

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

**UM SERVIÇO DE PERSISTÊNCIA DE CONTEXTO E
SELEÇÃO CONTEXTUALIZADA DE DOCUMENTOS
PARA A ARQUITETURA CLINICSPACE**

DISSERTAÇÃO DE MESTRADO

Vinícius Maran

Santa Maria, RS, Brasil

2011

**UM SERVIÇO DE PERSISTÊNCIA DE CONTEXTO E
SELEÇÃO CONTEXTUALIZADA DE DOCUMENTOS PARA
A ARQUITETURA CLINICSPACE**

por

Vinícius Maran

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

Orientadora: Prof.^a Dr.^a Iara Augustin
Co-Orientadora: Prof.^a Dr.^a Deise de Brum Saccol

Santa Maria, RS, Brasil

2011

M311s Maran, Vinícius
Um serviço de persistência de contexto e seleção contextualizada de documentos para a arquitetura ClinicSpace / por Vinícius Maran. – 2011.
96 f. ; il. ; 30 cm

Orientador: Iara Augustin
Coorientador: Deise de Brum Saccol
Dissertação (mestrado) – Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Informática, RS, 2011

1. Computação ubíqua 2. Computação na saúde 3. Ontologia 4. Contexto
I. Augustin, Iara II. Saccol, Deise de Brum III. Título.

CDU 004

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

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

**UM SERVIÇO DE PERSISTÊNCIA DE CONTEXTO E SELEÇÃO
CONTEXTUALIZADA DE DOCUMENTOS PARA A ARQUITETURA
CLINICSPACE**

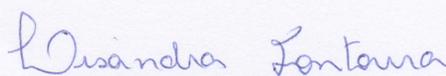
elaborada por
Vinícius Maran

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

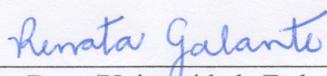
COMISSÃO EXAMINADORA:



Iara Augustin, Dra.
(Presidente/Orientadora)



Lisandra Manzoni Fontoura, Dra. (Universidade Federal de Santa Maria)



Renata de Matos Galante, Dra. (Universidade Federal do Rio Grande do Sul)

Santa Maria, 19 de dezembro de 2011.

AGRADECIMENTOS

A Deus por estar presente em todos os momentos da minha vida.

Aos meus pais, minha irmã e a toda minha família, por acreditarem nos meus sonhos, me fortalecer a cada novo passo e contribuir imensamente para minhas conquistas.

A minha namorada Francine, pelo companheirismo, cumplicidade, amor, apoio e por sonhar junto comigo.

Às Prof^{as} Dr^{as} Iara Augustin e Deise de Brum Saccol, pela oportunidade de trabalhar ao seu lado, pela amizade, confiança, carinho, disponibilidade e por partilhar seus conhecimentos e experiências comigo durante essa caminhada.

RESUMO

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

UM SERVIÇO DE PERSISTÊNCIA DE CONTEXTO E SELEÇÃO CONTEXTUALIZADA DE DOCUMENTOS PARA A ARQUITETURA CLINICSPACE

Autor: Vinícius Maran
Orientadora: Iara Augustin
Co-Orientadora: Deise de Brum Saccol
Santa Maria, 19 de dezembro de 2011

Os sistemas de saúde atuais sofrem de uma alta taxa de rejeição por parte dos profissionais clínicos que utilizam estes sistemas, pois é necessário que os usuários forneçam informações de forma explícita e constante. Desta forma, um dos maiores desafios para sistemas de saúde pervasivos é encontrar uma forma de utilizar informações de contexto do ambiente de um modo simples e funcional entre diferentes sistemas computacionais. Na literatura, ontologias são frequentemente utilizadas para a representação de contexto e possuem um importante papel em sistemas pervasivos se utilizadas em conjunto com formas de persistência e recuperação de informações de contexto. Para resolver o problema citado, está em fase de desenvolvimento uma arquitetura chamada ClinicSpace que tem como foco fornecer ajuda aos profissionais clínicos na execução de suas tarefas diárias, através de conceitos definidos na computação ubíqua, os quais permitem ao sistema se adaptar constantemente ao usuário e as suas necessidades. Para que isto ocorra, é necessário que o sistema seja capaz de (i) armazenar informações de contexto e de (ii) apresentar os documentos frequentemente utilizados pelos profissionais clínicos, de forma adaptada ao contexto das tarefas executadas, e disponíveis a qualquer hora e em qualquer lugar (pervasividade). Este trabalho descreve o processo de desenvolvimento de um serviço integrado à arquitetura ClinicSpace, o qual fornece suporte à utilização de dados de contexto e documentos clínicos de uma forma distribuída, e também permite a seleção contextualizada de documentos clínicos, utilizando dados de contexto no momento de consulta a informações clínicas.

Palavras-chave: computação ubíqua na saúde, sistemas de informação clínica, seleção contextualizada, persistência de contexto.

ABSTRACT

Master's Dissertation
Post-Graduate Program in Informatics
Federal University of Santa Maria

A Service for Context Persistence and Context-Based Selection of Documents for ClinicSpace Architecture

Author: Vinícius Maran
Advisor: Iara Augustin
Co-Advisor: Deise de Brum Saccol
Santa Maria, December 19, 2011

Actual healthcare systems suffer from a high rate of rejection by the physicians who use these systems, because it is necessary that the users explicitly provide information constantly to these systems. This way, one of the biggest challenges for pervasive healthcare systems is to find a way to use context information of the environment in a simple and functional form between different computer systems. In the literature, ontologies are frequently used for the representation of context and have an important role in pervasive systems if used together with forms of persistence and retrieval of context information. To solve the mentioned problem, is under development an architecture called ClinicSpace which focuses on providing assistance to physicians in performing their daily tasks, using concepts defined in ubiquitous computing, which allows the system to adapt constantly to the user and their needs. To make this, it is necessary that the system be capable of (i) store information of context and (ii) provide the documents frequently used by physicians, in an adapted way to the context of the tasks performed, and available anytime and anywhere (pervasiveness). This work describes the process of developing an integrated service ClinicSpace to architecture, which supports the use of context data and clinical documents in a distributed manner, and also allows the contextualized selection of clinical documents, using data from the context at the time of clinical information query.

Keywords: ubiquitous computing, healthcare, contextualized selection, context persistence.

LISTA DE FIGURAS

Figura 1 – Exemplo de arquivo OWL	20
Figura 2 - Exemplo de código SWRL	20
Figura 3 – Arquitetura ClinicSpace	28
Figura 4 – Cenário de utilização do sistema.....	31
Figura 5 – Componentes internos do serviço de persistência de contexto e seleção contextualizada de documentos	32
Figura 6 – Modelagem de elementos de contexto (MACHADO, 2010).....	33
Figura 7 – Processo de desenvolvimento proposto pelo guia <i>Ontology Development 101</i> (RAUTENBERG et al, 2008).....	34
Figura 8 – Representação de uma tarefa na ontologia proposta para a arquitetura ClinicSpace	35
Figura 9 – Atividades clínicas comuns em ambientes hospitalares (BARDRAM; CHRISTENSEN, 2007).....	36
Figura 10 – Ontologia proposta para a representação de elementos de contexto.....	39
Figura 11 – Exemplo de representação das informações clínicas de um documento clínico...	41
Figura 12 – Modelagem de documentos clínicos na ontologia	42
Figura 13 – Arquitetura básica da arquitetura SemantiCouch.....	45
Figura 14 – Diagrama de classes do pacote <i>Comm</i>	46
Figura 15 – Diagrama de classes do pacote <i>OO</i>	47
Figura 16 – Estrutura de armazenamento de um arquivo OWL no banco de dados	49
Figura 17 – Documentos em uma instância do banco de dados CouchDB.....	51
Figura 18 – Processo de conversão de uma instância do banco de dados para um arquivo OWL.....	52
Figura 19 – Métodos definidos na classe <i>OOWLMModel</i> para a consulta de informações da ontologia	54
Figura 20 – (a) método de consulta de entidade; (b) exemplo de chamada do método	55
Figura 21 - Replicação da ontologia pela arquitetura SemantiCouch	55
Figura 22 - Replicação das definições da ontologia pela arquitetura SemantiCouch	56
Figura 23 – Exemplo de replicação de documentos pelo banco de dados <i>CouchDB</i>	56
Figura 24 – Exemplo de configuração da replicação de instâncias pelo utilitário <i>Futon</i>	57
Figura 25 – Associação de informações de contexto ao paciente através da IETC (MACHADO, 2010).....	59
Figura 26 – Protótipo do módulo de inserção de documentos e associação de contexto	60
Figura 27 – Processo de Inserção de um documento no sistema.....	63
Figura 28 – Processo de consulta de um documento no sistema.....	64
Figura 29 - Código SQWRL da Consulta 1	66
Figura 30 – Consulta 2.....	66
Figura 31 – Resultados da Consulta 3	67
Figura 32 - Código SWRL da inferência 1	67
Figura 33 - Código SWRL da inferência 2.....	68
Figura 34 – Interface de testes da arquitetura <i>SemantiCouch</i>	69
Figura 35 – Resultados dos testes de inserção de ontologias	69
Figura 36 - Consulta de dados na arquitetura SemantiCouch	71
Figura 37 - Operações realizadas em um indivíduo na ontologia	71
Figura 38 – Associação de informações de contexto ao paciente através da IETC (MACHADO, 2010).....	73

Figura 39 – Indivíduos criados para o teste de seleção de documentos	74
Figura 40 – Exemplo de exame clínico fictício criado em diferentes versões em relação a resolução de tela	75
Figura 41 – Comparação entre os tempos de carregamento do sistema com as alterações realizadas	76
Apêndice A. Figura 1 - Demonstração da IETC para a edição do fluxo de trabalho (MACHADO, 2010).....	89
Apêndice A. Figura 2 - Representação-exemplo do relacionamento entre as tarefas e subtarefas (MACHADO, 2010).....	90
Apêndice A. Figura 3 - Diagrama de possíveis estados de uma tarefa	91
Apêndice A. Figura 4 - Arquitetura ClinicSpace	92
Apêndice A. Figura 5 - Interface de Edição de Tarefas (SILVA, 2009).....	94
Apêndice A. Figura 6 - Ontologia utilizada para a representação de contexto (MACHADO, 2010).....	97

LISTA DE TABELAS

Tabela 1 – Comparativo entre formas de representação de contexto (MACHADO, 2010).....	18
Tabela 2 – Comparativo de soluções para a persistência de ontologias.....	23
Tabela 3 – Comparativo entre projetos de <i>healthcare</i> e a arquitetura ClinicSpace	26
Tabela 4 – Lista de documentos frequentemente utilizados no ambiente hospitalar	40
Tabela 5 – Relação de <i>flags</i> utilizadas para nomear documentos no banco de dados.....	51
Apêndice A. Tabela 1 - Conjunto de tarefas clínicas mínimas (SILVA, 2009).....	90

