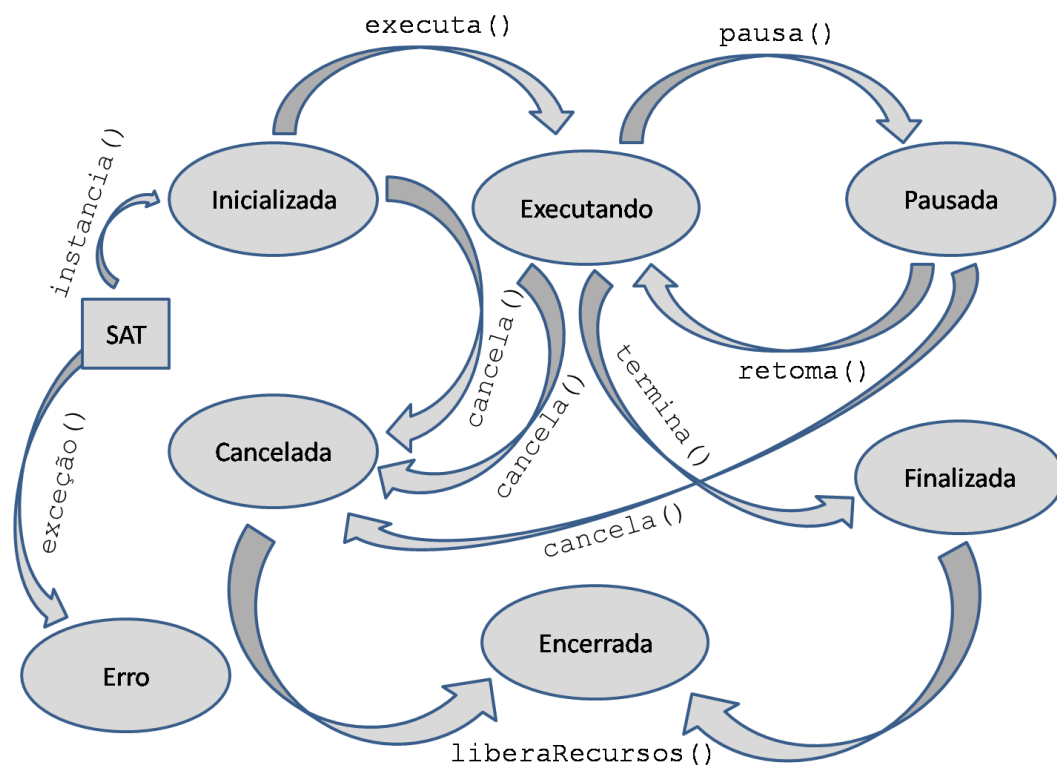
- **Identificação:** permitem identificação automática de profissionais e pacientes;
- **Busca:** responsáveis por recuperar informações de profissionais e pacientes;
- **Preenchimento:** registram as informações dos pacientes no sistema de saúde;
- **Visualização:** responsáveis por mostrar as informações dos pacientes.

O usuário interage com o sistema, personaliza suas tarefas, as quais são armazenadas com o vínculo a esse usuário. Assim, cada médico tem um conjunto de tarefas vinculadas ao seu perfil.

As tarefas são modeladas como elementos dinâmicos e de definição recursiva, que podem conter (i) subtarefas ou (ii) uma tarefa inteira composta por subtarefas, e contida em outra tarefa de propósito mais abrangente.

3.3 Funcionamento do Serviço de Gerenciamento de Tarefas (SGT)

A Figura 3.2 representa a máquina de estados para tarefas. Uma tarefa instanciada pelo Serviço de Acesso a Tarefas (SAT) e configurada pelo SGT está no estado “Inicializada”. No momento em que o Serviço de Gerenciamento de Tarefas inicia a execução da tarefa, o estado dela passa a ser “Executando”.



SAT – Serviço de Acesso a Tarefas

Figura 3.2: Máquina de Estados para as Tarefas (FERREIRA, 2009)

Estando, a tarefa, no estado “Executando”, ela pode assumir três outros estados: (i) “Cancelada”, se o usuário cancelar a execução; (ii) “Finalizada”, se todas as suas subtarefas forem concluídas ou (iii) “Pausada”, quando sua execução é interrompida. A interrupção pode ser solicitada tanto pelo usuário quanto por outros serviços, como o Serviço de Interceptação (para tratar um evento do sistema pervasivo de informação de saúde (pEHS)) e o Serviço de Gerenciamento de Tarefas (SGT), para migração da tarefa.

Do estado “Pausada”, a tarefa pode retornar ao estado “Executando”, caso o usuário solicite

ao SGT sua retomada. Ou pode ir para o estado “Cancelada”, se o usuário não desejar continuar a execução da tarefa. Já, dos estados “Finalizada” e “Cancelada”, uma tarefa só pode passar ao estado “Encerrada”, quando o SGT libera os recursos alocados a ela, tornando-a indisponível (FERREIRA, 2009).

3.4 *Middleware* de Gerenciamento para Ambientes Pervasivos - EXEHDA

O *middleware* EXEHDA (YAMIN et al., 2005) fornece os serviços de mais baixo nível da arquitetura proposta, abstraindo do programador de subtarefas e aplicações pEHS a dinamicidade e heterogeneidade do ambiente pervasivo, o controle da adaptação, a mobilidade física e lógica de recursos, a portabilidade e a conectividade no tratamento de conexões e desconexões realizadas por dispositivos de acesso sem fio.

O EXEHDA disponibiliza os serviços básicos responsáveis pelo gerenciamento pervasivo de ambientes, que são divididos nos seguintes subsistemas:

- **Subsistema de Reconhecimento de Contexto e Adaptação:** responsável pelo gerenciamento dos elementos do contexto, onde os dados coletados pelos sensores são fornecidos através de um serviço de subscrição (*publish/subscribe*¹);
- **Subsistema de Execução Distribuída:** responsável pelo suporte ao processamento distribuído no EXEHDA. Esse subsistema interage com os outros subsistemas, como o de reconhecimento do contexto e adaptação, para promover uma execução efetivamente pervasiva. Além disso, ele gerencia a execução e a migração das aplicações no ambiente pervasivo;
- **Subsistema de Comunicação:** responsável pelo gerenciamento da troca de mensagens e a descoberta dinâmica de outros serviços e recursos separados geograficamente. Disponibiliza mecanismos para atender os aspectos relacionados às desconexões existentes em um ambiente pervasivo, devido tanto à infraestrutura de hardware sem fio como às estratégias de economia de energia dos dispositivos móveis;
- **Subsistema de Acesso Pervasivo:** tem por finalidade dar suporte à premissa da Computação Pervasiva de acesso em qualquer lugar e todo o tempo a dados e código. É responsável por gerenciar a Base de Dados da Aplicação (BDA) e também o Ambiente

¹ *Publish/subscribe* é um padrão de mensagens em que os remetentes (*publishers*) não enviam as mensagens diretamente a receptores específicos (*subscribers*).

Virtual do Usuário (AVU), disponibilizando-o para acesso a partir de qualquer dispositivo (FERREIRA, 2009).

3.5 Arquitetura ClinicSpace

A arquitetura do projeto ClinicSpace foi construída através da modificação e criação de novos serviços que foram adicionados como extensões ao *middleware* pervasivo EXEHDA (YAMIN et al., 2005).

A arquitetura para a programação e gerenciamento personalizado das tarefas (Figura 3.3) foi organizada em níveis que refletem as visões do sistema:

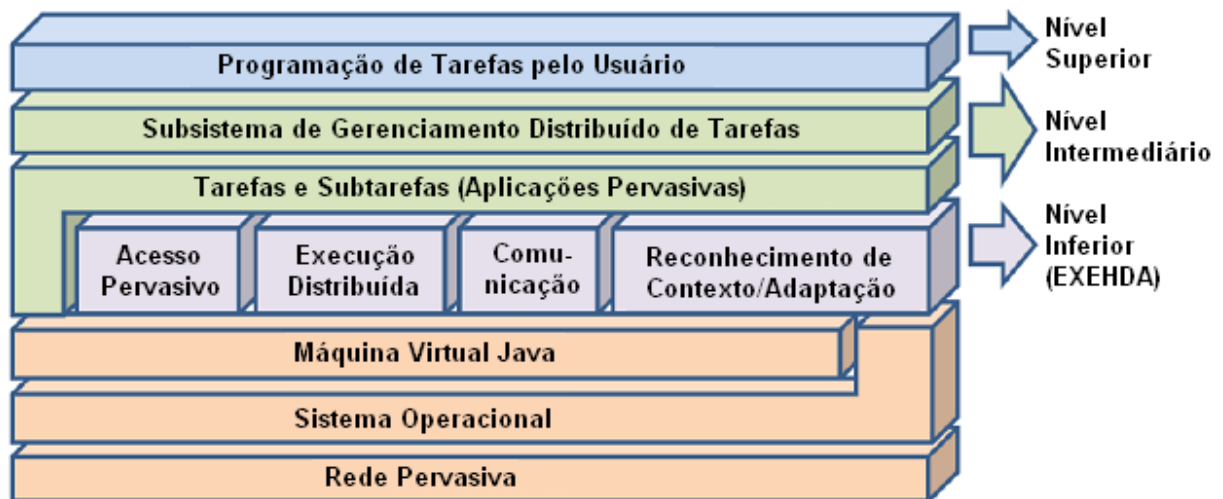


Figura 3.3: Arquitetura para programação e gerenciamento de tarefas (FERREIRA, 2009)

- **Nível superior:** é composto pelo usuário-final (médico) que interage com a ferramenta para (re) definir suas tarefas que executarão num ambiente pervasivo;
- **Nível intermediário:** é composto pelo mapeamento entre tarefas (definidas pelo usuário) e subtarefas (aplicações pervasivas) e pelo gerenciamento de ambas;
- **Nível inferior:** é composto pelo conjunto de serviços do *middleware* de gerenciamento do ambiente pervasivo e de suporte à execução das aplicações pervasivas: EXEHDA.

O *middleware* pervasivo EXEHDA não foi projetado para ter suporte a tarefas, conceito necessário para o projeto ClinicSpace, então foi desenvolvido um novo subsistema na infraestrutura computacional chamado Subsistema de Gerenciamento Distribuído de Tarefas (SGDT), que tem a função de fazer a ligação entre tarefas e aplicações pervasivas, conforme definidas

na arquitetura do *middleware*. A extensão criada é responsável por gerenciar, em alto nível, as tarefas, delegando o gerenciamento das subtarefas (aplicações pervasivas) para os serviços atuais do EXEHDA.

Na interação do usuário com o ambiente pervasivo (hospital), mais especificamente na sua movimentação pelo ambiente, existem três casos a serem considerados:

- O usuário pode mover-se dentro do ambiente pervasivo portando um dispositivo móvel. Nesse caso não há desconexão e a sessão do usuário permanece ativa durante o deslocamento;
- O usuário pode mover-se pelo ambiente sem carregar um dispositivo. Nesse caso, ao sair do dispositivo onde estava trabalhando, o sistema encerra a sessão aberta, interrompendo e armazenando as tarefas que estavam em execução. Ao aproximar-se de outro dispositivo, seu login é habilitado nesse dispositivo, bastando o usuário logar-se para que sua sessão seja restaurada, e suas tarefas sejam mostradas na tela do dispositivo;
- O usuário pode trocar de dispositivo sem mover-se. Nesse caso, ele pode deslogar-se do dispositivo que está usando e logar-se no que irá usar. Ou, simplesmente, logar-se no dispositivo que irá utilizar, fazendo com que o sistema encerre sua sessão no dispositivo que estava sendo usado. Em ambos os casos, as tarefas em execução são interrompidas, armazenadas e disponibilizadas para continuação na tela do dispositivo no qual o usuário se logou.

Toda vez que a sessão do usuário é encerrada, o novo subsistema do EXEHDA faz a migração da sessão e das tarefas para um servidor, com o auxílio dos outros serviços do *middleware*. Da mesma forma, o SGGT usa esses serviços para restaurar a sessão e as tarefas do usuário, quando esse logar em um dispositivo (FERREIRA, 2009).

A arquitetura ClinicSpace foi desenvolvida para se tornar um ambiente voltado às atividades clínicas em conjunto com um Sistema de Informação em Saúde (pEHS - *pervasive Electronic Health System*) (VICENTINI, 2010) e o ambiente de execução pervasivo, gerenciado pelo *middleware* EXEHDA (YAMIN et al., 2005).