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Program Interface</i>
CCR	<i>Continuity of Care Record Standard</i>
EHS	<i>Electronic Healthcare System</i>
EXEHDA	<i>Execution Environment for Highly Distributed Applications</i>
HL7	<i>Health Level Seven International Standard</i>
IETC	Interface de Edição de Tarefas e Contexto
OWL	<i>Web Ontology Language</i>
pEHS	<i>Pervasive Electronic HealthCare Systems</i>
RDF	<i>Resource Description Framework</i>
RDQL	<i>RDF Data Query Language</i>
SAT	Serviço de Acesso a Tarefas
SATA	Serviço de Acesso a Tarefas Ativas
SCT	Serviço de Contexto de Tarefas
SemantiCouch	<i>Semantic Data in CouchDB Database</i>
SGDT	Subsistema de Gerenciamento Distribuído de Tarefas
SGT	Serviço de Gerenciamento de Tarefas
SI	Serviço de Inferência
SPC	Serviço de Persistência de Contexto
SQL	<i>Simple Query Language</i>
SQWRL	<i>Semantic Query-Enhanced Web Rule Language</i>
SWRL	<i>Semantic Web Rule Language</i>
URI	<i>Uniform Resource Identifier</i>
XML	<i>Extensible Markup Language</i>
W3C	<i>World Wide Web Consortium</i>

SUMÁRIO

Lista de Figuras	7
Lista de Tabelas	9
Lista de Abreviaturas e Siglas	10
Sumário	11
1 Introdução.....	13
1.1 Problema e Objetivos	14
1.2 Organização do Texto.....	15
2 Computação Ubíqua Baseada em Contexto	16
2.1 Computação Ubíqua	16
2.2 Computação Consciente do Contexto	17
2.2.1 Representação de Contexto.....	18
2.2.2 Persistência de Contexto	22
2.3 Computação Ubíqua na Saúde	24
2.3.1 Projetos de Computação Ubíqua e Saúde.....	25
2.3.2 Projeto ClinicSpace	26
2.3.2.1 Arquitetura	27
2.3.2.2 Problemas	28
3 Modelagem de Conhecimento e Contexto Para a Arquitetura ClinicSpace....	30
3.1 Cenário de Uso	30
3.2 Representação do Domínio	32
3.2.1 Representação de Elementos de Contexto	34
3.2.2 Representação do Conhecimento Clínico	40
4 Um Serviço de Persistência de Contexto e Seleção Contextualizada de Documentos	43
4.1 Persistência de Contexto – Arquitetura SemantiCouch.....	43
4.1.1 Definição da Arquitetura SemantiCouch.....	43
4.1.2 Processo de Conversão de Formatos de Arquivos.....	48
4.1.3 Inserção e Consulta.....	53
4.1.4 Distribuição de Informações.....	55
4.2 Seleção Contextualizada de Documentos Clínicos.....	57
4.2.1 Alterações na IETC.....	58
4.2.2 Associação de Informações de Contexto	60
4.2.3 Versionamento	61
4.2.4 Processo de Inserção de Documentos	62

4.2.5 Processo de Consulta de Documentos	63
5 Testes Realizados e Caso de Uso	65
5.1 Testes de Representação de Contexto	65
5.2 Testes de Armazenamento de Contexto.....	68
5.3 Caso de Uso	71
6 Conclusões.....	77
6.1 Contribuições	77
6.2 Trabalhos Futuros	78
6.3 Publicações	79
Referências Bibliográficas.....	80
Apêndice A – Arquitetura ClinicSpace	86

1 INTRODUÇÃO

O paradigma conhecido como computação ubíqua, proposto por Mark Weiser (1991), define um espaço onde computadores e dispositivos digitais, dos mais diversos tipos, estão totalmente integrados ao usuário e ao seu ambiente.

Deste modo, a computação ubíqua tem como principal objetivo fazer com que a computação se torne onipresente, ou seja, fazer com que ela esteja sempre presente em todos os ambientes de forma invisível (não no foco da atenção do usuário) e transparente, sem que o usuário tenha consciência da existência dos recursos computacionais no ambiente em que ele se encontra (WEISER, 1991).

Pesquisas em sistemas de saúde, como o hospital do futuro (BARDRAM; CHRISTENSEN, 2007), propõem a utilização de tecnologias definidas pela computação ubíqua. Tendo como principal característica o constante monitoramento do ambiente e das pessoas que estão inseridas nele, e a adaptação automática do comportamento do sistema ao contexto. Desta forma, estes sistemas criam ambientes inteligentes, que são reativos e proativos (FERREIRA et al, 2009).

Observou-se em pesquisas realizadas recentemente (JHA et al, 2009) (BARDRAM; CHRISTENSEN, 2007) que sistemas hospitalares sofrem de um alto grau de rejeição por parte dos profissionais clínicos que utilizam estes sistemas.

Os principais motivos desta rejeição são as características dinâmicas inerentes às atividades clínicas, onde o profissional clínico precisa locomover-se constantemente no ambiente hospitalar, trocar informações com outros profissionais, diagnosticar e prescrever rapidamente com base em informações de saúde do paciente e trabalhar com situações de emergência, em um ambiente restrito quanto a recursos e ou informações (VICENTINI, 2010).

Deste modo, acredita-se que a construção de uma ferramenta baseada na computação ubíqua, que equilibre a personalização na modelagem de tarefas clínicas e a pró-atividade de sistemas ubíquos, diminua este fator de rejeição aos sistemas de informação na saúde (JHA et al, 2009).

O projeto ClinicSpace, em desenvolvimento na Universidade Federal de Santa Maria – no laboratório de pesquisas GMob (2011), possui como proposta analisar soluções, modeladas na forma de uma ferramenta-protótipo, de auxílio às atividades dos profissionais de saúde

baseado na consciência de contexto, mobilidade e personalização das atividades.

O diferencial deste projeto é que o profissional pode personalizar suas tarefas no sistema, informando ao sistema como ele as realiza; além disso, estas tarefas são sensíveis ao contexto, adaptando-se às informações do ambiente, garantindo acesso às informações relevantes na medida em que são necessárias (VICENTINI, 2010).

1.1 Problema e Objetivos

O projeto ClinicSpace tem várias frentes de estudos das questões envolvidas (ver Capítulo 2). Neste trabalho, abordam-se as questões relativas ao armazenamento contextualizado de informações clínicas.

Uma das principais características dos sistemas ubíquos é a sensibilidade ao contexto. Ela permite que sistemas reconheçam alterações de comportamento em elementos do ambiente e se adaptem a estas mudanças, seja na forma de funcionamento ou na forma de apresentação do conteúdo (BARDRAM, 2003).

Atualmente, na área de *healthcare*, não se dispõe de uma ferramenta que, a partir da modelagem de contexto, armazene e recupere informações de forma contextualizada (com base no contexto corrente), facilitando a recuperação e utilização destas por sistemas ubíquos.

Portanto, são necessárias soluções que possibilitem a gravação de dados de contexto de uma forma distribuída entre nós de uma rede pervasiva e a recuperação destes dados, tanto de forma simples - apenas como uma consulta de elementos de contexto, quanto de formas mais complexas - consultando informações de contexto semelhantes as informadas. Além disso, deve-se contemplar um comportamento adaptativo, tanto no código de execução quanto no conteúdo clínico, conforme se apresenta o contexto de uma determinada tarefa.

O principal objetivo deste trabalho é modelar um serviço que, em conjunto com a arquitetura ClinicSpace, permita: (a) a persistência e recuperação de informações de contexto, constantemente utilizadas por sistemas ubíquos e, (b) a seleção de documentos baseada em informações de contexto de acordo com as requisições feitas pelos serviços da arquitetura ClinicSpace.

Como metodologia para atingir esse objetivo, buscou-se: (i) estudar as formas de representação de contexto e ferramentas para a persistência de contexto; (ii) pesquisar sobre elementos de contexto que possam ser utilizados na arquitetura ClinicSpace em um ambiente

hospitalar pervasivo; (iii) propor uma arquitetura de persistência de dados de contexto de uma forma distribuída; (iv) propor um serviço que integrado a arquitetura ClinicSpace seja capaz de selecionar documentos de acordo com o contexto informado no momento da consulta de informações; (v) testar a implementação das soluções propostas; (vi) desenvolver um estudo de caso, para demonstrar a viabilidade do serviço proposto.

1.2 Organização do Texto

O restante da dissertação é apresentado com a estrutura dada a seguir. O Capítulo 2 apresenta um estudo sobre computação ubíqua e suas aplicações na área de saúde, formas de representação de contexto, uma comparação com trabalhos relacionados e a arquitetura ClinicSpace. O Capítulo 3 apresenta a modelagem ontológica realizada neste trabalho para representar elementos de contexto e informações clínicas para a arquitetura ClinicSpace. O Capítulo 4 detalha a proposta do serviço de persistência de contexto e da seleção contextualizada de documentos. O Capítulo 5 descreve a realização de testes na implementação do serviço proposto e um cenário de uso do serviço de seleção contextualizada de conteúdo. O Capítulo 6 apresenta as conclusões, contribuições deste trabalho e as sugestões de trabalhos futuros.

2 COMPUTAÇÃO UBÍQUA BASEADA EM CONTEXTO

Neste capítulo, abordam-se as questões relacionadas à computação ubíqua e computação ciente de contexto, apresentando conceitos importantes para o domínio destes temas e projetos relacionados a esta área, principalmente em projetos na área de saúde.

2.1 Computação Ubíqua

Em 1991, Mark Weiser (1991) publicou um artigo que descrevia a proposta chamada computação ubíqua, nesta proposta, ele descrevia uma previsão de como o futuro da computação seria e como ela se comportaria.

No futuro descrito por Weiser (1991), os ambientes seriam monitorados constantemente por sensores, espalhados em objetos de uso diário das pessoas e no ambiente de uma forma geral. Este monitoramento constante das atividades das pessoas e as mudanças do ambiente fariam com que a computação conseguisse se adaptar às necessidades das pessoas.

Para que a computação ubíqua atenda as necessidades dos usuários, assume-se que (WEISER, 1991): (i) é necessária a existência de um ambiente computacional invisível e transparente, com métodos intuitivos para que os usuários possam interagir com a computação sem precisar de conhecimentos específicos na área, como ocorre hoje; e (ii) o ambiente deve identificar quem está inserido nele, e assim propor recursos para atendê-lo de forma personalizada.

Através destas características, a computação se tornaria onipresente, ou seja, presente em todos os ambientes onde o usuário estivesse mesmo que este não soubesse ou não conseguisse perceber a presença da computação envolvida em suas atividades.

Neste momento, o foco das pesquisas relacionadas a computação ubíqua se concentra na construção de modelos que, em conjunto, definem como funcionarão os sistemas ubíquos em determinadas áreas (AUGUSTIN; LIMA; YAMIN, 2006).

A criação de novos protocolos e tecnologias de comunicação e a melhoria nas infra-estruturas de redes sem fio permitem o desenvolvimento de projetos e protótipos de aplicações da computação pervasiva em diversas áreas; entre elas, destacam-se as áreas da

saúde e emergências - *smart hospital*; casas virtuais - *smart home*; educação - *pervasive learning*; entretenimento e segurança de residências - *home security* (AUGUSTIN; LIMA; YAMIN, 2006).

Para a criação de ambientes inteligentes provenientes da utilização de sistemas ubíquos, o contexto destes ambientes tem uma importância significativa, pois este determina como os sistemas se comportarão no auxílio aos usuários (AUGUSTIN; LIMA; YAMIN, 2006). A próxima seção trata de contexto de uma forma geral, tanto em relação às teorias relacionadas com modelagem e reconhecimento quanto em relação a pesquisas recentes que envolvem a utilização de contexto na computação ubíqua.

2.2 Computação Consciente do Contexto

Reconhecer e tratar as informações dos ambientes onde os usuários da computação estão inseridos faz com que os sistemas sejam capazes de se adaptar às necessidades específicas destes usuários. Para que isto aconteça, as aplicações devem ser cientes do contexto (*context-awareness*), se adaptando automaticamente às necessidades dos usuários e às mudanças dos ambientes (AUGUSTIN; YAMIN; GEYER, 2005).