A arquitetura possui componentes bem definidos que permitem a modelagem de tarefas pelo usuário final, bem como o gerenciamento do ciclo de vida das tarefas. A Figura 3.4 mostra uma visão geral do subsistema de gerenciamento distribuído de tarefas e os serviços que fazem parte da arquitetura ClinicSpace, que serão descritos nas próximas seções.

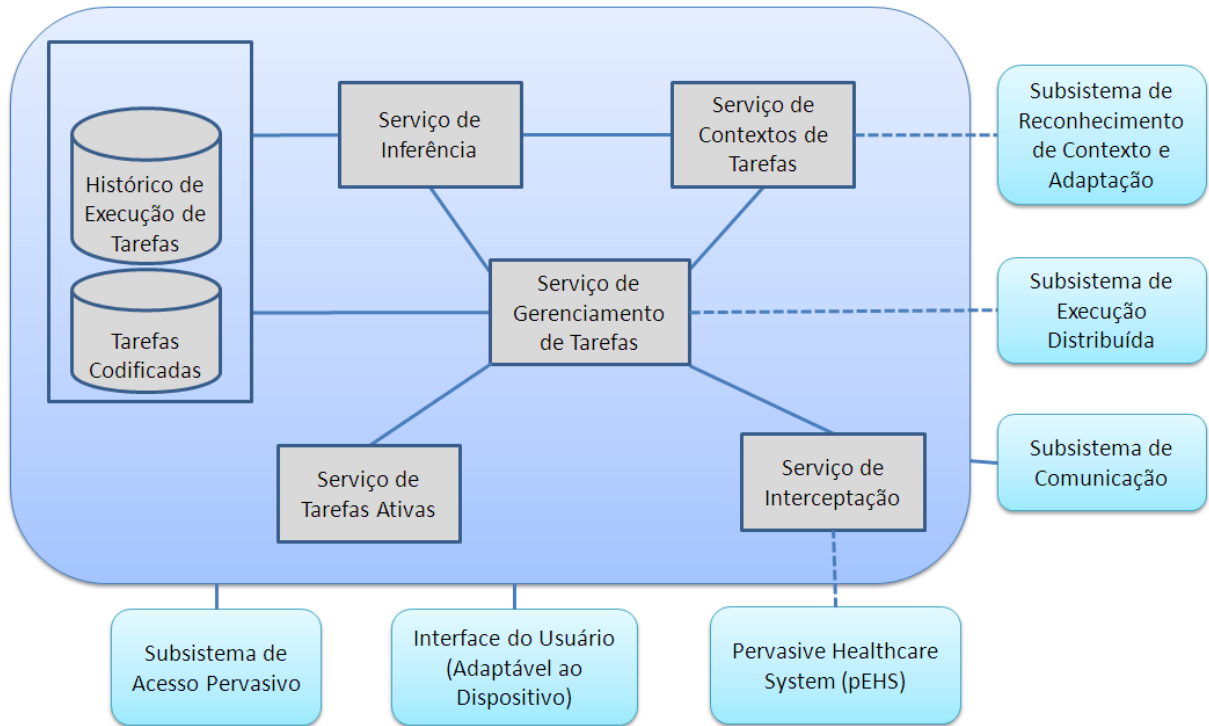


Figura 3.4: Subsistema de Gerenciamento Distribuído de Tarefas (FERREIRA, 2009)

3.5.1 Serviço de Interceptação (SI)

O Serviço de Interceptação é responsável por fazer o tratamento de eventos gerados pelo sistema pervasivo de informação de saúde (pEHS). Os eventos podem ser críticos, quando exigem a atenção imediata do usuário, ou normais, quando o usuário deve ser apenas notificado.

Esse serviço é bastante dependente do pEHS, pois somente após as funcionalidades desse estarem bem definidas é que se poderá classificar seus eventos e fazer a integração com o Serviço de Interceptação. Essa integração envolve, além do tratamento de eventos, a troca de informações sobre o usuário, para que o sistema pEHS tenha conhecimento se ele está usando o sistema computacional ou não.

3.5.2 Sistema de Informação Pervasivo em Saúde (pEHS - *pervasive Electronic Health System*)

De maneira a permitir a utilização e validação da arquitetura ClinicSpace, foram pesquisados Sistemas Eletrônicos de Saúde (EHS) existentes e, a partir desse estudo, foi projetado um novo sistema com características de pervasividade, chamado pEHS (VICENTINI, 2010). Entre os principais requisitos desta aplicação estão: colaboração entre profissionais para a realização de tarefas, compartilhamento de informações e suporte à mobilidade.

As funcionalidades projetadas para o pEHS foram desenvolvidas na menor granularidade possível para que elas possam ser utilizadas isoladamente para os mais diversos fins na composição das tarefas dos usuários. Com isso, adotou-se o conceito de subtarefa que representa o mapeamento entre uma funcionalidade presente no pEHS e o restante da arquitetura ClinicSpace. Como exemplo de subtarefa, pode-se citar: “identificação de paciente”, “busca de exames visuais do paciente”, etc.

A composição de várias subtarefas compõe a tarefa a ser executada pelo usuário, representando o fluxo de execução necessário para a realização de uma ou várias atividades do seu cotidiano.

A partir deste requisito, fica evidente que o pEHS deve ser altamente modularizado, permitindo que cada funcionalidade possa ser mapeada para uma subtarefa e um conjunto de subtarefas possa formar atividades. Caso o pEHS possua uma granularidade muito alta das suas funções, ou seja, agrupando várias funcionalidades que não podem ser decompostas, corre-se o risco de que as tarefas modeladas pelo usuário não representem a melhor forma de realização do seu trabalho (VICENTINI, 2010).

A falta de flexibilidade de um sistema de informação hospitalar poderia levar a uma rejeição do sistema por parte dos usuários, contrariando o objetivo principal da arquitetura ClinicSpace que é permitir que o usuário modele o sistema de acordo com as suas rotinas (FERREIRA et al., 2009)

3.5.3 Interface de Edição de Tarefas e Contexto (IETC)

Com o intuito de disponibilizar ao usuário uma ferramenta de modelagem de suas tarefas, está sendo desenvolvida a Interface de Edição de Tarefas e Contexto. Essa ferramenta-piloto é constituída de uma interface para que o usuário modele/ programe as suas tarefas de acordo com a sua forma particular/personalizada de realizar as atividades clínicas diárias.

Para fazer a composição de uma determinada tarefa, o usuário seleciona as subtarefas disponíveis (relacionadas às funcionalidades mínimas disponíveis no pEHS) e as ordena de maneira a refletir o seu fluxo de trabalho. Dessa forma, a execução das tarefas no sistema ocorrerá de acordo com o modo de trabalho de cada usuário, garantindo a personalização do sistema (SILVA, 2009).

A Figura 3.5 ilustra a interface gráfica da Interface de Edição de Tarefas e Contexto (IETC). Pode-se perceber a presença de uma janela que disponibiliza ao usuário as subtarefas passíveis

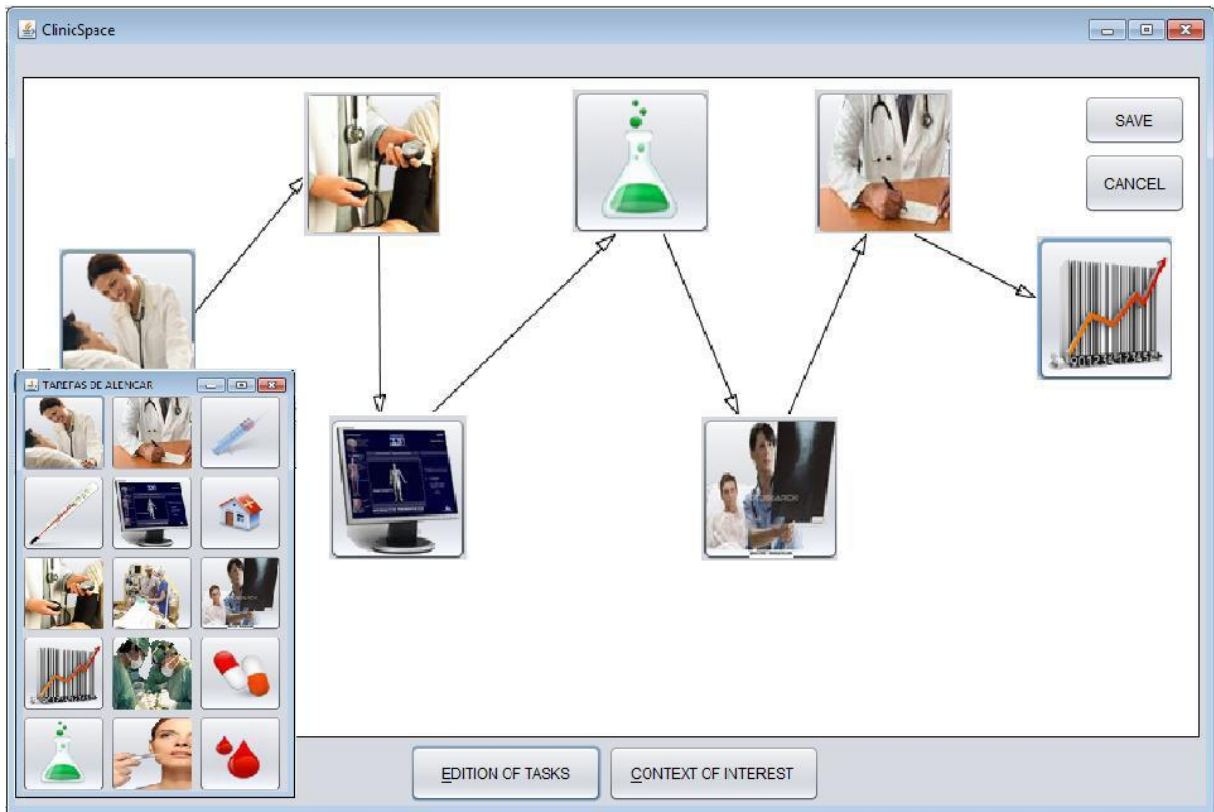


Figura 3.5: Interface de Edição de Tarefas e Contexto (MACHADO, 2010)

de serem usadas, representadas por ícones, de forma a tornar seu uso mais natural. Conforme a tarefa modelada, o usuário escolhe as subtarefas de seu interesse e as arrasta para o painel, aonde deve ser especificada a ordem, através de setas, na qual o fluxo de execução das subtarefas deve ocorrer.

Como forma de implantar características do ambiente nas tarefas, o usuário tem a opção de anexar elementos do contexto disponível para cada subtarefa, ao acionar o botão “*Context of Interest*”(MACHADO, 2010). A interface exhibe quais os elementos do contexto podem ser introduzidos em cada subtarefa, bem como um formulário no qual o usuário especifica o comportamento do sistema em relação ao estado do contexto.

A partir dessas configurações, o usuário pode escolher quais alertas serão emitidos quando uma determinada condição for satisfeita, permitindo que o sistema aja proativamente a seu favor. Assim, as características do contexto influenciam no comportamento da aplicação e podem auxiliar o usuário na tomada de decisões, por exemplo.

Após finalizar a modelagem da tarefa, o usuário salva a modelagem/programação realizada e o sistema a armazena como um descritor XML, contendo todas as subtarefas envolvidas, bem como a sequência configurada. No momento de execução de uma tarefa, o Sistema de Gerenci-

amento de Tarefas (SGT) identifica as subtarefas que compõem a tarefa e as instancia conforme a sua execução acontece, obedecendo às configurações realizadas na sua modelagem feita na IETC.

3.5.4 Serviço de Contextos de Tarefas (SCT)

Esse serviço armazena o relacionamento das tarefas do usuário com o contexto configurado no momento da modelagem. Além disso, o SCT identifica elementos do contexto que podem servir para a diminuição da entrada manual de dados nos formulários do pEHS, facilitando o seu uso e diminuindo a propensão a erros humanos (RIZZETTI, 2009). Ainda, o usuário pode realizar o agendamento de execução de determinadas tarefas quando o contexto estiver em um estado predeterminado.

3.5.5 Serviço de Gerenciamento de Tarefas (SGT)

O objetivo desse serviço é realizar o mapeamento entre as subtarefas do ClinicSpace e os serviços da plataforma EXEHDA, usada para prover a pervasividade do sistema como um todo. Em alguns casos, uma subtarefa pode envolver a execução de diversos serviços EXEHDA para que a sua funcionalidade seja alcançada.

O principal papel desse serviço é tratar o mapeamento entre as funcionalidades do sistema pEHS e as subtarefas modeladas nas tarefas do usuário. Assim, sempre que uma subtarefa for executada, esse serviço identifica qual a respectiva funcionalidade do sistema pEHS deve entrar em execução e invoca os serviços EXEHDA. É responsabilidade do SGT, ainda, gerenciar o ciclo de vida das tarefas. Caso o usuário necessite parar temporariamente ou finalizar uma tarefa, o SGT executa as operações necessárias para atualizar o seu estado ou terminá-la, mantendo assim o controle de execução do sistema (FERREIRA et al., 2009).

O SGT utiliza a base de Tarefas Codificadas que contém os descritores das tarefas modeladas de cada usuário, armazenadas na forma de arquivos XML. No momento em que uma tarefa é iniciada, o SGT busca o seu descritor XML e identifica a sequência de subtarefas necessárias. De acordo com o andamento da execução, o sistema instancia as demais subtarefas necessárias até que a tarefa seja concluída.

O Serviço de Gerenciamento de Tarefas atua como servidor para a maioria dos demais componentes da arquitetura, uma vez que ele é o responsável por instanciar e executar as tarefas do usuário.

3.5.6 Serviço de Tarefas Ativas (STA)

Esse serviço tem como objetivo manter as tarefas estão em execução pelos usuários do sistema para realizar a migração entre dispositivos de acordo com o deslocamento do usuário. Para tal, o STA utiliza serviços do *middleware* EXEHDA, para identificar a movimentação do usuário e realizar a semântica *follow-me*, trazendo dinamicidade ao sistema como um todo (FERREIRA et al., 2009).

O Serviço de Tarefas Ativas, pela sua natureza, possui uma forte ligação com o SGT, citado anteriormente, uma vez que ele é notificado sempre que uma tarefa entra em execução ou é finalizada.

3.5.7 Serviço de Histórico de Execução (SHE)

Esse componente da arquitetura tem como objetivo armazenar o histórico de execução das tarefas executadas pelos usuários, assim como o contexto captado no início da execução das tarefas. A partir desse armazenamento, realizado em Banco de Dados relacional, pode-se traçar o perfil de utilização do usuário e, com isso, realizar inferências sobre as tarefas a serem executadas no futuro.

Outra vantagem da utilização do histórico é a criação de *logs* da utilização do sistema permitindo auditorias sobre as ações tomadas pelos usuários, bem como das circunstâncias (contexto) presentes na execução das tarefas.

3.5.8 Serviço de Inferência ou Serviço de Predição de Tarefas (SPT)

Esse serviço tem a responsabilidade de inferir as próximas tarefas do usuário de acordo com o seu histórico de execução e a situação do ambiente captada no momento da sua execução. Para a realização desse objetivo, o Serviço de Inferência possui uma forte interação com os serviços citados anteriormente (SGT e SHE), assim como com os componentes do *middleware* EXEHDA para a captura do contexto.

De forma a não tornar os serviços envolvidos na inferência de tarefas demasiadamente acoplados entre si, utilizou-se uma arquitetura com interfaces bem definidas e adoção de padrões de projeto, visando facilitar as manutenções futuras.

4 COLABORAÇÃO NA ARQUITETURA CLINICSPACE

Neste capítulo é detalhado o novo Serviço de Colaboração adicionado à arquitetura do projeto ClinicSpace. São apresentados os principais pontos sobre a implementação e as modificações realizadas em outros serviços do projeto para dar o suporte à colaboração na arquitetura.

4.1 Colaboração no ClinicSpace

O Serviço de Colaboração é responsável por disponibilizar a comunicação e a transferência de tarefas que ainda não foram concluídas entre os usuários. Esse serviço foi modelado para atender ao requisito de colaboração de ambientes clínicos, aspecto fundamental no trabalho dos profissionais da saúde (BARDRAM; CHRISTENSEN, 2007). O suporte à colaboração foi projetado com duas funcionalidades: Comunicação e Delegação de Tarefas em Execução.

Uma forma comum de colaboração entre os profissionais da saúde é a consulta à opinião de outro profissional (BARDRAM; CHRISTENSEN, 2007). Por isso, foi previsto, para o Serviço de Colaboração, o suporte à comunicação dos usuários, inicialmente por troca de mensagem, podendo ser expandido para formas mais robustas, como vídeoconferência. Através dessa funcionalidade, os profissionais podem compartilhar informações sobre o paciente, discutir problemas e soluções, solicitar e dar opiniões sobre determinado caso (FERREIRA, 2009).

Uma possível implementação para essa funcionalidade é a criação de uma tarefa dedicada à comunicação entre os usuários. Essa tarefa não seria exatamente uma tarefa clínica. Ela executaria em *background*, funcionando como um cliente do Serviço de Colaboração, notificando o usuário sobre novas mensagens e disponibilizando uma interface para leitura e escrita.

Outra forma de colaboração em ambientes hospitalares é a transferência de tarefas em execução entre os clínicos, realizada principalmente na troca de plantão. Para isso, o sistema permite aos usuários a transferência das tarefas ainda não concluídas para outro profissional.

Como visto, a arquitetura ClinicSpace é baseada em tarefas. Tarefas são definidas como um conjunto de ações (atividades) executadas colaborativamente por humanos e pelo sistema pervasivo para alcançar um objetivo. Uma tarefa pode ser composta, estática ou dinamicamente, por um número de subtarefas que podem ser unidas usando um conceito similar a sistema de workflow (AUGUSTIN; LIMA; YAMIN, 2006).

Dessa forma, a funcionalidade de transferência de tarefas em execução foi modelada da seguinte maneira: quando um usuário recebe a tarefa que outro transferiu para ele, o sistema

permite ao profissional que recebeu a tarefa escolher entre aceitá-la ou rejeitá-la. Ao rejeitá-la, a tarefa retorna ao emissor. Já aceitando a tarefa, ela será incorporada à sua lista de tarefas interrompidas, podendo ser reiniciada ou retomada do ponto onde foi parada.

Como o EXEHDA não incorpora os conceitos de tarefas e colaboração entre os usuários, não foi prevista nenhuma forma de troca de objetos (do EXEHDA) entre os usuários. Isso impõe um problema para a implementação da transferência de tarefas em execução para outro usuário.

Para evitar os problemas causados por essa limitação, o controle de usuários e suas tarefas ativas foi implementado no Subsistema de Gerenciamento Distribuído de Tarefas (SGDT). Assim, o EXEHDA controla o Ambiente Virtual do Usuário, e o SGDT controla a lista de tarefas ativas dos usuários. Com isso, podem-se trocar tarefas entre os usuários, sem que haja necessidade de envolver diretamente os outros serviços do *middleware*.

4.2 Arquitetura do Serviço de Colaboração

O Serviço de Colaboração (SC) é o novo componente integrado à arquitetura ClinicSpace e é responsável pelo gerenciamento da troca (delegação) de tarefas ativas entre os usuários. A Figura 4.1 mostra uma visão geral do SC e sua interação com o Gerenciador de Tarefas e o STA.

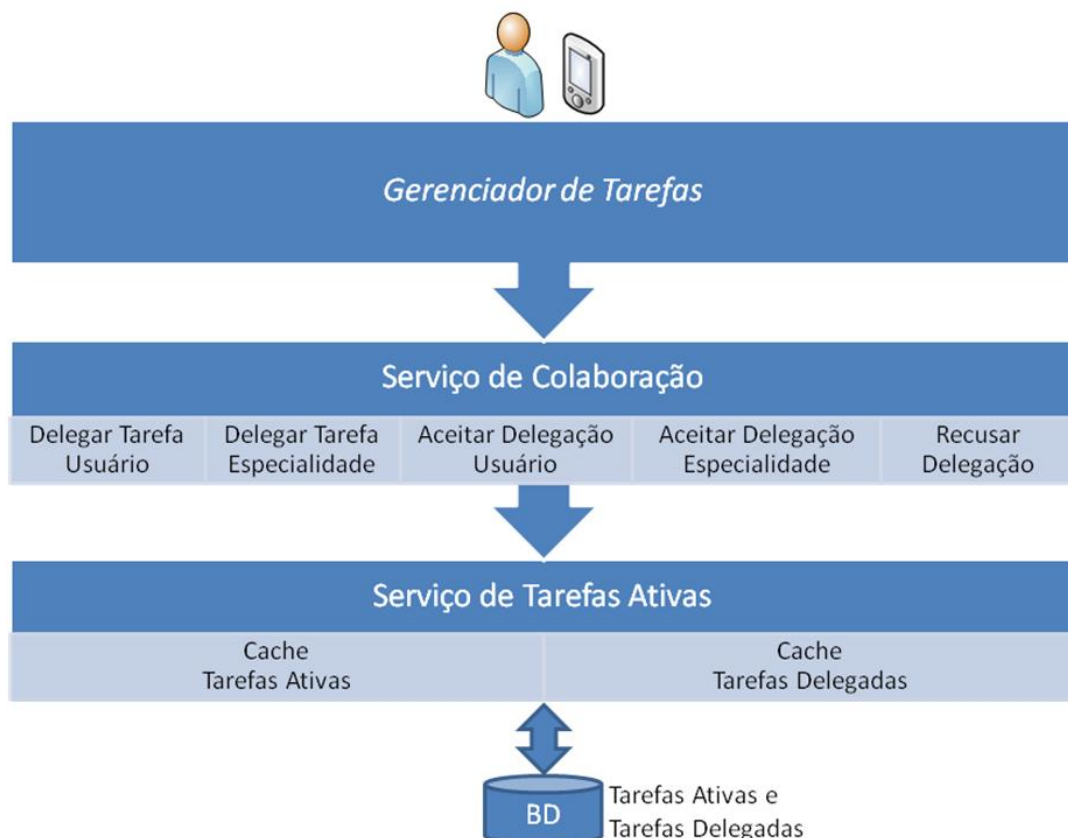


Figura 4.1: Serviço de Colaboração

Só é possível realizar a delegação para outro usuário de uma tarefa ativa. Uma tarefa é considerada ativa quando ela foi inicializada pelo Serviço de Gerenciamento de Tarefas e ainda não foi concluída nem cancelada.

Além da criação do Serviço de Colaboração, outros componentes da arquitetura sofreram modificações para que fosse possível dar suporte à colaboração de forma nativa na infraestrutura computacional.

A Figura 4.2 mostra os componentes da arquitetura do projeto ClinicSpace que foram criados ou sofreram modificações (em laranja) e também os demais componentes que permaneceram sem alterações.