A adaptação pode explorar diversas características do ambiente, como a localização de pessoas e objetos, tempo, identificação de dispositivos, entre outros. A modelagem de contexto é um elemento central na construção desses sistemas. Ela define tipos, nomes, propriedades e atributos de todas as entidades que são relevantes para a aplicação (AUGUSTIN; YAMIN; GEYER, 2005).

O grau de refinamento e sensibilidade dos modelos determinam a “percepção do ambiente” pela aplicação. Os modelos devem somente representar as entidades e relacionamentos relevantes. Por outro lado, a sensibilidade ao contexto introduz uma variedade de desafios as áreas de modelagem e desenvolvimento de *software* (HENRICKSEN; INDULSKA, 2006).

O modelo de contexto deve suportar múltiplas representações do mesmo contexto, em diferentes formas e em diferentes níveis de abstrações. Assim como também deve ser capaz de capturar os relacionamentos existentes entre representações alternativas (HENRICKSEN et al, 2002).

O termo contexto pode ter diversas definições, pois se trata de um termo amplo e que

abrange diversas áreas. Segundo Dey & Abowd (2006), contexto pode ser definido como uma variedade de informações que podem ser utilizadas com o objetivo de caracterizar a situação de um grupo de entidades.

Alguns elementos de contexto são facilmente identificados com a utilização de sensores, como a temperatura do ambiente, o tamanho do display de um dispositivo ou o horário atual. Porém, outras formas de contexto ainda são complexas quanto ao reconhecimento e modelagem. Como o estado emocional ou o nível de estresse de uma pessoa quando esta realiza uma determinada atividade (MACHADO; AUGUSTIN, 2010).

2.2.1 Representação de Contexto

Dados de contexto podem ser representados de diversas formas em sistemas computacionais. Strang & Popien (2005) realizaram uma comparação entre as formas de representação de contexto e os requisitos que estas formas de representação atendem. A pesquisa se baseou em seis fatores principais: composição distribuída (*dc*), validação parcial (*pv*), qualidade da informação (*qua*), incompleto e ambíguo (*inc*), nível de formalidade (*for*) e aplicabilidade em ambientes existentes (*app*) (MACHADO, 2010).

A tabela resultante desta comparação é apresentada na Tabela 1.

Tabela 1 – Comparativo entre formas de representação de contexto (MACHADO, 2010)

Abordagens Requisitadas	dc	pv	qua	inc	for	app
Chave Valor	-	-	-	-	-	X
Esquema de Marcação	X	X	-	-	X	X
Orientação a Objetos	X	X	X	X	X	X
Baseada em Lógica	X	-	-	-	X	-
Gráfica	-	-	X	-	X	*
Baseada em Ontologias	X	X	X	X	X	X

Desta forma, constata-se que a modelagem de contexto baseada em ontologias e em orientação a objetos atendem aos principais requisitos para a representação completa de contextos. Porém, a representação baseada em orientação a objetos não proporciona a mesma

representação entre linguagens de programação distintas, pois cada linguagem possui sua sintaxe. Fato que não ocorre com ontologias, que podem ser utilizadas com uma sintaxe única por diversas linguagens de programação. Portanto, este modelo foi o escolhido para a representação do domínio no qual o ambiente ClinicSpace se encontra (MACHADO, 2010).

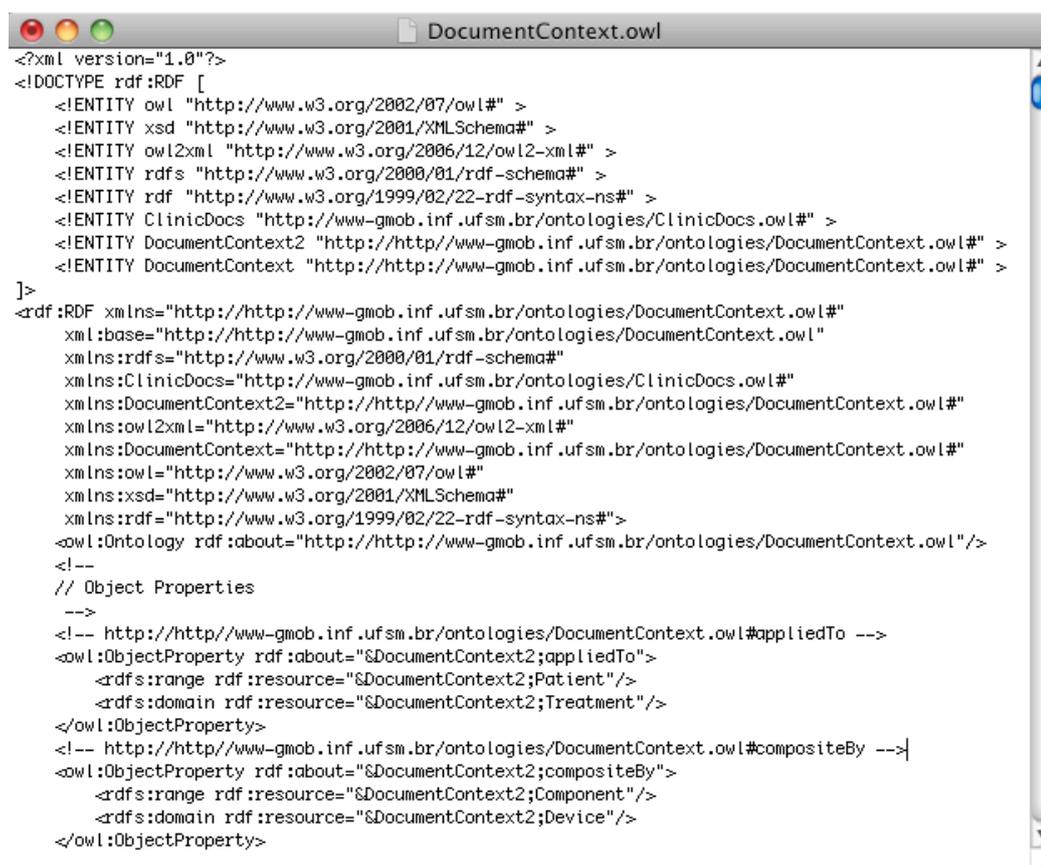
Ontologias possuem características e componentes comuns bem definidos. Os componentes básicos de uma ontologia são: classes - entidades organizadas taxonomicamente; relações - o tipo de interação entre os conceitos de um domínio; axiomas - usados para modelar sentenças sempre verdadeiras; instâncias - utilizadas para representar elementos específicos, ou seja, os próprios dados (GRUBER, 1996).

Para a construção de ontologias, é fundamental o uso de uma linguagem com semântica bem definida e expressiva para definir e criar relacionamentos entre os objetos definidos na ontologia. A linguagem OWL (*Web Ontology Language*) é uma das linguagens mais utilizadas para esta finalidade, sendo recomendada pela W3C - *World Wide Web Consortium* (W3C, 2011). Uma ontologia OWL pode incluir: relações de taxonomia entre classes; propriedades dos tipos de dados e descrições dos atributos de elementos das classes, propriedades do objeto e descrições das relações entre elementos das classes, instâncias das classes e instâncias das propriedades.

Na Figura 1 é apresentado um exemplo da representação de uma ontologia no formato de arquivo OWL. Como se pode observar, o formato OWL é um arquivo XML (*Extensible Markup Language*) com regras bem definidas para a representação de objetos da ontologia (W3C, 2004). Todas as propriedades, classes e indivíduos são definidos com *tags* específicas, que definem o tipo do objeto na ontologia (se é uma propriedade, classe ou indivíduo), o nome único deste objeto na ontologia (conhecido como URI¹ (*Uniform Resource Identifier*)) e propriedades que variam de acordo com o tipo de objeto definido.

Estas regras permitem com que a estrutura da ontologia seja modificada com o tempo, através da utilização de um motor de inferência (*reasoner*), que realiza operações e pode mudar a estrutura, as informações e o significado das informações definidas em uma ontologia (W3C, 2011).

¹ Cadeia de caracteres compacta usada para identificar ou denominar um recurso na internet (W3C, 2011).



```

<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY ClinicDocs "http://www-gmob.inf.ufsm.br/ontologies/ClinicDocs.owl#" >
  <!ENTITY DocumentContext2 "http://http://www-gmob.inf.ufsm.br/ontologies/DocumentContext.owl#" >
  <!ENTITY DocumentContext "http://http://www-gmob.inf.ufsm.br/ontologies/DocumentContext.owl#" >
]>
<rdf:RDF xmlns="http://http://www-gmob.inf.ufsm.br/ontologies/DocumentContext.owl#"
  xmlns:base="http://http://www-gmob.inf.ufsm.br/ontologies/DocumentContext.owl#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:ClinicDocs="http://www-gmob.inf.ufsm.br/ontologies/ClinicDocs.owl#"
  xmlns:DocumentContext2="http://http://www-gmob.inf.ufsm.br/ontologies/DocumentContext.owl#"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:DocumentContext="http://http://www-gmob.inf.ufsm.br/ontologies/DocumentContext.owl#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <owl:Ontology rdf:about="http://http://www-gmob.inf.ufsm.br/ontologies/DocumentContext.owl"/>
  <!--
  // Object Properties
  -->
  <!-- http://http://www-gmob.inf.ufsm.br/ontologies/DocumentContext.owl#appliedTo -->
  <owl:ObjectProperty rdf:about="&DocumentContext2;appliedTo">
    <rdfs:range rdf:resource="&DocumentContext2;Patient"/>
    <rdfs:domain rdf:resource="&DocumentContext2;Treatment"/>
  </owl:ObjectProperty>
  <!-- http://http://www-gmob.inf.ufsm.br/ontologies/DocumentContext.owl#compositeBy -->
  <owl:ObjectProperty rdf:about="&DocumentContext2;compositeBy">
    <rdfs:range rdf:resource="&DocumentContext2;Component"/>
    <rdfs:domain rdf:resource="&DocumentContext2;Device"/>
  </owl:ObjectProperty>

```

Figura 1 – Exemplo de arquivo OWL

Embora a linguagem OWL inclua um conjunto relativamente rico de construtores de classe, ela é muito fraca no que se refere às propriedades. Em particular, não há construtor de composição; por isso, é impossível capturar relacionamentos entre uma propriedade composta e outra propriedade (possivelmente composta). Para contornar esse problema, pode-se utilizar a linguagem SWRL (W3C, 2010) - uma extensão da OWL para a construção de regras em uma representação OWL.

A SWRL (*Semantic Web Rule Language*) é uma linguagem utilizada para a definição de regras em lógica de primeira ordem. Utilizada em conjunto com OWL, forma uma dupla potente de linguagens de representação do conhecimento (SWRL, 2010). O trecho de código da Figura 2 apresenta um exemplo de regra SWRL.

```

Pessoa(?p) ^ Hospital(?h) ^
EstáEm(?p, ?h) ^ Sensor(?s) ^
MonitoradoPor(?p, ?s) -> Paciente(?p)

```

Figura 2 - Exemplo de código SWRL

Observa-se que a SWRL é formada por um conjunto de anotações antecedentes, que são validadas por um motor de inferência e resultam na aplicação de uma regra posterior, definida após o símbolo “->”.

SWRL e OWL juntas permitem guardar informação categorizada e recuperá-la. Além disso, também permitem que um motor de inferência possa usar essa base para acrescentar novos conhecimentos à própria base, ou seja, as regras podem ser aplicadas sobre uma representação de conhecimento em OWL para representar um novo conhecimento (W3C, 2011).

A representação de elementos de contexto baseada em ontologias permite a representação e raciocínio lógico das informações para diversos propósitos (BETTINIA et al, 2010).

Segundo Bettinia (2010), a variação de ontologias geralmente utilizada para a representação de contexto é a OWL-DL, uma vez que ela está se tornando um padrão para vários domínios de aplicação e é suportada por um grande número de servidores de raciocínio. Esta variação de ontologias permite a modelagem de um domínio específico sem as limitações encontradas na variação OWL-Lite.

Várias linguagens OWL têm sido propostas para representar a descrição distribuída para os dados do contexto. As propostas mais promissoras são SOUPA (CHEN et al, 2004) para modelagem de contexto em ambientes pervasivos, e CONON (ZHANG et al, 2005), ontologias para ambientes de casas inteligentes. Estas ontologias distribuídas podem ser integradas com modelos específicos de aplicação de contexto por meio de extensões da linguagem OWL, tal como proposta por Bouquet (2004). Porém, nenhuma delas possui foco específico para elementos de contexto e informações clínicas de hospitais – mais precisamente na área de *healthcare*.

Para a arquitetura ClinicSpace, Machado (2010) definiu o OWL-DL como padrão utilizado, devido a sua capacidade de representação. Além da representação, a persistência de contexto é importante no que se refere a sistemas ubíquos pois como estes sistemas se adaptam constantemente, e necessitam de ferramentas para persistência e recuperação de contexto (MACHADO; AUGUSTIN, 2010).

2.2.2 Persistência de Contexto

Batzios & Mitko (2009) realizaram uma análise completa de ferramentas para o armazenamento de ontologias. Esta comparação foi escolhida como base, pois o padrão OWL-DL foi o modelo de representação escolhido por Machado (2010) para a representação dos elementos de contexto na arquitetura ClinicSpace.

Entre as ferramentas analisadas, estão a 3store (NOSQL, 2011), Sesame (NOSQL, 2011), Jena2 (NOSQL, 2011) e DB4Owl (BATZIOS; MITKO, 2009). Analisando os resultados dessa análise, observou-se que os principais formatos utilizados para persistência são: (i) os bancos de dados relacionais - adotados principalmente pela utilização de padrões consolidados e por possuírem bom desempenho em consultas SQL (*Simple Query Language*); e (ii) os bancos de dados baseados em triplas (*triple store databases*) – utilizados para o armazenamento de documentos RDF (*Resource Description Framework*) (W3C, 2011).

O projeto 3store é baseado na persistência de documentos RDF, utilizando banco de dados de modelo relacional. Para a realização de consultas, o 3store utiliza a linguagem RDQL (*RDF Data Query Language*), voltada para consultas na linguagem RDF (NOSQL, 2011). A realização de inferências e consultas é feita através de conversões para a linguagem SQL – este método gera muitos dados extras, comprometendo o desempenho em inferências realizadas em grandes bases de dados (BATZIOS; MITKO, 2009).

Já os projetos Jena2 e Sesame consistem de APIs que utilizam sistemas de banco de dados relacionais como sistemas para persistência de arquivos RDF e são compatíveis com o padrão OWL-Lite (W3C, 2011).

Para amenizar o volume de dados extras criados na conversão de ontologias para bancos de dados relacionais, o projeto DB4Owl (BATZIOS; MITKO, 2009) propôs a utilização do banco de dados orientado a objetos DB4o (NOSQL, 2011) para persistir um conjunto de classes que representam as entidades de uma ontologia. Na realização de consultas, é utilizada uma linguagem com semântica própria baseada em XML, que converte a consulta na linguagem de consulta utilizada pelo banco de dados DB4o.

Baseado na comparação entre os projetos relacionados, propôs-se a definição de uma ferramenta de persistência baseada em ontologias, chamada de SemantiCouch (mais detalhes na Seção 4.1) para ser utilizada em conjunto com a arquitetura ClinicSpace. O resumo comparativo entre as ferramentas é apresentado na Tabela 2.

Tabela 2 – Comparativo de soluções para a persistência de ontologias

	3store	Sesame	Jena2	Db4owl	SemantiCouch
Linguagem Consulta	RDQL	SeRQL	SPARQL	XML	SWRL, SQWRL
Suporte Nativo a OWL	Não	Não	OWL-Lite	Sim	Sim
Facilmente Escalável	Não	Não	Não	Não	Sim
Modelo de BD	RD	RD	RD	OO	JSON Docs
Baixo Overhead de dados	Não	Não	Não	Sim	Sim

A construção da arquitetura SemantiCouch é justificada com o intuito de oferecer algumas características não encontradas em outras ferramentas de persistência de ontologias. São elas:

- **Escalabilidade Vertical e Horizontal.** As ferramentas de persistência que suportam dados semânticos (documentos RDF ou OWL) oferecem escalabilidade vertical (expansão do banco de dados em uma única máquina). Porém, para a utilização destas ferramentas em sistemas pervasivos, é necessário que elas possam utilizar uma infraestrutura distribuída para armazenar e consultar dados. Os dados persistidos nesta arquitetura podem ser consultados em diversos lugares, por dispositivos dos mais diversos tipos - escalabilidade horizontal¹ (*scale out*);
- **Suporte a linguagens de consulta e inferência padronizadas para OWL.** As ferramentas analisadas suportam ou criam linguagens distintas para a consulta ou inferência, mas poderiam utilizar linguagens já padronizadas e amplamente utilizadas para a manipulação de dados semânticos no formato OWL. Dentre essas linguagens, destacam-se as linguagens SWRL e SWQRL, por se tratarem de padrões consolidados e permitirem a inferência e consulta de dados na linguagem OWL (W3C, 2011) (NOSQL, 2011);
- **Persistir apenas dados relacionados a ontologia.** A constante conversão de formatos e linguagens de consulta faz com que ocorra *overhead* constantemente nas ferramentas atuais, o que acarreta perda de desempenho considerável dependendo do tamanho da base de dados e do número de consultas realizadas. Além disso, o espaço ocupado por estes dados aumenta consideravelmente com o uso destas bases de dados (BATZIOS;

¹ Escalabilidade Horizontal: significa adicionar mais nós em um sistema de banco de dados; como, por exemplo, adicionar um novo computador para uma aplicação de software distribuído (ANDERSON et al, 2010).

MITKO, 2009);

- **Utilizar apenas parte da ontologia em memória.** O processamento de inferências e consultas deve ser feito apenas com dados relevantes, principalmente porque o desempenho e consumo de memória aumenta consideravelmente com o aumento de indivíduos na ontologia. Por isso, é importante que as modificações sejam processadas em uma parte limitada de dados (consultada na base de dados) (BATZIOS; MITKO, 2009).

2.3 Computação Ubíqua na Saúde

Na área da saúde, a computação ubíqua possui duas linhas de pesquisa principais:

- Homecare*, destinada à pesquisa e prototipação de sistemas que possuem como foco o cuidado de pacientes e monitoramento destes em suas próprias residências. Nesses sistemas, os pacientes são monitorados em relação a seus sinais vitais e, em caso de qualquer mudança fora dos padrões normais, clínicos são chamados e realizam o atendimento nas residências dos pacientes;
- Healthcare*, com foco na implantação de soluções ubíquas nas instituições hospitalares, auxiliando os profissionais clínicos na realização de suas tarefas diárias.

Conforme discussões realizadas em *workshops* com foco na área médica, como o *UbiHealth* (2010), a maioria das pesquisas tem abordado tecnologias ubíquas para desordens somáticas, como diabetes, hipertensão, obesidade. Isto foi evidenciado desde os primeiros *workshops*, ainda que outras pesquisas tenham se dirigido à demência e autismo, com foco menor em outras desordens mentais como depressão, ansiedade e manias. Essas desordens são universais, afetam todas as regiões do planeta, e torna-se importante contribuir para aliviar o sofrimento dos pacientes que sofrem com estas doenças (MACHADO, 2010).

Além dos estudos relacionados à detecção e auxílio no tratamento de doenças, a computação ubíqua é aplicada na criação de ambientes inteligentes em hospitais - *smart hospitals*, onde os sistemas modificam a sua rotina e forma de funcionamento de acordo com as necessidades dos profissionais clínicos, prevendo as constantes interrupções e colaboração entre profissionais na realização das suas atividades diárias (MACHADO, 2010).

Na próxima seção será apresentado um estudo sobre os projetos relacionados a área de

computação ubíqua aplicada na área de *healthcare*.

2.3.1 Projetos de Computação Ubíqua e Saúde

A área de hospitais inteligentes é a que mais influenciou nas definições do projeto ClinicSpace (Subseção 2.3.2), escopo desse trabalho. Entre os principais projetos que influenciaram a arquitetura ClinicSpace, destacam-se os projetos ABC (*Activity-Based Computing*), Aura e Gaia.

O Projeto ABC (BARDRAM; CHRISTENSEN, 2007) apresenta uma proposta para utilização de computação baseada em tarefas destinadas aos ambientes de saúde, e influenciou muitas das decisões tomadas no desenvolvimento da arquitetura ClinicSpace. O *framework* ABC foi desenvolvido para ser uma plataforma de programação onde as aplicações orientadas às atividades poderiam ser executadas. O *framework* lida com a complexidade computacional de gerenciar atividades distribuídas e colaborativas, adaptando-as aos serviços e recursos existentes no ambiente (BARDRAM, 2003).

A abordagem de contexto utilizada no projeto ABC é focada na questão da descoberta das atividades do usuário, baseada no contexto em que ele se encontra e em modelos de atividades previamente conhecidos.

Neste projeto, é disponibilizado um modelo de programação para o desenvolvimento de serviços e aplicações. A utilização deste modelo requer conhecimentos de programação que, geralmente, apenas profissionais de computação possuem. Assim, a programação de atividades não é uma tarefa simples e acessível aos profissionais clínicos. Além disso, não são providos mecanismos para permitir que os profissionais personalizem as atividades, inserindo nelas a sua forma particular de executá-las (BARDRAM; CHRISTENSEN, 2007).

A proposta do projeto Gaia (2011) visualiza um futuro no qual os espaços habitados pelas pessoas são interativos e programáveis. Os usuários interagem com seus escritórios, casas ou carros. Além disso, podem configurar o comportamento do ambiente se beneficiando dos recursos disponíveis. Dados e tarefas estão sempre acessíveis e são mapeados dinamicamente para os recursos convenientes e presentes na localização corrente pelo sistema GaiaOS (GAIA, 2011), podendo estender o seu espaço habitado para dispositivos pessoais que se integram ao ambiente. Este ambiente interativo, centrado no usuário, exige uma infraestrutura de software para operar com os recursos, observar propriedades do contexto, assistir

ao desenvolvimento e execução das aplicações. Assim, o *Middleware* Gaia objetiva o gerenciamento de recursos e fornece ao usuário interfaces orientadas ao espaço físico (GAIA, 2011).

A Tabela 3 apresenta uma comparação das principais soluções ubíquas para *healthcare* e a arquitetura ClinicSpace.

Tabela 3 – Comparativo entre projetos de *healthcare* e a arquitetura ClinicSpace

	Mobilidade	Contexto / Informações Clínicas	Personalização de Tarefas	Compartilhamento de Informações Clínicas
Gaia	X	Apenas Contexto	-	-
Aura	X	Apenas Contexto	X	-
ClinicSpace com o serviço SPC (Definido neste trabalho)	X	X	X	X

Como se pode observar, a arquitetura ClinicSpace possui características não encontradas em outros projetos para a área da saúde.