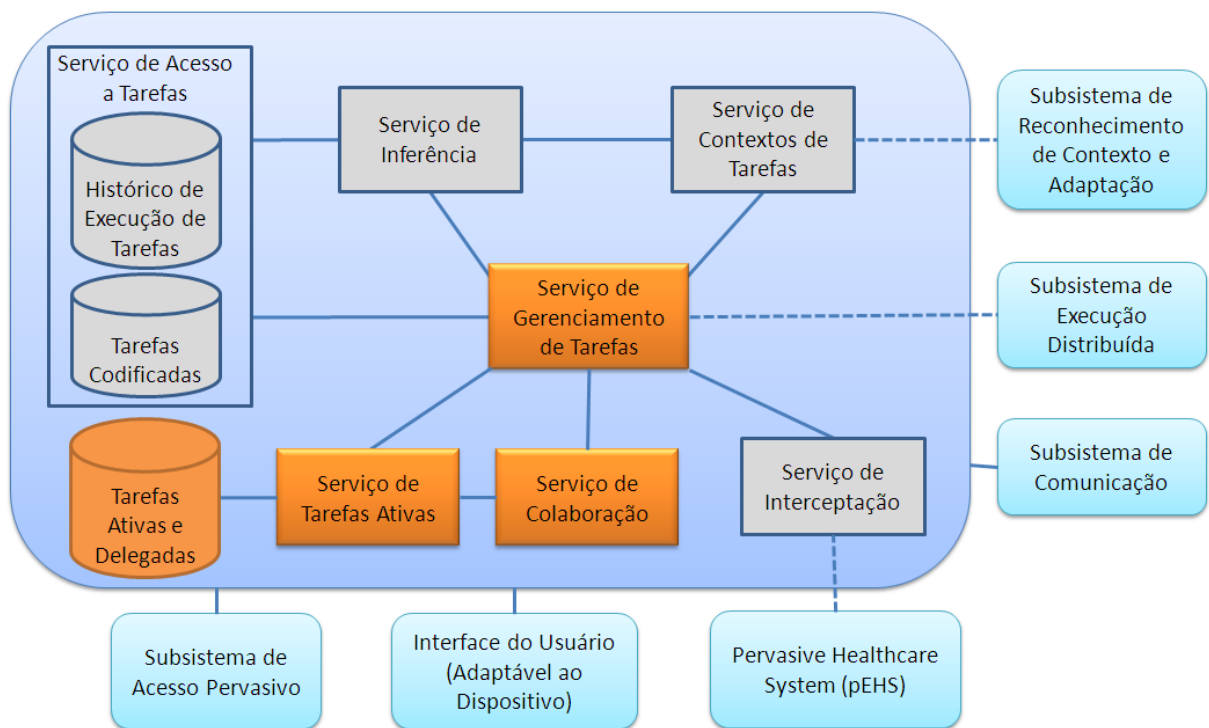


Figura 4.2: Modificações na Arquitetura ClinicSpace

Os componentes que foram criados ou sofreram modificações na arquitetura do ClinicSpace são:

- **Serviço de Colaboração** (novo) - foi criado para dar suporte à colaboração no projeto através da delegação de tarefas.
- **Banco de Dados de Tarefas Ativas e Tarefas Delegadas** (novo) - foi adicionado um novo componente na arquitetura para dar suporte à persistência das tarefas ativas e das tarefas delegadas em substituição ao controle exclusivo em memória que existia para as tarefas ativas.

- **Serviço de Tarefas Ativas** (modificado) - foi alterado para utilizar o novo banco de dados criado e, por motivos de otimização de desempenho (mais detalhes na seção 5.3), foi necessária a criação de uma cache para as tarefas ativas e outra cache para as tarefas delegadas.
- **Serviço de Gerenciamento de Tarefas** (modificado) - o serviço foi alterado para adicionar o conceito de especialidade médica relacionado aos usuários do ClinicSpace (clínicos).

4.3 Funcionamento do Serviço de Colaboração

O Diagrama de Sequência é uma das ferramentas UML (*Unified Modeling Language*) usadas para representar interações entre objetos de um cenário, realizadas através de operações ou métodos (procedimentos ou funções), dando ênfase a ordenação temporal em que as mensagens são trocadas entre os objetos de um sistema (OMG, 2010). A Figura 4.3 mostra o diagrama de sequência da delegação de tarefa para uma especialidade.

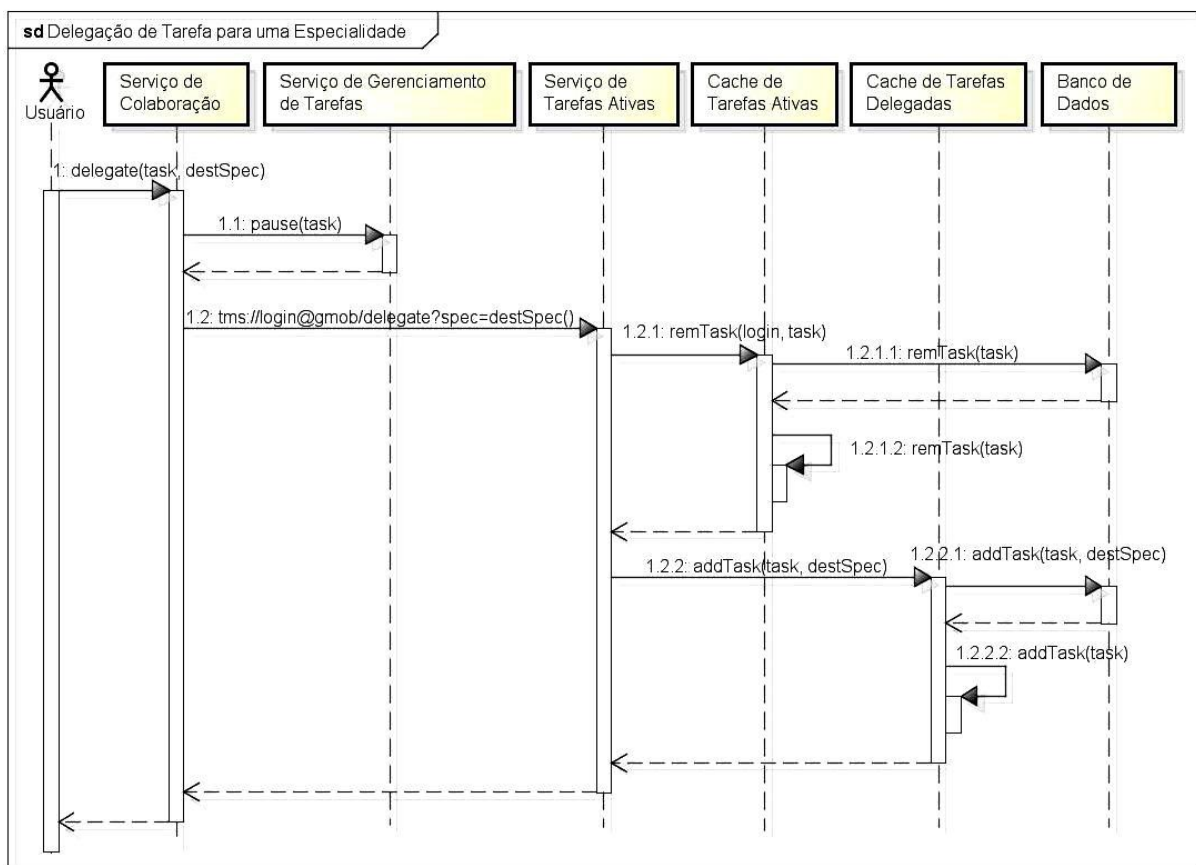


Figura 4.3: Diagrama de Sequência da Delegação de Tarefa para uma Especialidade Médica

Quando o médico realiza uma delegação de tarefa através do gerenciador de tarefas é utilizado o método `delegateTask(task, destUser, destSpec)` (1), fornecido pela API (*Application Programming Interface* ou Interface de Programação de Aplicações) do Serviço de Colaboração (Figura 4.4), que recebe como parâmetros a tarefa e a especialidade para qual será delegada. Quando é feita uma delegação para uma especialidade médica, o parâmetro `destUser` é passado um valor nulo, no outro caso quando a delegação é feita para um usuário, então o parâmetro `destSpec` é nulo.

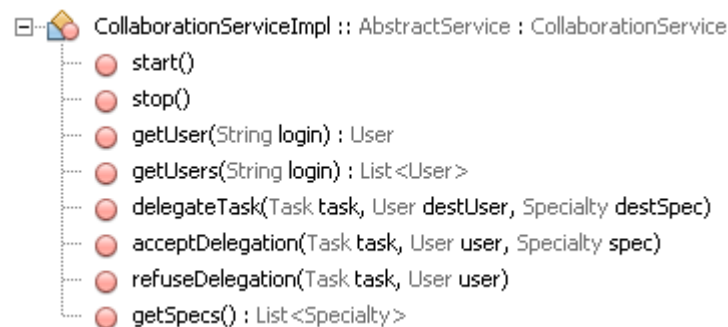


Figura 4.4: API do Serviço de Colaboração

O Serviço de Colaboração, então, verifica se a tarefa está em execução e, se estiver, pausa a tarefa (1.1). No momento que a tarefa é pausada (pode ser que ela já esteja pausada no momento da delegação), o objeto é serializado e persistido no banco de dados. Logo após, o Serviço de Colaboração faz uma requisição ao Serviço de Tarefas Ativas através de um protocolo próprio que posteriormente será transformado em uma conexão HTTP (1.2) (KROTH; AUGUSTIN; MACHADO, 2011).

O protocolo utilizado tem um formato padrão para a construção da URL (*Uniform Resource Locator*): `protocol://user@cell/op`, onde:

- *protocol* - indica o protocolo usado;
- *user* - indica para qual usuário a operação está sendo solicitada;
- *cell* - indica a EXEHDAcel (área de atuação) a qual pertence o usuário;
- *op* - indica a operação que está sendo solicitada.

Exemplo de URL construída:

`tms://tuco@gmob/delegate?spec=1`, onde:

- *tms* (*task manager service*) - é o protocolo utilizado, que depois será transformado em uma requisição HTTP;
- *tuco* - é o usuário que está solicitando a delegação da tarefa;
- *gmob* - é a EXEHDAcel;
- *delegate* - é a operação solicitada.

Os demais parâmetros (neste exemplo *spec* representa a especialidade médica) são passados utilizando o formato *query string*¹.

Após processar a requisição vinda do Serviço de Colaboração, a tarefa é retirada da lista de tarefas ativas do usuário (1.2.1) e adicionada à lista de tarefas delegadas da especialidade (1.2.2). Quando um médico, que possui a especialidade para qual a tarefa foi delegada, atribuir a tarefa para si, a tarefa será removida da lista de tarefas delegadas e adicionada na lista de tarefas ativas do usuário. Quando o clínico retomar a tarefa, o objeto é buscado do banco de dados, podendo continuar a tarefa do ponto onde parou.

O Serviço de Tarefas Ativas foi implementado na arquitetura ClinicSpace como uma extensão do *HttpService* do EXEHDA, que é usado como base para todos os serviços do tipo cliente/servidor do *middleware*.

```

1  protected void handleRequest(ServiceRequest req) {
2      switch (getSgtOperation(req)) {
3          case OP_GET_DELEGATED_USER:
4              handleSgtGetDelegatedUser(req);
5              break;
6          case OP_GET_DELEGATED_SPECS:
7              handleSgtGetDelegatedSpecs(req);
8              break;
9          case OP_DELEGATE:
10             handleSgtDelegate(req);
11             break;
12         case OP_DO_DELEGATE:
13             handleSgtDoDelegate(req);
14             break;
15         case OP_UNDO_DELEGATE:
16             handleSgtUndoDelegate(req);
17             break;
18     }
19 }

```

Código Fonte 4.1: Tratamento das Requisições do Serviço de Colaboração

O Serviço de Tarefas Ativas faz o processamento das requisições vindas do Serviço de Colaboração (Código Fonte 4.1) para: (i) buscar as tarefas delegadas para o usuário (linha 3), (ii) buscar as tarefas delegadas para a(s) especialidade(s) do médico (linha 6), (iii) delegar uma

¹parte de uma URL que contém dados que são passados para as aplicações web

tarefa para outro usuário (linha 9), (iv) delegar uma tarefa para uma especialidade (linha 9), (v) aceitar uma tarefa delegada por outro usuário (linha 12), (vii) atribuir para si uma tarefa delegada para a sua especialidade médica (linha 12) e (vi) recusar uma tarefa delegada por outro usuário (linha 15).

4.4 Classes do Serviço de Colaboração

A Figura 4.5 mostra o Diagrama de Classes das Tarefas Ativas e Delegadas e seus relacionamentos.