2.3.2 Projeto ClinicSpace

O projeto ClinicSpace (FERREIRA, 2009) (MACHADO, 2010) (RIZZETI, 2009) (SILVA, 2009) (SOUZA, 2010) (VICENTINI, 2010), atualmente em fase de desenvolvimento pelo grupo de pesquisas GMob na Universidade Federal de Santa Maria, possui como principal objetivo auxiliar os profissionais clínicos na realização das suas atividades em ambientes hospitalares. Para tal, o projeto visa a utilização de tecnologias provenientes das áreas de computação ubíqua e pervasiva e *End-user Programming*, permitindo com que os profissionais modelem suas tarefas de uma maneira fácil através de interfaces intuitivas.

Algumas pesquisas (BARDRAM; CHRISTENSEN, 2007) (QUEIROZ, 2011) demonstram que o foco em questões de administração e finanças faz com que exista uma alta

taxa de rejeição ao uso de sistemas hospitalares por profissionais da saúde. Os clínicos, além de utilizarem formulários em papel paralelamente ao uso dos sistemas devem alimentar manualmente os sistemas com as informações já preenchidas em papel. Outro aspecto é que esses sistemas não oferecem suporte à interrupção e colaboração na realização de tarefas clínicas, características predominantes no ambiente hospitalar (BARDRAM; CHRISTENSEN, 2007).

Além disso, estes sistemas não permitem o compartilhamento de informações do histórico de pacientes entre diferentes instituições. Isto dificulta o acesso a informações em caso de tratamento do paciente fora da área onde este reside, e implica na demora na coleta de informações do paciente e na redundância de dados – pois os mesmos exames realizados anteriormente pelo paciente devem ser refeitos em localidades diferentes (BARDRAM; CHRISTENSEN, 2007).

2.3.2.1 Arquitetura

Para prover o suporte computacional necessário para o auxílio na execução de tarefas, a arquitetura ClinicSpace é composta por camadas que integram um conjunto de serviços. Todas as camadas de serviços são gerenciadas pelo *middleware* EXEHDA (YAMIN, 2004), responsável pela integração dos componentes da arquitetura ClinicSpace.

Como apresentada na Figura 3, a arquitetura é separada por níveis, sendo eles: (i) nível superior, responsável pelas interfaces de interação do sistema com os usuários (médicos), composta pela IETC; (ii) nível intermediário, responsável pela modelagem de tarefas e interligação entre a camada superior (de interfaces) e a camada inferior; (iii) nível inferior, composto pelo *middleware* EXEHDA, que gerencia uma gama de recursos para a utilização pela arquitetura ClinicSpace (SILVA, 2009).

O *middleware* EXEHDA tem como principal característica a adaptação de recursos conforme o contexto de tarefas. A sua arquitetura é baseada em serviços, que em conjunto gerenciam um ambiente ubíquo (AUGUSTIN; YAMIN; GEYER, 2005).

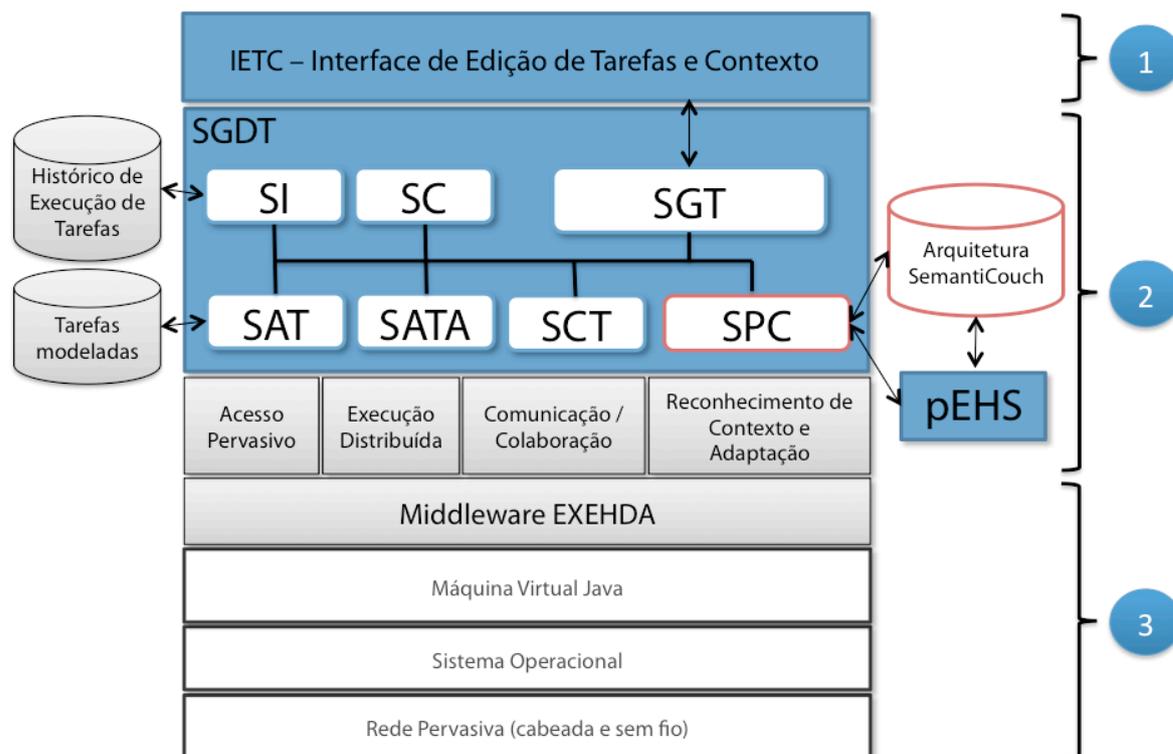


Figura 3 – Arquitetura ClinicSpace

Maiores detalhes sobre a arquitetura ClinicSpace são descritos no Apêndice A.

2.3.2.2 Problemas

Detectaram-se alguns problemas com a abordagem utilizada pela arquitetura ClinicSpace no que se refere à utilização de ontologias para a representação de contexto. São eles:

- **Utilização de toda a ontologia em memória:** A ontologia definida na arquitetura ClinicSpace é representada em um arquivo OWL único. Para utilizar as informações deste arquivo, a arquitetura precisa carregá-lo em memória de forma completa para executar qualquer operação. Isto ocorre em todos os nós da rede que precisam das informações que estão representadas na ontologia, causando (i) perda de desempenho, pois a arquitetura precisa carregar toda a ontologia mesmo que utilize somente uma parte dela, e (ii) inconsistência de dados, pois os dados da ontologia utilizados em um nó de rede não são compartilhadas entre os outros nós utilizados pela arquitetura

ClinicSpace.

- **A ontologia foi definida com o foco em atividades, mas não contempla o conteúdo clínico e alguns elementos de contexto relevantes nestas atividades:** A ontologia proposta por Machado (2010) teve como objetivo representar alguns dos elementos de contexto incorporados às atividades dos profissionais clínicos. Porém, não representa o conhecimento clínico envolvido nestas atividades, nem os prognósticos, doenças e medicamentos envolvidos nos atendimentos aos pacientes.
- **Os documentos utilizados pela arquitetura ClinicSpace não possuem semântica e sempre são apresentados da mesma forma aos usuários:** A arquitetura ClinicSpace não realiza nenhuma ligação entre as informações de contexto que são utilizadas constantemente na adaptação da arquitetura e os documentos clínicos armazenados na arquitetura. Além disso, as informações clínicas não possuem semântica, não oferecendo a possibilidade de adaptação ou seleção de partes destas informações conforme o contexto informado no momento da consulta a estas informações.
- **As informações clínicas não são compartilhadas:** As informações clínicas contidas nos documentos utilizados pelo pEHS (VICENTINI, 2010) e pela arquitetura ClinicSpace (não são compartilhados entre redes diferentes; ou seja, se um paciente realiza um exame em uma determinada cidade e precisa de informação contidas neste exame em outra cidade, o exame deverá ser realizado novamente pelo paciente.

A partir desse levantamento das necessidades da arquitetura ClinicSpace, propôs-se um novo serviço, que incorporado à arquitetura auxilie no tratamento de dados de contexto e de informações clínicas.

3 MODELAGEM DE CONHECIMENTO E CONTEXTO PARA A ARQUITETURA CLINICSPACE

Como o modelo de funcionamento da arquitetura ClinicSpace é baseado no auxílio aos profissionais clínicos em suas atividades cotidianas, faz-se necessária a criação de um serviço que: (i) forneça um mecanismo para a persistência e recuperação de contexto – uma base de dados de contexto, constantemente utilizado pela arquitetura ClinicSpace; (ii) permita a persistência de documentos clínicos e a sua recuperação de uma forma distribuída entre instituições hospitalares; e (iii) permita a recuperação de documentos e exames clínicos de acordo com o contexto da atividade realizada pelo usuário clínico, no momento da consulta desta informação.

3.1 Cenário de Uso

Para exemplificar o uso contextualizado da informação, descreve-se um possível cenário de utilização do sistema (Figura 4):

“Um médico neurologista deseja consultar o histórico de exames de seu paciente no seu dispositivo móvel. O sistema obtém, do contexto atual, a identificação do paciente, a localização do médico, do paciente, dos dispositivos e dos sensores. O médico recebe os resultados de exames realizados recentemente pelo paciente que está sendo atendido, relativos à sua especialidade clínica.

Além disso, os exames são apresentados ao médico de forma adaptada ao seu dispositivo – em relação a capacidade de visualização do *display*, banda da rede disponível e à localização onde o médico se encontra e as pessoas ao seu redor (se esta informação exige ou não privacidade)”.

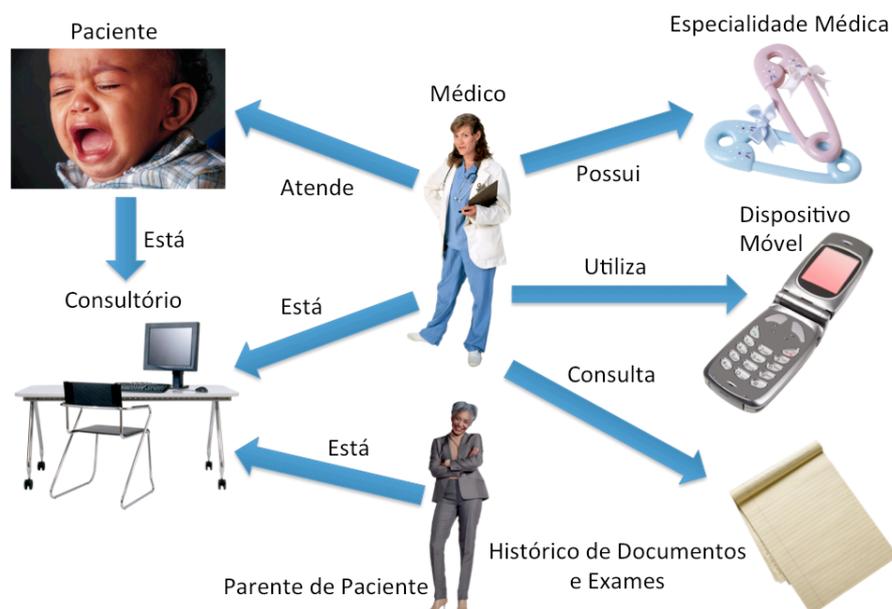


Figura 4 – Cenário de utilização do sistema

Para disponibilizar essas funcionalidades, propõe-se um novo serviço que, incorporado à arquitetura ClinicSpace, permite o armazenamento de dados de contexto e a recuperação dos mesmos pelos serviços internos do ClinicSpace. Além disso, baseado nas informações de contexto armazenadas, o serviço oferece acesso aos documentos armazenados de forma contextualizada. Para tal, seleciona versões dos mesmos de acordo com as informações de contexto informadas na consulta destes documentos, em comparação com as informações de contexto que estão integradas aos documentos armazenados na arquitetura ClinicSpace.