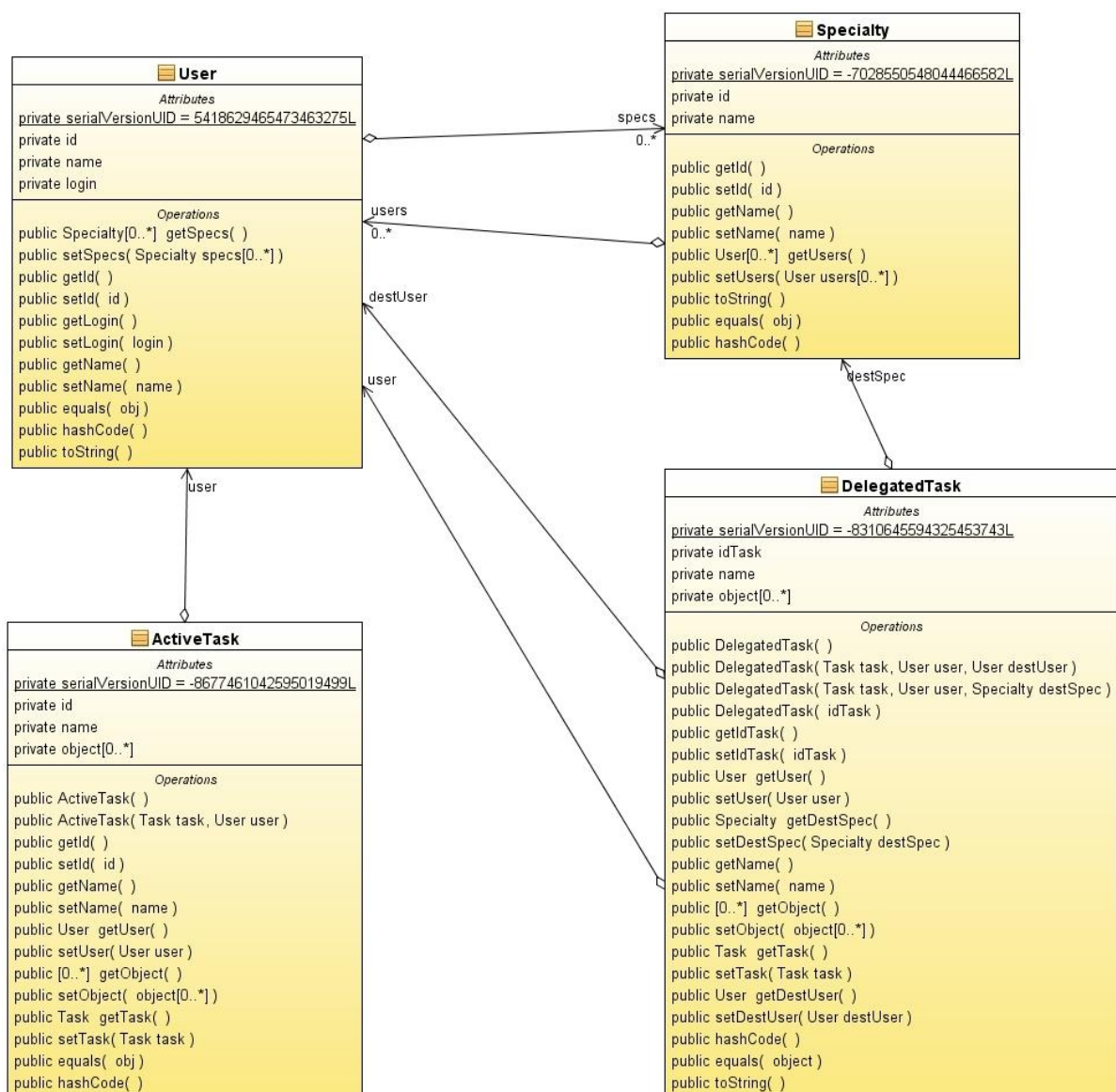


Figura 4.5: Diagrama de Classes das Tarefas Ativas e Delegadas

Foram criadas as classes `User` (usuário) e `Specialty` (especialidade) para poder realizar as simulações do protótipo. A identificação do usuário é uma informação vinda do contexto

utilizando, por exemplo, a tecnologia RFID (*Radio-Frequency IDentification* ou Identificação por Rádio Frequência). A(s) especialidade(s) médica relacionada(s) com o clínico é uma informação buscada através da integração com o pEHS.

As classes `ActiveTask` (tarefa ativa) e `DelegatedTask` (tarefa delegada) possuem o atributo `object` que é o próprio objeto da tarefa que está em execução. Quando a tarefa é pausada, o objeto é serializado e armazenado no banco de dados e quando a tarefa é retomada, o objeto é recuperado e desserializado. No caso da delegação de uma tarefa, o objeto que contém a tarefa em execução é movido da classe `ActiveTask` para a classe `DelegatedTask`. Quando a delegação de uma tarefa é aceita pelo clínico (no caso de delegação para o usuário) ou quando o médico atribui para si uma tarefa delegada (no caso de delegação para uma especialidade), o usuário da tarefa é alterado² e, em seguida, é realizado o movimento no sentido contrário, ou seja, o objeto da tarefa é movido da classe `DelegatedTask` para a classe `ActiveTask`.

²Neste caso é alterado o atributo `proprietario` do objeto da classe `Tarefa` que está serializado

5 DISCUSSÃO E RESULTADOS

Neste capítulo é realizada uma análise comparativa do suporte à colaboração implementada no ClinicSpace com o existente em outros projetos. Também é apresentado um cenário de uso e são discutidos os resultados alcançados com os testes de desempenho após as modificações na arquitetura.

5.1 Análise dos Trabalhos Relacionados

Várias pesquisas combinam alguma característica de ambientes clínicos com requisitos de sistemas ubíquos ou pervasivos (FERRAZ; DINIZ, 2010) (SAIN; LEE; CHUNG, 2009) (TENTORI; FAVELA, 2008); porém, existem poucos esforços para a criação de uma infraestrutura para atender às principais características do trabalho do profissional da saúde no ambiente hospitalar.

Alguns projetos serviram como fonte de inspiração para as ideias utilizadas no projeto ClinicSpace: o projeto Aura e o projeto *Activity-Based Computing* (ABC), apresentados na seção 2.3. O ClinicSpace dá suporte à definição, armazenamento e migração de tarefas entre dispositivos, requisitos fundamentais para atender às necessidades de mobilidade e interrupção.

As características de mobilidade, interrupção e colaboração relacionadas às atividades da área da saúde fazem com que ambientes clínicos tenham um grande potencial para pesquisas sobre infraestruturas computacionais. Os projetos ABC e ClinicSpace foram inovadores ao aplicar os conceitos e tecnologia de Computação Ubíqua ou Pervasiva para atender os requisitos de tais ambientes com o objetivo de melhorar o trabalho dos clínicos.

Para uma análise comparativa, foram observados os tipos de colaboração que são suportados de forma nativa na infraestrutura computacional dos projetos: colaboração síncrona (vários usuários em tempo real; por exemplo, videoconferência) e colaboração assíncrona (várias pessoas participando de uma atividade em momentos diferentes; por exemplo, delegação de tarefas por troca de turno).

As características que foram analisadas nos projetos são:

- **automatização** - capacidade de continuidade da execução das tarefas de forma automática no deslocamento de um ambiente para outro;
- **controle** - ações explícitas disponíveis para o usuário que possibilitam executar, inter-

romper, continuar ou cancelar uma tarefa;

- **personalização** - possibilidade de customização/programação de acordo com as preferências pessoais do usuário;
- **colaboração síncrona** - colaboração de vários usuários em tempo real, por exemplo, videoconferência;
- **colaboração assíncrona** - várias pessoas participando, em momentos diferentes, de uma mesma atividade.

Tabela 5.1: Comparativo entre as infraestruturas computacionais pervasivas

	Aura	ABC	ClinicSpace
Automatização	x	x	x
Controle		x	x
Personalização			x
Colaboração Síncrona		x	
Colaboração Assíncrona		x	x

A Tabela 5.1 mostra um comparativo realizado entre as infraestruturas computacionais dos projetos com relação às características presentes de forma nativa no *middleware* para o gerenciamento e execução das tarefas.

O projeto Aura (SOUSA; GARLAN, 2002) (GARLAN et al., 2002) serviu de inspiração para vários outros projetos de infraestrutura computacional pervasiva por ser o precursor a utilizar o conceito de computação baseada em tarefas e possuir requisitos que podem ser utilizados para colaboração. A infraestrutura pode ser usada para o desenvolvimento de alguma aplicação com finalidade específica; por exemplo, um quadro colaborativo para auxiliar em diagnósticos usando como base o Idealink. Porém, a colaboração não é uma característica intrínseca às atividades na arquitetura Aura.

O projeto ABC (BARDRAM, 2003) (BARDRAM, 2005) (BARDRAM; CHRISTENSEN, 2007) (BARDRAM, 2009) tem como ponto forte a colaboração síncrona, utilizada principalmente para a discussão de diagnósticos, possuindo mecanismos de comunicação entre os profissionais envolvidos, como compartilhamento de imagens de raios-X. A colaboração assíncrona usa os mesmos mecanismos da colaboração síncrona, ou seja, os profissionais autorizados podem participar e modificar a atividade a qualquer momento.

O projeto ClinicSpace possui o foco na atividade médica e tem o diferencial de permitir personalização das tarefas por parte do usuário. Permite a colaboração assíncrona entre os

médicos através da delegação de tarefas, que pode ser usada para solicitar a opinião de outro especialista ou encaminhar a um colega uma atividade que ainda não foi concluída, semelhante a um sistema de workflow.

5.2 Cenário de Uso

Como prova de conceito, faz-se um cenário de uso, que visa demonstrar a usabilidade e funcionalidades do Serviço de Colaboração e sua integração com os demais serviços da arquitetura ClinicSpace. Para tal, considera-se o seguinte cenário: o Dr. Kroth chega para trabalhar no hospital em seu último dia antes das férias. Ele inicia o turno indo ao escritório localizado na ala leste do hospital. O sistema detecta automaticamente sua presença e abre uma tela de confirmação de identificação no computador da sala. Depois da verificação feita, a lista dos pacientes que estão sob seu cuidado é mostrada na tela, além das tarefas que foram delegadas (Figura 5.1).

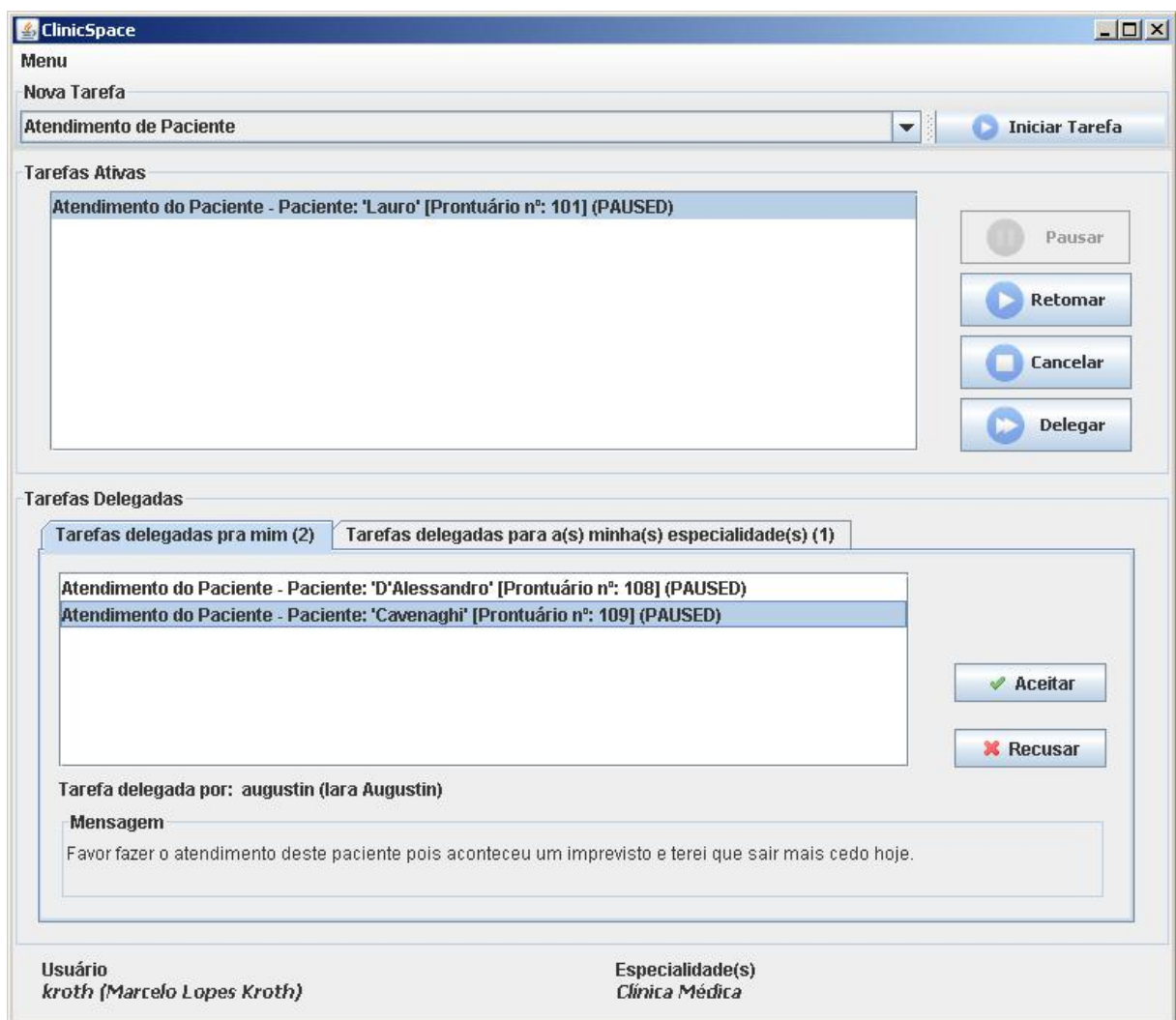


Figura 5.1: Tela do Gerenciador de Tarefas

Neste caso de uso, existe uma aplicação que é responsável pelo gerenciamento das tarefas do usuário utilizando a infraestrutura da arquitetura ClinicSpace.