O desenvolvimento do serviço definido neste trabalho foi dividido em três módulos, como mostrado na Figura 5:

- **Ontologia:** definiu-se como o conhecimento clínico e as informações de contexto são representados computacionalmente, através da representação semântica de informações contidas em documentos clínicos (Seção 3.3) e dos elementos de contexto comuns em ambientes hospitalares, os quais podem interferir na seleção contextualizada destes documentos pela arquitetura ClinicSpace (Seção 3.2);
- **Arquitetura SemantiCouch:** para realizar a persistência dos dados de contexto e de informações clínicas, desenvolveu-se uma arquitetura de persistência chamada SemantiCouch, que permite a utilização de ontologias em bases de dados distribuídas, provendo acesso a dados semânticos por sistemas ubíquos (Seção 4.1);
- **Módulo de Seleção Contextualizada:** Para realizar a análise de contexto e das

informações utilizadas pelo serviço proposto, desenvolveu-se um módulo de seleção contextualizada. O módulo interpreta os dados de contexto e as informações clínicas, e realiza a comunicação deste serviço com os outros serviços da arquitetura ClinicSpace (Seção 4.2).



Figura 5 – Componentes internos do serviço de persistência de contexto e seleção contextualizada de documentos

A primeira etapa na definição do serviço de persistência de contexto e seleção contextualizada de documentos trata da modelagem de conhecimento e contexto para a arquitetura ClinicSpace, ou seja, (i) a forma como as informações do domínio são representadas computacionalmente, e (ii) quais informações são relevantes¹ para o serviço proposto. Este processo é descrito na seção a seguir.

3.2 Representação do Domínio

Para realizar a seleção contextualizada de documentos e exames clínicos na arquitetura ClinicSpace, a ontologia original definida por Machado (2010), ilustrada na Figura 6, foi estendida e modificada neste trabalho. Desta forma possuindo como principal objetivo

¹ Relevantes na realização das atividades clínicas modeladas na arquitetura ClinicSpace – as atividades são descritas no Apêndice A.

representar outros elementos de contexto, tais como documentos e exames clínicos, que são armazenados juntamente com a ontologia.

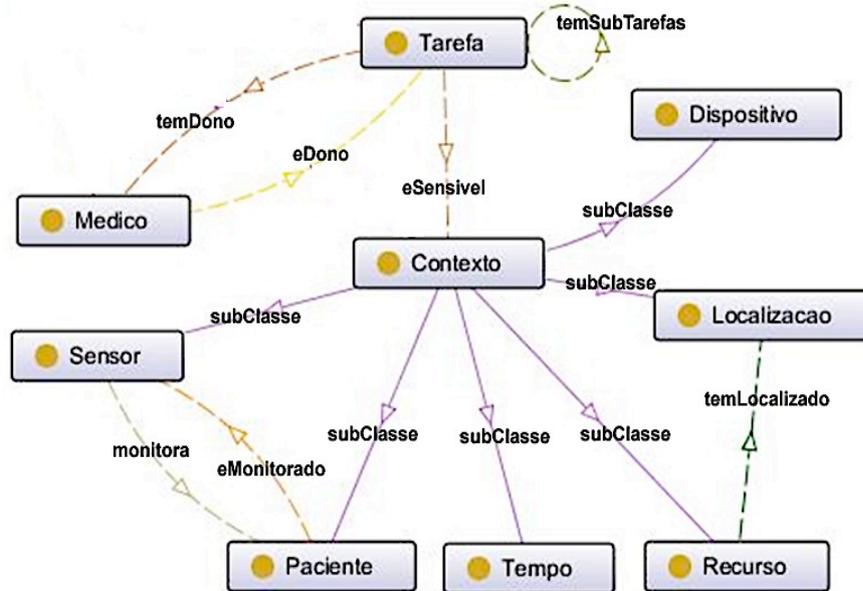


Figura 6 – Modelagem de elementos de contexto (MACHADO, 2010)

Como se pode observar na Figura 6, os elementos de contexto representados na ontologia utilizada pela arquitetura ClinicSpace estão diretamente relacionados à realização das tarefas pelos médicos, porém não representam questões clínicas, como doenças e documentações, utilizadas nos ambientes hospitalares.

A partir da definição desta ontologia por Machado (2010), foram expandidas as representações de contexto e informações clínicas. Para definir a ontologia, seguiu-se o processo denominado *Ontology Development 101* (RAUTENBERG et al, 2008). Este processo consiste de um guia, com um conjunto de passos e dicas para a determinação de entidades e relacionamentos entre estas entidades, para a construção de uma ontologia.

A Figura 7 apresenta o fluxo de trabalho definido no método *Ontology Development 101*, sendo que: (a) descreve os passos sugeridos pelo guia e (b) descreve um exemplo da concatenação dos passos, no processo de construção de uma ontologia.

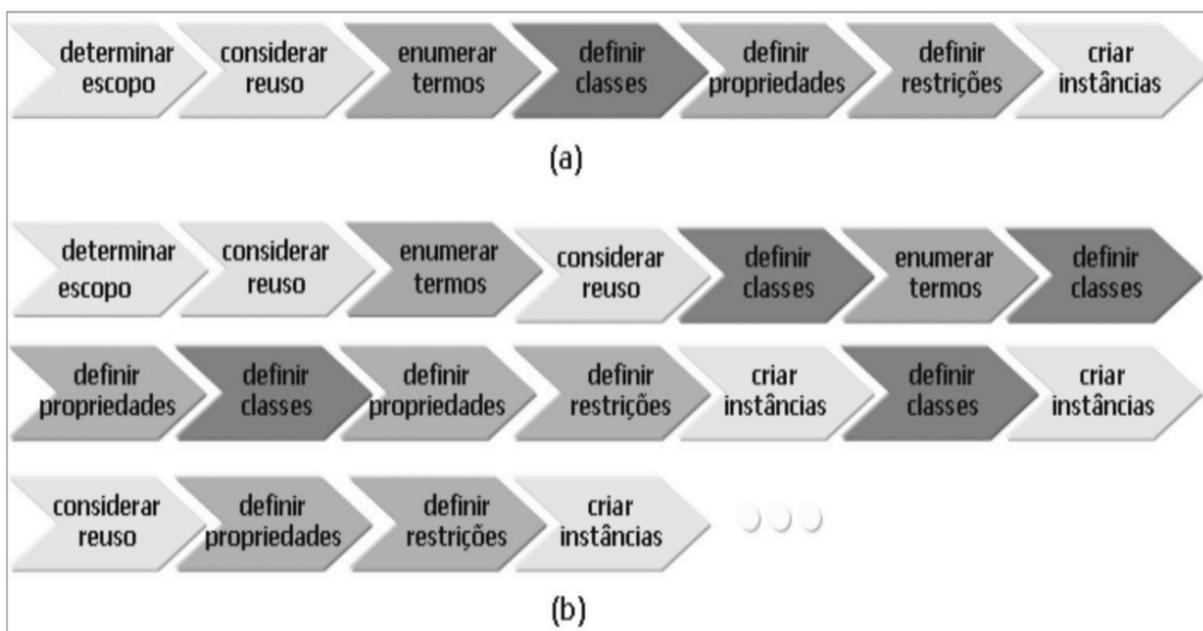


Figura 7 – Processo de desenvolvimento proposto pelo guia *Ontology Development 101* (RAUTENBERG et al, 2008)

A utilização desta metodologia para o desenvolvimento da ontologia se deve principalmente pelo equilíbrio entre as representações, utilizando restrições e a possibilidade de extensão da ontologia, permitindo futuras definições (RAUTENBERG et al, 2008).

3.2.1 Representação de Elementos de Contexto

Para realizar a construção de uma ontologia capaz de representar elementos de contexto, primeiramente criou-se uma descrição formal representando o domínio que se quer representar. A descrição relata as atividades cotidianas do ambiente hospitalar e os elementos de contexto presentes neste tipo de ambiente e que, posteriormente, serão utilizados pela arquitetura ClinicSpace para a adaptação de código/conteúdo e no auxílio à realização de tarefas clínicas.

A descrição dos elementos de contexto que devem ser cobertos pela ontologia é limitada a variáveis do ambiente hospitalar pervasivo que podem, de alguma forma, interferir na adaptação de documentos. Isso permite aos profissionais de saúde realizarem suas atividades cotidianas com o auxílio das tecnologias de computação pervasiva e ubíqua, e monitorarem o ambiente, buscando por informações de contexto que possam interferir no

andamento de uma atividade que vai ser iniciada ou que já está sendo executada (MACHADO, 2010).

As informações de contexto contidas neste ambiente podem ser classificadas tanto na forma física, através de informações de temperatura dos pacientes e localização dos profissionais de saúde, dispositivos, pacientes ou parentes dos pacientes dentro do ambiente hospitalar, quanto na forma lógica, detectando alterações em sistemas computacionais (VICENTINI, 2010).

O acesso às informações pode ser realizado por pessoas utilizando dispositivos dos mais diversos tipos, sejam eles móveis ou fixos. Desta forma, as informações e interfaces devem se adaptar à heterogeneidade de dispositivos e componentes que compõe estes dispositivos. Médicos montam o modelo das suas atividades em seus dispositivos móveis e, a partir deste momento, o ambiente pervasivo monitora o ambiente de forma a auxiliar o profissional na realização das suas atividades cotidianas (VICENTINI, 2010).

As atividades podem ser realizadas por um único profissional ou por um grupo de profissionais, que podem ser especializados em diferentes áreas médicas. Além disso, as informações relevantes são diferentes para cada especialidade médica ou profissional (MACHADO et al, 2010).

Apesar de os sistemas de informação em saúde possuírem todas as informações dos pacientes, devem ser repassadas aos profissionais somente as informações relevantes para a execução de suas atividades, ou as informações permitidas para o acompanhamento do estado de saúde de um paciente por seus parentes. A falta de garantia à confidencialidade é um dos problemas encontrados nos sistemas de saúde utilizados atualmente (VICENTINI, 2010).

Na arquitetura ClinicSpace, as tarefas também são descritas em uma forma ontológica, permitindo o tratamento e adaptação de acordo com as atividades realizadas no momento. A Figura 8 ilustra a representação de uma atividade na ontologia utilizada na arquitetura ClinicSpace.

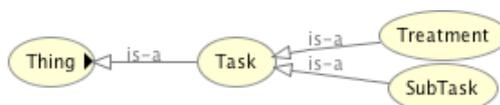


Figura 8 – Representação de uma tarefa na ontologia proposta para a arquitetura ClinicSpace

As atividades descritas na Figura 9, foram definidas como as mais comuns em

ambientes hospitalares em uma pesquisa realizada por (BARDRAM; CHRISTENSEN, 2007).