É possível pausar, retomar, cancelar e delegar as tarefas do usuário que estão ativas, bem como iniciar uma nova tarefa. Na mesma tela do gerenciador de tarefas também são mostradas (i) as tarefas delegadas para o usuário e (ii) as tarefas delegadas para sua(s) a(s) especialidade(s), permitindo que o usuário as aceite ou recuse.

Após fazer o planejamento de suas atividades, o Dr. Kroth vai até o quarto do primeiro paciente que ele irá visitar hoje. Chegando lá, ele se aproxima de uma tela localizada junto ao leito e confirma uma mensagem que está perguntando se ele quer continuar o atendimento ao paciente.

Deve-se salientar que, neste cenário de uso, o ClinicSpace, utilizando informações de contexto, detecta a presença do médico no quarto do paciente e sugere a retomada da tarefa que está pausada envolvendo o paciente do quarto.

Dr. Kroth termina a visita, as informações são atualizadas no sistema, e encaminha, utilizando o botão “Delegar” na tela do gerenciador de tarefas, a tarefa do atendimento para a Dra. Cameron, que divide com ele o atendimento ao paciente.

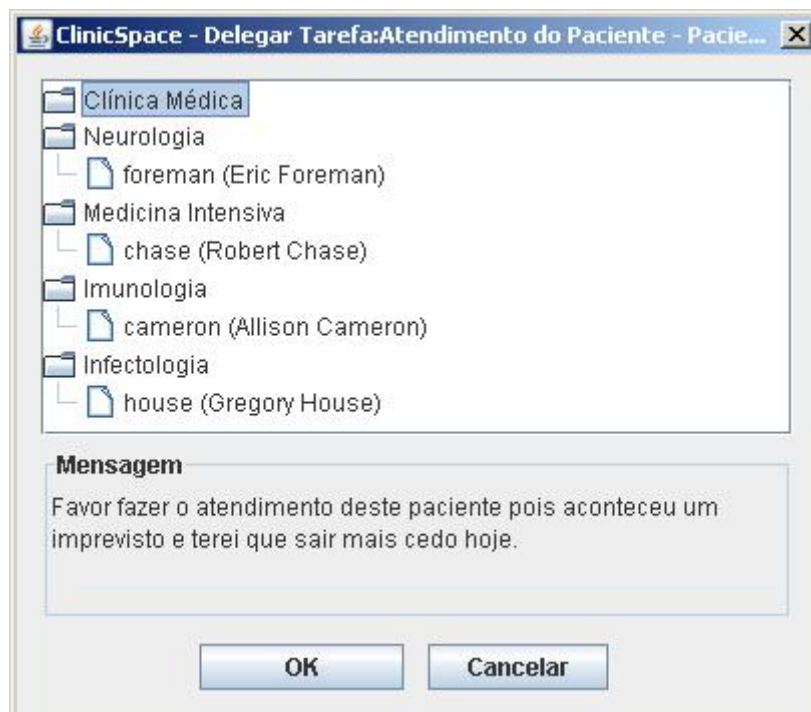


Figura 5.2: Tela da Delegação de Tarefa

A Figura 5.2 mostra a tela da delegação de tarefa onde o médico pode escolher delegar a tarefa para uma especialidade médica ou para algum profissional específico, informando uma

mensagem para quem receber a tarefa.

O Dr. Kroth sai do quarto e, enquanto espera o elevador (todos no hospital reclamam da demora do elevador), ele retira um dispositivo móvel do bolso e consulta o resultado dos exames do segundo paciente que ele está indo visitar.

Chegando ao quarto do paciente, os dados dos exames realizados são mostrados na tela junto ao leito. Algumas anomalias no resultado dos exames, combinados com alguns sintomas do paciente, faz o Dr. Kroth tomar a decisão de pedir uma opinião para um colega de outra especialidade. Ele explica a situação ao paciente e, utilizando a tela junto ao leito, encaminha a solicitação para a especialidade médica desejada. Neste caso, o médico aciona o botão “Delegar” no gerenciador de tarefas e escolhe a especialidade médica desejada na tela da delegação de tarefa.

O Dr. Kroth volta ao escritório da ala leste do hospital. O sistema detecta sua presença na sala e lhe mostra o resultado dos exames de um paciente que o Dr. House lhe encaminhou. Neste caso, o Dr. House usou a funcionalidade “Delegar” do gerenciador de tarefas e encaminhou a tarefa para o Dr. Kroth. Na tela do Dr. Kroth a tarefa apareceu na aba “Tarefas delegadas para mim” do gerenciador de tarefas. O Dr. Kroth aceita a tarefa, que passa a ser mostrada na lista de tarefas ativas. Agora o Dr. Kroth pode retomar a tarefa, analisar o resultado dos exames e delegar a tarefa de volta ao Dr. House.

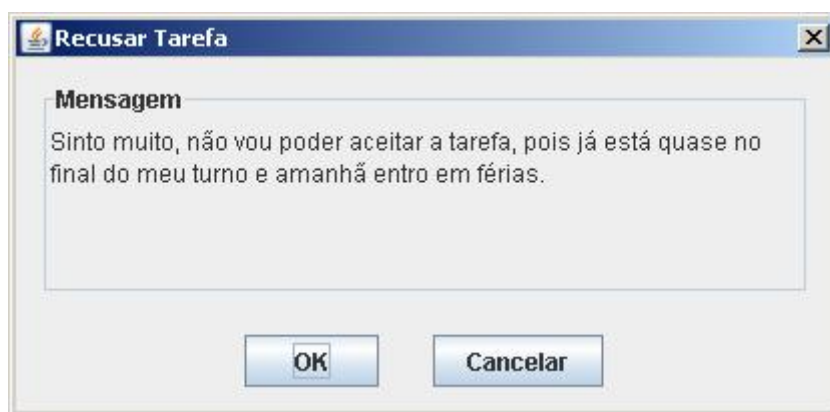


Figura 5.3: Tela para Recusar uma Tarefa Delegada

O Dr. Kroth percebe, neste instante, que apareceu outra tarefa delegada para o seu usuário, mas como já está se aproximando do final do horário do seu turno, ele recusa a tarefa informado os motivos (Figura 5.3). O Dr. Kroth também verifica que existem atividades que ainda não estão finalizadas, então ele as delega para os médicos ou especialidades que julga serem os mais adequados. Não tendo mais nenhuma atividade pendente, o Dr. Kroth está com o sentimento

de dever cumprido, se despede dos colegas do hospital e começa a pensar nos detalhes das suas férias.

5.3 Análise de Desempenho do Serviço de Colaboração

Para possibilitar o suporte à colaboração na projeto ClinicSpace, foram adicionados novos componentes à arquitetura e outros sofreram modificações. Com o intuito de garantir que as alterações realizadas não impactassem negativamente no tempo de resposta das aplicações no ambiente pervasivo, foi realizada uma análise de desempenho do Serviço de Colaboração.

Como parâmetros para a simulação dos testes, foi utilizado um ambiente hospitalar fictício, de médio porte, com um número arbitrário de 1.000 médicos. Foram atribuídas, para cada médico, 10 tarefas ativas e também 10 tarefas delegadas, totalizando, no banco de dados, 10.000 tarefas ativas e 10.000 tarefas delegadas. Para esta análise foi escolhido o método `getActiveTasks(login)`, que retorna as tarefas ativas de um usuário, pois é chamado sempre que o médico interage com o sistema e também porque usa a maioria das modificações realizadas na arquitetura.

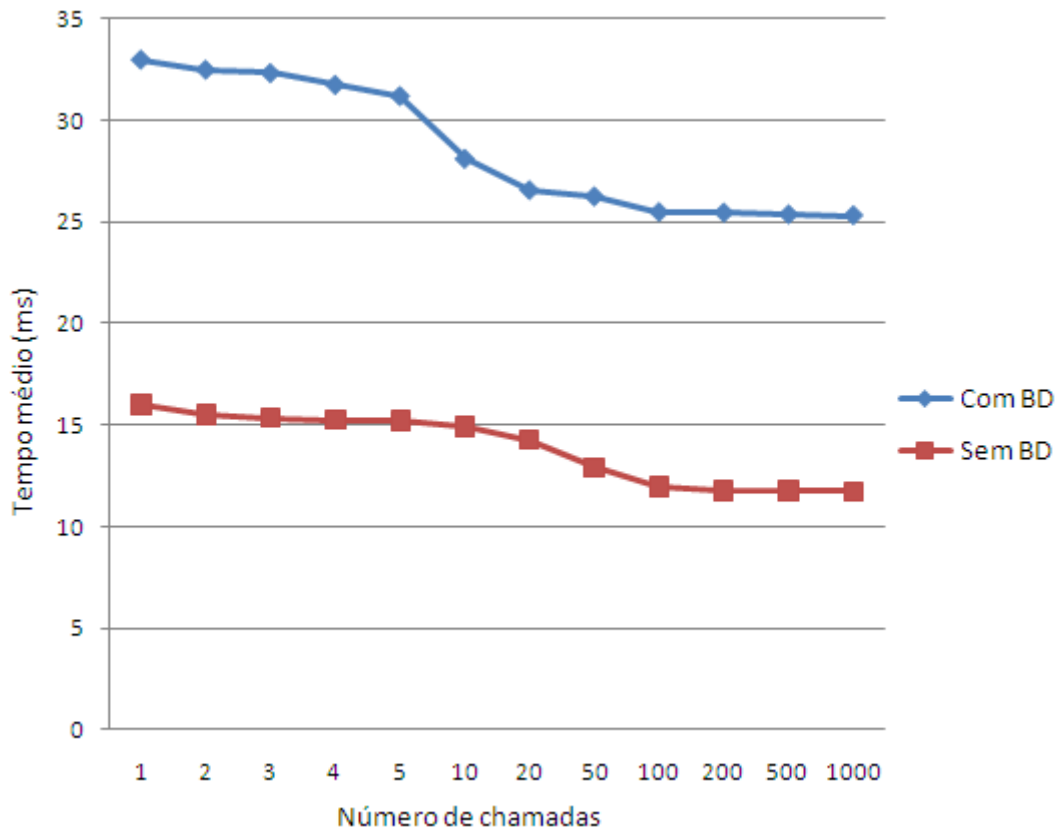


Figura 5.4: Comparativo de desempenho com e sem o banco de dados

A Figura 5.4 apresenta um comparativo do tempo médio (em ms) de chamada do método `getActiveTasks(login)` entre a versão sem as modificações na arquitetura do projeto ClinicSpace, legenda “Sem BD” (em vermelho) e a nova versão desenvolvida, legenda “Com BD” (em azul).

Percebe-se claramente que houve um reflexo significativo no desempenho do sistema e o principal motivo deste impacto negativo foi a introdução da persistência das tarefas no banco de dados. As chamadas para verificação das tarefas ativas de um usuário deixaram de fazer um acesso direto à memória e passaram a acessar os dados através de requisições ao banco de dados.

Com o objetivo de melhorar o tempo de resposta dos acesso às tarefas ativas e delegadas, e, conseqüentemente, no desempenho das aplicações no ambiente pervasivo, foram criadas duas caches no Serviço de Tarefas Ativas, a Cache de Tarefas Ativas e a Cache de Tarefas Delegas. A ideia é manter em memória as tarefas dos médicos após o primeiro acesso ao sistema, evitando novas consultas ao banco de dados. A API utilizada para acessar as caches criadas pode ser vista na Figura 5.5.

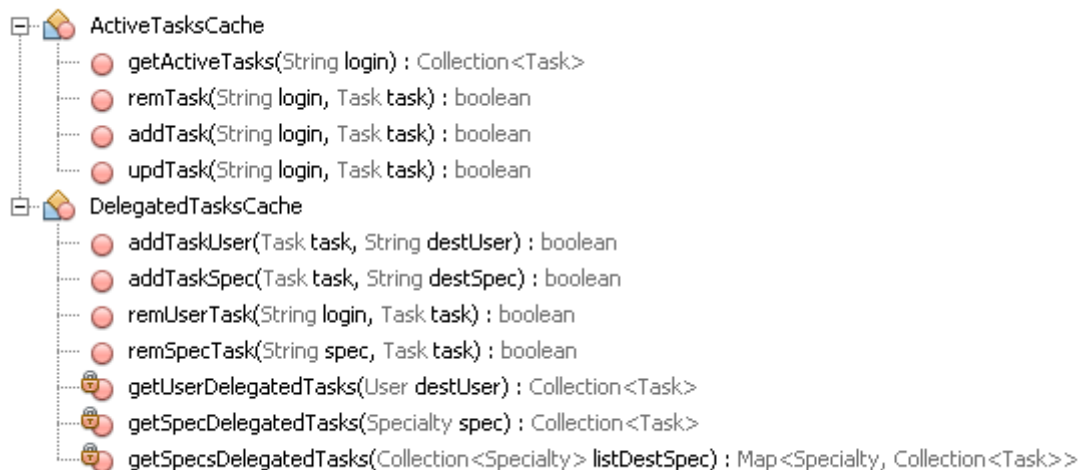


Figura 5.5: API das Caches de Tarefas Ativas e Tarefas Delegadas

As caches foram implementadas utilizando mapa em memória (classe *HashMap* do Java) e funcionam da seguinte forma: quando chega uma requisição solicitando as tarefas ativas de um usuário (`getActiveTasks(login)`), primeiramente, é feita uma busca na cache para verificar se existem os dados do usuário, retornando o que estiver na memória. Caso não tenha, é feita uma consulta no banco de dados e, depois, o resultado é adicionado na cache. Todas as operações que atualizam as tarefas ativas são feitas no banco de dados e na cache, mantendo a

consistência das informações.

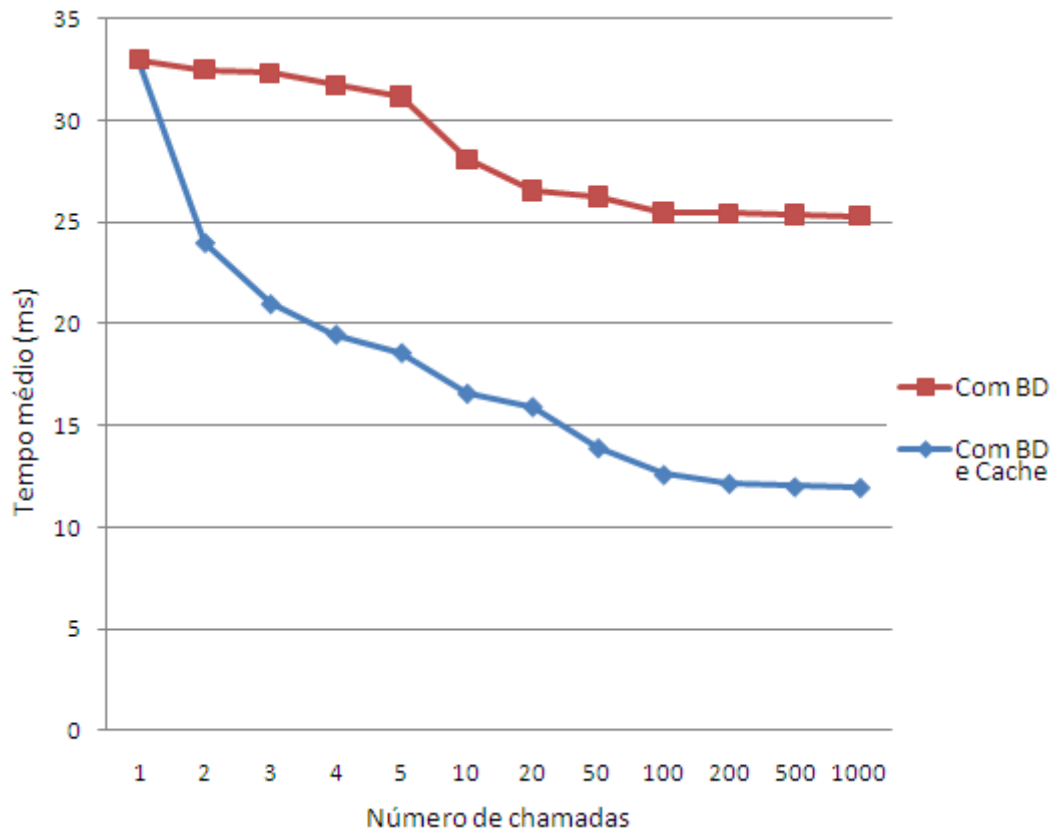


Figura 5.6: Comparativo de desempenho com banco de dados e com as caches

A Figura 5.6 mostra um comparativo do tempo médio (em ms) de chamada do método `getActiveTasks(login)` da nova versão do sistema, com o banco de dados, antes, legenda “Com BD” (em vermelho) e após a introdução das caches, legenda “Com BD e Cache” (em azul).

Inicialmente, o tempo utilizando a versão com as caches fica muito próximo do tempo utilizando somente com o banco de dados. Isso ocorre porque os testes de desempenho foram feitos com as caches vazias, ou seja, na primeira requisição realizada, o tempo tende a ser parecido ao tempo sem a cache, pois a requisição deverá fazer uma consulta ao banco de dados para buscar as informações e popular a cache. Já a partir do segundo acesso às tarefas ativas do mesmo usuário, o tempo tende a ser praticamente muito próximo ao tempo que se tinha antes das modificações na arquitetura, pois a busca passa a ser em memória e não mais ao banco de dados.

Finalmente, foram comparados o tempo médio de execução para a chamada ao método `getActiveTasks(login)` entre a versão original da arquitetura do projeto ClinicSpace,

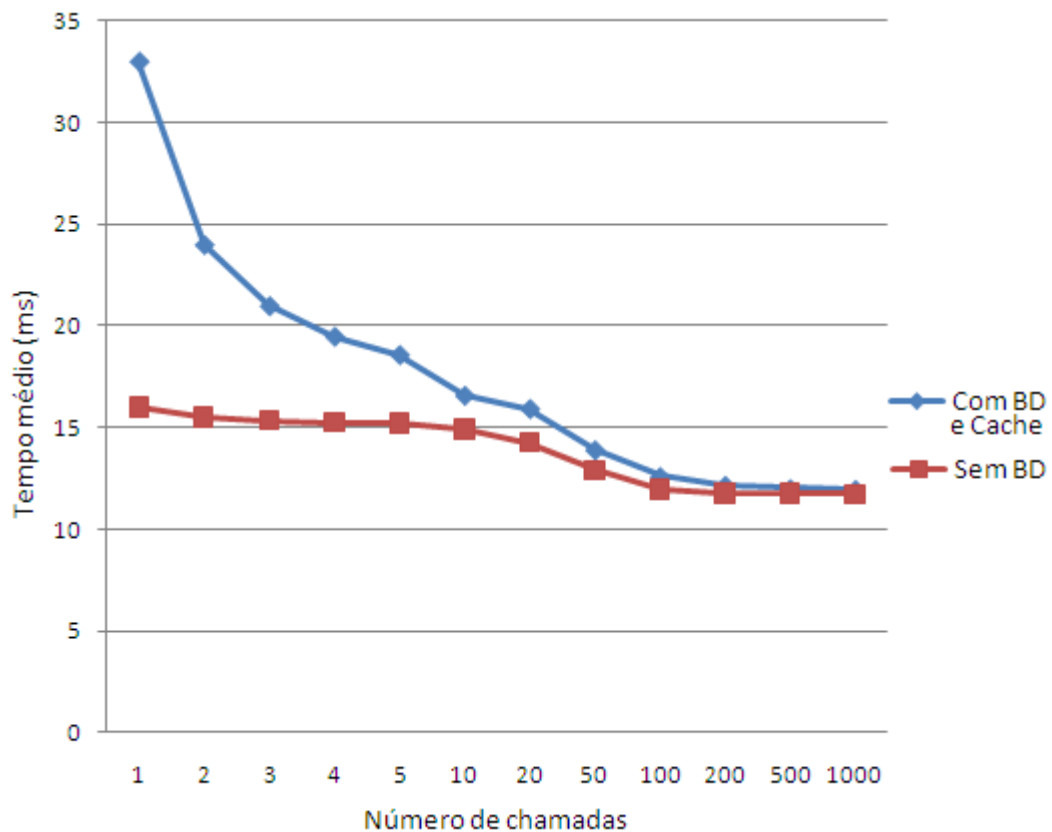


Figura 5.7: Comparativo de desempenho sem banco de dados e com as caches

legenda “Sem BD” (em vermelho) e a versão após a introdução do Serviço de Colaboração e a criação das caches, legenda “Com BD e Cache” (em azul), mostradas na Figura 5.7.

É possível observar que quanto maior é o número de chamadas, mais se aproximam os tempos médios de execução das duas versões. Esse comportamento acontece devido à diferença de desempenho no acesso ao banco de dados, se comparado ao tempo de acesso à memória. Após a introdução das caches, esse tempo somente tem impacto na primeira chamada, quando a cache não tem dados; as próximas chamadas serão feitas com um tempo semelhante ao que se tinha antes das modificações na arquitetura, pois a implementação das caches utilizou os mesmos padrões da versão anterior.

Em um cenário de utilização real, a situação mais comum é a interação dos usuários com o sistema, em grande parte do tempo. Com isso, pode-se concluir que o impacto no desempenho tende a ser mínimo, com os tempos muito próximos dos tempos verificados antes das modificações realizadas na arquitetura do projeto ClinicSpace.

6 CONCLUSÕES

As principais características do trabalho em um hospital são o alto grau de mobilidade, interrupções frequentes das atividades que estão em andamento e a colaboração entre os clínicos de diferentes especialidades médicas.

Existem várias pesquisas que utilizam os conceitos de Computação Ubíqua ou Pervasiva para atender alguma necessidade específica de ambientes hospitalares, mas poucos projetos de infraestrutura computacional buscam atender às principais características do trabalho clínico de forma integrada.

O projeto ClinicSpace tem como objetivo a criação de uma infraestrutura computacional que dê suporte à mobilidade e interrupção das atividades em andamento e que permita aos clínicos a personalização da execução de suas tarefas um ambiente pervasivo, utilizando como base o *middleware* EXEHDA.

Este trabalho apresentou as soluções criadas para dar suporte à colaboração em uma infraestrutura computacional pervasiva no auxílio às atividades clínicas. Foi criado o Serviço de Colaboração, um novo componente integrado à arquitetura do projeto ClinicSpace. Apesar de não possuir mecanismos de colaboração síncrona, como videoconferência, o novo serviço permite a colaboração assíncrona entre os clínicos, onde é possível transferir (delegar) as tarefas que ainda não foram concluídas para outros profissionais. Essa característica é muito frequente em situações de trocas de turno ou para encaminhamentos a colegas de outras especialidades médicas.

Com o objetivo de assegurar que as modificações realizadas nos componentes da arquitetura ClinicSpace manteriam o padrão de desempenho para as aplicações no ambiente pervasivo, foram realizados testes para verificar o tempo de resposta antes e depois das alterações na arquitetura. Após a introdução de um mecanismo de cache para as tarefas ativas e para as tarefas delegadas, chegou-se a conclusão que o impacto no desempenho das aplicações tende a ser mínimo.

6.1 Contribuições

Foi desenvolvido um Gerenciador de Tarefas com a inclusão das novas funcionalidades criadas pelo Serviço de Colaboração. Foi implementado, também, um mecanismo para troca dinâmica de usuários e adicionado o conceito de especialidade médica que poder ser utilizado

para simulações em trabalhos futuros do projeto.

Atualmente, a maioria dos trabalhos que abordam os aspectos colaborativos de ambientes clínicos com o suporte de uma infraestrutura computacional pervasiva tem o foco na colaboração síncrona, com vários esforços para a criação de mecanismos para auxiliar, principalmente, os diagnósticos em equipes espacialmente distribuídas como, por exemplo, funcionalidades para compartilhamento de exames.