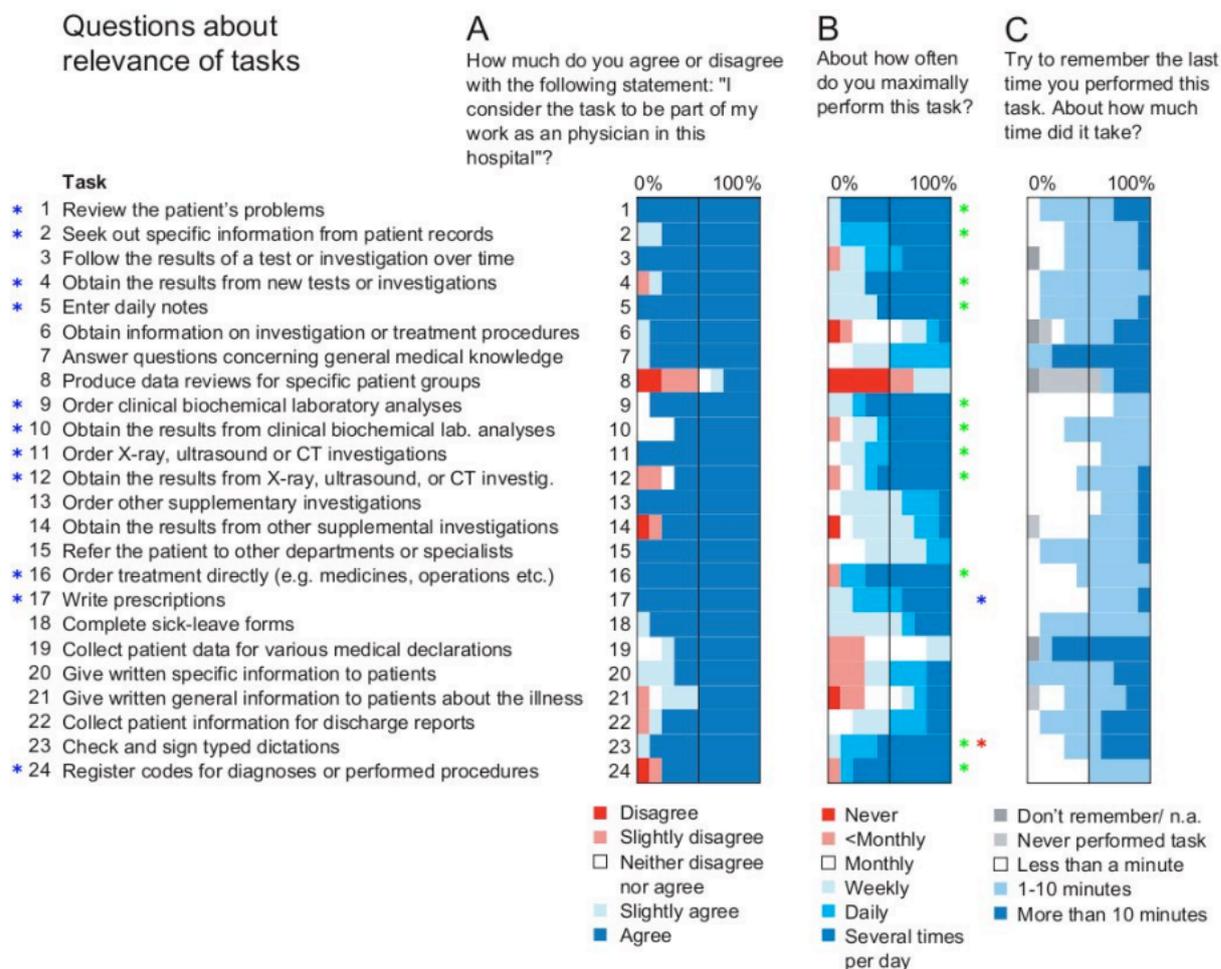


Figura 9 – Atividades clínicas comuns em ambientes hospitalares (BARDRAM; CHRISTENSEN, 2007)

A partir desta pesquisa, (VICENTINI, 2010) elencou as 11 tarefas realizadas com maior frequência no ambiente hospitalar. Cada tarefa possui, por sua vez, subtarefas que auxiliam na execução de uma determinada atividade. As subtarefas podem ser atividades realizadas no ambiente clínico que, em conjunto, resultam em outra atividade ou podem ser processos realizados por sistemas ubíquos para auxiliar na realização de uma atividade.

A partir da pesquisa realizada no domínio hospitalar e as atividades realizadas neste ambiente, definiu-se as principais classes da ontologia, também chamadas de superclasses, que definem as áreas de contexto abordadas na arquitetura ClinicSpace. Estas áreas foram sendo refinadas durante a modelagem da ontologia, elas representam elementos de contexto que podem influenciar o ambiente pervasivo, na seleção contextualizada de conteúdo e adaptação em relação às funcionalidades.

Como principais elementos de contexto, definiram-se:

- **Aplicação (*Application*)**. As aplicações descritas são os próprios serviços e subsistemas da arquitetura ClinicSpace, ou sistemas externos que desejam ter acesso as informações utilizadas pelo ClinicSpace;
- **Tempo (*Time*)**. O tempo é determinante na realização de tarefas, pois de acordo com a duração das atividades, o sistema pode distribuir tarefas entre profissionais clínicos, dependendo se o profissional está perto do horário de saída ou está no seu turno de trabalho normal, por exemplo;
- **Doença (*Disease*)**. As doenças são relacionadas com informações presentes nos documentos relacionados aos pacientes. Além disso, quando um paciente é detectado pelo sistema e não foi atendido ainda, pode receber um atendimento prioritário, caso o sistema verifique que este paciente possua uma doença séria ligada com a sua ficha hospitalar;
- **Localização (*Localization*)**. A localização dos recursos e pessoas é determinante na delegação de tarefas, recursos e no acesso às informações clínicas. O sistema filtra o acesso às informações de pacientes somente quando os médicos estão presentes no ambiente hospitalar, mesmo que eles possuam acesso a IETC em seus dispositivos móveis pessoais. Em relação à delegação de tarefas e recursos, o sistema delega tarefas e realiza avisos de acordo com a localização de recursos, pacientes e médicos. Por exemplo, se um médico está realizando um procedimento na unidade de terapia intensiva, este não pode ser interrompido pelo sistema ou receber avisos externos não relacionados com a atividade que ele está realizando no momento, pois a atividade que ele está realizando é de alto risco;
- **Pessoa (*Person*)**. Todas as atividades da arquitetura ClinicSpace são realizadas baseando-se nas tarefas realizadas por pessoas e elementos de contexto do ambiente. Desta forma, torna-se indispensável a representação das pessoas presentes no ambiente hospitalar. Tanto em relação ao paciente, quanto em relação aos funcionários e profissionais clínicos envolvidos nas atividades médicas;
- **Medicamento (*Medicine*)**. De acordo com as necessidades do paciente e dos pedidos realizados pelo médico, o sistema pode encaminhar o paciente para o setor responsável pelos medicamentos ou emitir uma receita médica para o paciente. Além disso, o sistema analisa o histórico de medicamentos do paciente, se este está disponível e informa ao médico sobre esse histórico no momento da consulta;

- **Componente (*Component*).** Os componentes são determinantes na coleta e apresentação de informações. Basicamente, todos os dispositivos são compostos de componentes, os quais determinam as características e capacidades de um dispositivo em si;
- **Material para Exame (*MaterialForClinicExamination*).** Caso um paciente precise realizar um exame, ele pode ser encaminhado ao setor responsável de acordo com a urgência, ou seja, se o exame precisa ser realizado o mais rápido possível, o paciente é encaminhado ao laboratório do próprio hospital;
- **Conhecimento (*Knowledge*).** O conhecimento médico define o acesso aos documentos de pacientes pelos profissionais clínicos. Por exemplo, se um especialista em pediatria está presente na instituição hospitalar, ele terá acesso aos documentos relacionados a área médica de pediatria no atendimento aos seus pacientes. Além disso, em caso de emergência e dependendo da disponibilidade do médico, ele pode receber a delegação de atendimento a um paciente que tenha algum problema relacionado à sua especialidade médica;
- **Prognóstico (*Prognostic*).** O prognóstico do paciente pode determinar a delegação de tarefas a determinados profissionais clínicos. Por exemplo, se o paciente está esperando o atendimento de um médico e apresenta o prognóstico “vômito”, o sistema dispara um evento pedindo a colaboração de um profissional clínico para fazer um atendimento ao paciente;
- **Dispositivo (*Device*).** Os dispositivos utilizados no ambiente hospitalar definem como a informação será apresentada, a formatação da informação é de acordo com as características da rede e do dispositivo, tais como largura de banda, capacidade de armazenamento e propriedades de um *display*, caso o dispositivo o possua. Dispositivos podem ser fixos – computadores e terminais de salas específicas, ou móveis – telefones celulares ou sensores.

Para realizar a modelagem da ontologia, utilizou-se o software *Protégé* na versão 3.4.4 (STANFORD, 2011). Uma visão geral das classes que representam elementos de contexto na ontologia é apresentada na Figura 10.