Diferentemente de tais abordagens, este trabalho foca nas características assíncronas da colaboração em ambientes hospitalares, quer seja na troca de turno dos clínicos ou nos encaminhamentos de tarefas ainda não concluídas para outro profissional, sempre com o foco na atividade médica.

6.2 Trabalhos Futuros

Como sugestão de trabalho futuro, poderia ser criado no Serviço de Colaboração um mecanismo para troca de mensagens entre os clínicos (de forma semelhante a um correio eletrônico). Essa nova funcionalidade poderia ser implementada de uma forma semelhante à delegação de tarefas (adicionando novos itens na API do Serviço de Colaboração), onde um médico teria a possibilidade de enviar uma mensagem ou arquivo para outro profissional ou especialidade médica.

Além disso, poderia ser criado, integrado ao Serviço de Tarefas Ativas, um controle de usuários ativos (*online*). Essa informações seria utilizada caso o médico queira saber, antes de fazer a delegação de uma tarefa, se o profissional desejado está conectado ou não ao sistema pervasivo.

Outro trabalho que poderia ser desenvolvido é a integração do Serviço de Colaboração com o Serviço de Inferência de Tarefas ou Serviço de Predição de Tarefas, ou seja, registrar o histórico de delegação de tarefas e as informações de contexto associadas às tarefas no momento da delegação; com isso, poderiam ser feitas inferências acerca das delegações de tarefas do usuário, bem como sobre a aceitação e recusa de tarefas delegadas por determinados usuários ou especialidades médicas.

Devido à personalização das tarefas pelo clínico, quem for receber a tarefa pode ter outra maneira particular de realizar a mesma atividade. Um possível trabalho futuro poderia ser resolver a transição das formas diferentes dos médicos ao realizar as suas atividades na delegação de tarefas.

O projeto ClinicSpace ainda está em desenvolvimento, um próximo passo da pesquisa é a realização de uma avaliação do sistema pelos clínicos, focando os aspectos de usabilidade em um ambiente hospitalar.

6.3 Publicações

As publicações feitas até o momento relativas ao trabalho são:

SIRC 2010 - “Comparativo entre Projetos de Infraestrutura Computacional Pervasiva para Ambientes Clínicos”. In: IX Simpósio de Informática da Região Centro do Rio Grande do Sul. <http://www.sirc.unifra.br/artigos2010/1.pdf>

SBCUP 2011 - “Suporte à Colaboração em uma Infraestrutura Computacional Pervasiva para Auxílio às Atividades Clínicas”. In: XXXI Congresso da Sociedade Brasileira de Computação - III Simpósio Brasileiro de Computação Ubíqua e Pervasiva (CSBC 2011 - SBCUP).

JPC 2011 - “Serviço de Colaboração para a Arquitetura ClinicSpace”. In: X Congreso Internacional de la Sociedad Peruana de Computación - X Jornada Peruana de Computación. (aceito)

REFERÊNCIAS

- AUGUSTIN, I.; LIMA, J. C. D.; YAMIN, A. Computação Pervasiva: como programar aplicações. In: X SIMPOSIO BRASILEIRO DE LINGUAGENS DE PROGRAMAÇÃO, 2006, Itatiaia, RJ. **Anais...** [S.l.: s.n.], 2006.
- AUGUSTIN, I.; YAMIN, A.; GEYER, C. F. R. Managing the follow-me semantics to build large-scale pervasive applications. In: MIDDLEWARE FOR PERVASIVE AND AD-HOC COMPUTING, 3., 2005, New York, NY, USA. **Proceedings...** ACM, 2005. p.1–8. (MPAC '05).
- BARDRAM, J. E. Hospitals of the Future - Ubiquitous Computing support for Medical Work in Hospitals. In: UBIHEALTH 2003: THE 2ND INTERNATIONAL WORKSHOP ON UBIQUITOUS COMPUTING FOR PERVASIVE HEALTHCARE APPLICATIONS., 2003. **Anais...** [S.l.: s.n.], 2003.
- BARDRAM, J. E. Activity-based computing: support for mobility and collaboration in ubiquitous computing. **Personal and Ubiquitous Computing**, [S.l.], v.9, p.312–322, 2005. 10.1007/s00779-004-0335-2.
- BARDRAM, J. E. Activity-based computing for medical work in hospitals. **ACM Trans. Comput.-Hum. Interact.**, New York, NY, USA, v.16, p.10:1–10:36, June 2009.
- BARDRAM, J. E.; CHRISTENSEN, H. B. **Supporting Pervasive Collaboration in Healthcare - An Activity-Driven Computing Infrastructure**. Aarhus, Denmark: Centre for Pervasive Computing, 2004. Available from <http://www.pervasive.dk/publications>. (CfPC? Technical Report 2004-PB-63).
- BARDRAM, J. E.; CHRISTENSEN, H. B. Pervasive Computing Support for Hospitals: an overview of the activity-based computing project. **IEEE Pervasive Computing**, Piscataway, NJ, USA, v.6, p.44–51, January 2007.
- BARSOTTINI, C. G. N. **Estudo Etnográfico do Padrão de Colaboração Entre Médicos por Meio do Prontuário do Paciente**. 2007. Tese (Doutorado) — Universidade Federal de São Paulo.

CABITZA, F.; LOCATELLI, M. P.; SARINI, M.; SIMONE, C. CASMAS: supporting collaboration in pervasive environments. In: FOURTH ANNUAL IEEE INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING AND COMMUNICATIONS, 2006, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2006. p.286–295.

FERRAZ, C.; DINIZ, J. Supporting the Ubiquitous Doctor. **International Journal of Advanced Pervasive and Ubiquitous Computing (IJAPUC)**, [S.l.], v.2, p.148–173, 2010.

FERREIRA, G.; AUGUSTIN, I.; LIBRELOTTO, G. R.; SILVA, F. L. da; YAMIN, A. C. Middleware for management of end-user programming of clinical activities in a pervasive environment. In: WORKSHOP ON MIDDLEWARE FOR UBIQUITOUS AND PERVASIVE SYSTEMS, 2009., 2009, New York, NY, USA. **Proceedings...** ACM, 2009. p.7–12. (WMUPS '09).

FERREIRA, G. G. L. **Adicionando ao Middleware EXEHDA o Suporte a Aplicações Orientadas a Atividades Humanas Cotidianas**. 2009. Dissertação (Mestrado) — Programa de Pós-Graduação em Informática, Universidade Federal de Santa Maria, Santa Maria, RS, Brazil.

GARLAN, D.; SIEWIOREK, D.; SMAILAGIC, A.; STEENKISTE, P. Project Aura: toward distraction-free pervasive computing. **IEEE Pervasive Computing**, Piscataway, NJ, USA, v.1, p.22–31, April 2002.

JHA, A.; DESROCHES, C.; CAMPBELL, E.; DONELAN, K.; RAO, S.; FERRIS, T.; SHIELDS, A.; ROSENBAUM, S.; BLUMENTHAL, D. The Use of Electronic Health Records in U.S. Hospitals. **New England Journal of Medicine**, [S.l.], v.360, n.16, p.1628–1638, 2009.

KROTH, M.; AUGUSTIN, I. Comparativo entre Projetos de Infraestrutura Computacional Pervasiva para Ambientes Clínicos. In: IX SIMPÓSIO DE INFORMÁTICA DA REGIÃO CENTRO/RS, 2010, Santa Maria/RS, Brazil. **Anais...** [S.l.: s.n.], 2010.

KROTH, M.; AUGUSTIN, I.; MACHADO, A. Suporte à Colaboração em uma Infraestrutura Computacional Pervasiva para Auxílio às Atividades Clínicas. In: XXXI CSBC - III SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO UBÍQUA E PERVASIVA, 2011, Natal, Brazil. **Anais...** [S.l.: s.n.], 2011.

MACHADO, A. **Associação do Contexto de Interesse do Usuário às Atividades Clínicas na**

Arquitetura ClinicSpace. 2010. Dissertação (Mestrado) — Programa de Pós-Graduação em Informática, Universidade Federal de Santa Maria, Santa Maria, RS, Brazil.

MILLER, R. H.; SIM, I. Physicians' use of electronic medical records: barriers and solutions. **Health Affairs**, [S.l.], v.23, n.2, p.116–126, 2004.

OMG Unified Modeling LanguageTM (OMG UML). Version 2.3, Release date: May 2010, <http://www.omg.org/spec/UML/2.3/Infrastructure/PDF/>.

RANGANATHAN, A.; CAMPBELL, R. H. Supporting Tasks in a Programmable Smart Home. In: 2005, Magog, Canada. **Anais...** [S.l.: s.n.], 2005.

RIZZETTI, T. Um Ambiente de Contexto Personalizado e Orientado a Tarefas na Arquitetura ClinicSpace. 2009. Dissertação (Mestrado) — Programa de Pós-Graduação em Informática, Universidade Federal de Santa Maria, Santa Maria, RS, Brazil.

SAIN, M.; LEE, H.; CHUNG, W.-Y. Middleware for ubiquitous Healthcare Information system. In: ADVANCED COMMUNICATION TECHNOLOGY, 2009. ICACT 2009. 11TH INTERNATIONAL CONFERENCE ON, 2009. **Anais...** [S.l.: s.n.], 2009. v.03, p.2325–2328.

SATYANARAYANAN, M. Mobile Information Access. **IEEE Personal Communications**, [S.l.], v.3, p.26–33, 1996.

SATYANARAYANAN, M. The evolution of Coda. **ACM Trans. Comput. Syst.**, New York, NY, USA, v.20, p.85–124, May 2002.

SILVA, F. L. ClinicSpace - Modelagem de uma ferramenta-piloto para definição de tarefas clínicas em um ambiente de computação pervasiva baseado em tarefas e direcionado ao usuário-final. 2009. Dissertação (Mestrado) — Programa de Pós-Graduação em Informática, Universidade Federal de Santa Maria, Santa Maria, RS, Brazil.

SOUSA, J. P.; GARLAN, D. Aura: an architectural framework for user mobility in ubiquitous computing environments. In: IFIP 17TH WORLD COMPUTER CONGRESS - TC2 STREAM / 3RD IEEE/IFIP CONFERENCE ON SOFTWARE ARCHITECTURE: SYSTEM DESIGN, DEVELOPMENT AND MAINTENANCE, 2002, Deventer, The Netherlands, The Netherlands. **Proceedings...** Kluwer: B.V., 2002. p.29–43. (WICSA 3).

SOUZA, M. V. B. **Inferência de Atividades Clínicas na Arquitetura ClinicSpace a partir de Propriedades do Contexto**. 2010. Dissertação (Mestrado) — Programa de Pós-Graduação em Informática, Universidade Federal de Santa Maria, Santa Maria, RS, Brazil.

TENTORI, M.; FAVELA, J. Activity-Aware Computing for Healthcare. **IEEE Pervasive Computing**, Piscataway, NJ, USA, v.7, p.51–57, April 2008.

VICENTINI, C. F. **PEHS - Arquitetura de um Sistema Pervasivo de Informação em Saúde Orientado às Atividades Personalizadas pelo Usuário Clínico**. 2010. Dissertação (Mestrado) — Programa de Pós-Graduação em Informática, Universidade Federal de Santa Maria, Santa Maria, RS, Brazil.

VICENTINI, C. F.; MACHADO, A.; FERREIRA, G. L.; LORENZI, F.; AUGUSTIN, I. PEHS - Arquitetura de um Sistema de Informação Pervasivo para Auxílio às Atividades Clínicas. **Revista Brasileira de Computação Aplicada**, [S.l.], v.2, n.2, p.69–80, Setembro 2010.

WEISER, M. The computer for the 21st century. **Scientific American**, [S.l.], September 1991.

YAMIN, A.; AUGUSTIN, I.; SILVA, L. C. da; REAL, R. A.; FILHO, A. E. S.; GEYER, C. F. R. EXEHDA: adaptive middleware for building a pervasive grid environment. In: PROCEEDING OF THE 2005 CONFERENCE ON SELF-ORGANIZATION AND AUTONOMIC INFORMATICS (I), 2005, Amsterdam, The Netherlands, The Netherlands. **Anais...** IOS Press, 2005. p.203–219.