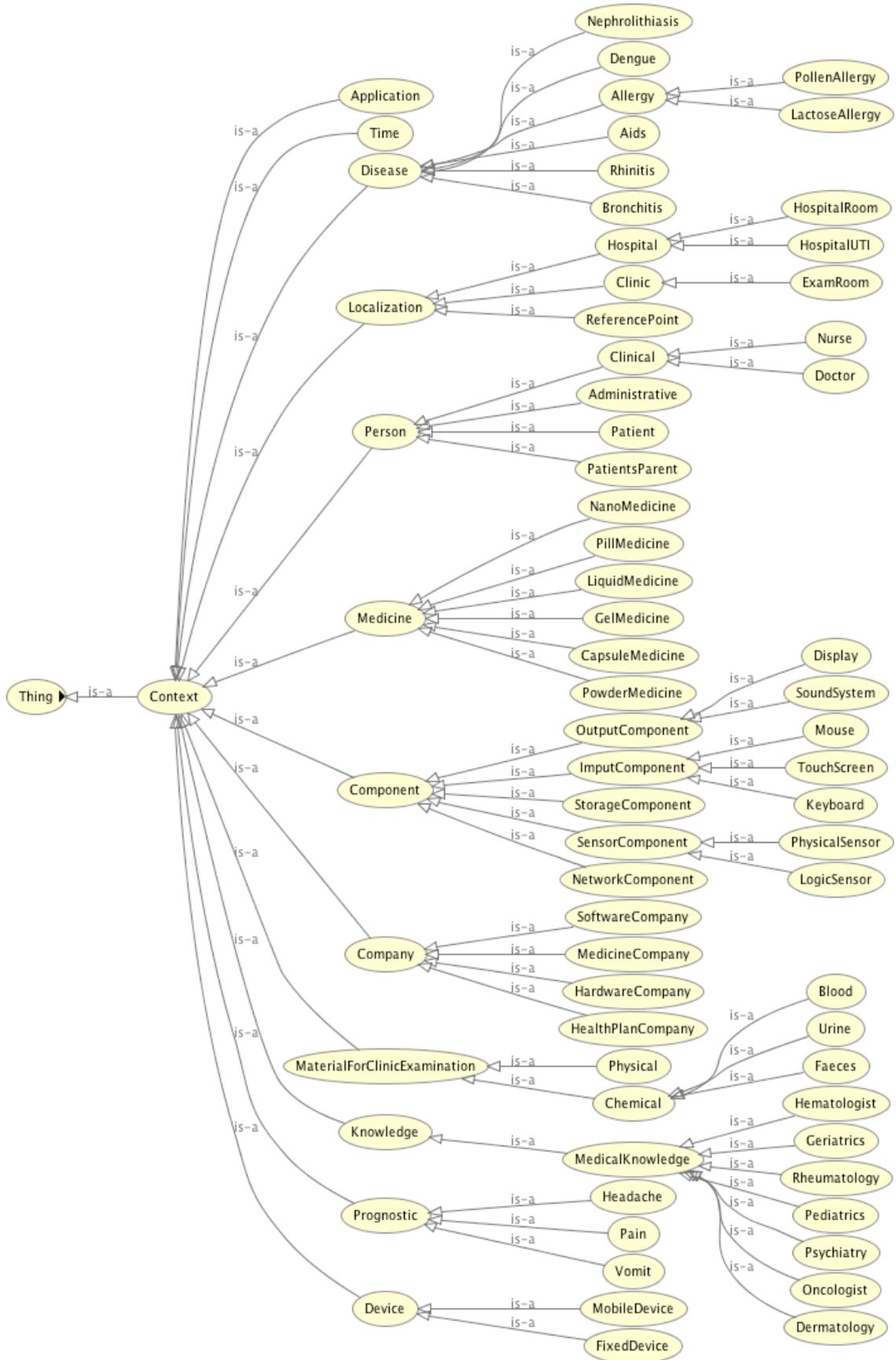


Figura 10 – Ontologia proposta para a representação de elementos de contexto

Além da definição dos elementos de contexto utilizados pela arquitetura ClinicSpace, é necessário que exista uma definição de informações clínicas relevantes do ambiente hospitalar. Tanto em relação a informações clínicas existentes em documentos e exames utilizados pelos profissionais clínicos quanto em relação a informações clínicas relacionadas a procedimentos realizados nos ambientes hospitalares. Portanto, a próxima etapa do trabalho trata de como as informações clínicas relevantes para a arquitetura ClinicSpace foram modeladas na ontologia utilizada pelo sistema.

3.2.2 Representação do Conhecimento Clínico

Para realizar a modelagem de documentos clínicos em uma ontologia para a utilização pelo ClinicSpace, usou-se documentos e exames clínicos provenientes de instituições hospitalares. A partir de uma lista de documentos utilizados com maior frequência em atividades hospitalares por profissionais clínicos, obtida através de uma pesquisa realizada em um hospital no interior do Rio Grande do Sul (Tabela 4), determinou-se o conhecimento clínico envolvido e as informações de contexto relacionadas no preenchimento destes documentos e inseriu-se estas informações na ontologia.

Tabela 4 – Lista de documentos frequentemente utilizados no ambiente hospitalar

Número	Documentos Utilizados
1	Relatório de Exame Laboratorial
2	Relatório de Evolução de Enfermagem
3	Prontuário Médico do Paciente
4	Relatório Cirúrgico
5	Prescrição Médica Diária
6	Relatório de Anamnese Diária
7	Relatório de Evolução Diária
8	Registro de Internação do Paciente
9	Solicitação de Exame
10	Solicitação de Encaminhamento para outro instituto hospitalar
11	Nota diária sobre o paciente

Após a listagem de quais documentos eram mais utilizados em atividades clínicas por médicos, utilizou-se os modelos de documentos utilizados em instituições hospitalares para a extração de informações clínicas relevantes. Um exemplo de mapeamento de informações de um documento para uma representação ontológica é apresentado na Figura 11.



Figura 11 – Exemplo de representação das informações clínicas de um documento clínico

As informações foram omitidas na Figura 11 por questões de ética em relação às informações reais de pacientes. As informações foram modeladas através de *DataProperties*¹ de classes na ontologia. Além disso, foram modeladas as ligações entre as classes.

A modelagem dos dados clínicos na ontologia foi realizada tendo como base duas superclasses principais: (i) a classe *DocumentPart*, que define uma pequena parte de um documento, suas informações e os elementos de contexto que permitem a adaptação deste documento, e (ii) a classe *Document*, que define um conjunto de partes de documentos e associa estas partes de documentos a elementos de contexto relevantes.

¹ Relacionamento entre um indivíduo da ontologia e um tipo de dado definido na ontologia (W3C, 2011)

Esta abordagem foi escolhida considerando as condições de uso. Por exemplo, se um exame de sangue padrão é realizado por um determinado paciente, este pode conter informações de interesse de diferentes especialidades médicas. Então, o serviço de seleção contextualizada pode selecionar apenas partes ou versões deste documento, conforme a necessidade da consulta de uma determinada especialidade. A Figura 12 apresenta as principais classes relacionadas à modelagem de documentos clínicos na ontologia.

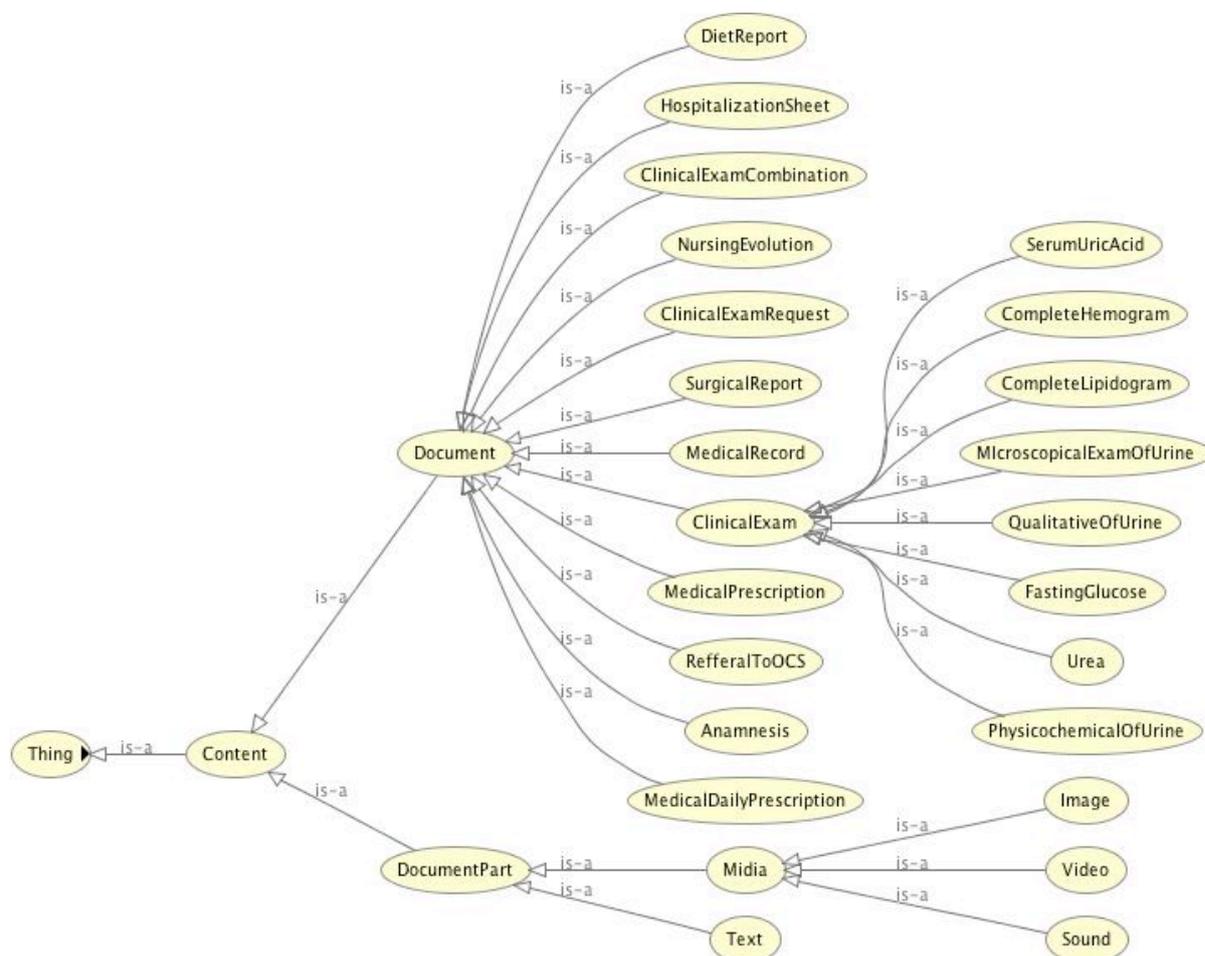


Figura 12 – Modelagem de documentos clínicos na ontologia

Desta forma, as informações contidas nos documentos clínicos possuem semântica, e podem ser facilmente convertidas para outros formatos, como por exemplo os padrões CCR (ASTM, 2011) e HL7 (HL7, 2011), utilizados em diversos sistemas que utilizam prontuário eletrônico.

4 UM SERVIÇO DE PERSISTÊNCIA DE CONTEXTO E SELEÇÃO CONTEXTUALIZADA DE DOCUMENTOS

A partir da definição da forma de representação e construção da ontologia que representa elementos de contexto e documentos clínicos, definiu-se o serviço de persistência de contexto e seleção contextualizada de documentos. O processo de definição do serviço é apresentado neste capítulo.

4.1 Persistência de Contexto – Arquitetura SemantiCouch

A fim de oferecer interfaces para o armazenamento e recuperação de contexto de forma simples e oferecer suporte à distribuição de ontologias, definiu-se uma arquitetura de persistência de contexto, em forma de serviço integrado à arquitetura ClinicSpace, que permite o acesso as ontologias que representam o contexto e informações clínicas.

4.1.1 Definição da Arquitetura SemantiCouch

Como base para o armazenamento e recuperação da ontologia definida para a utilização pela arquitetura ClinicSpace, criou-se a arquitetura SemantiCouch. Esta arquitetura trabalha como uma camada superior à camada de controle do banco de dados CouchDB (ANDERSON et al, 2010), e (i) realiza a conversão de arquivos OWL para o esquema de banco de dados utilizado pelo CouchDB, (ii) fornece suporte a um conjunto de classes definidas para a representação simplificada de ontologias na linguagem Java, e (iii) oferece suporte à realização de consultas e inferências nas ontologias persistidas no banco de dados CouchDB.

A escolha do banco de dados CouchDB se deve ao fato de atender aos seguintes critérios:

- **Facilmente escalável.** O banco de dados CouchDB oferece mecanismos para facilitar a distribuição de dados entre diversos nós em uma rede, oferecendo escalabilidade

horizontal. Entre esses mecanismos, destacam-se a fácil replicação de documentos entre nós distribuídos em uma rede e o controle de versões de documentos, permitindo o controle de concorrência nos acessos a estes documentos (ANDERSON et al, 2010);

- **Modelo de dados baseado em documentos.** O banco de dados utiliza o modelo de persistência baseado em documentos JSON (2011), os quais tem uma estrutura facilmente convertida para XML e possuem *frameworks* de auxílio para a sua utilização por *webservices* (ANDERSON et al, 2010);
- **Compatibilidade com diversas linguagens de programação.** O banco de dados CouchDB pode ser utilizado independentemente de linguagem de programação, devido a interface de consulta *RESTful API* (ANDERSON et al, 2010);
- **Portabilidade através do uso de versões compatíveis com dispositivos móveis.** O banco de dados CouchDB possui versões compatíveis com dispositivos móveis. Desta forma, aplicações móveis que utilizam o banco de dados CouchDB podem utilizar o banco de dados mesmo que o dispositivo esteja *offline*, pois o banco de dados irá sincronizar automaticamente as informações quando o dispositivo móvel ficar conectado a rede onde os outros nós do banco de dados estão em execução (COUCHBASE, 2011).

A arquitetura permite a ligação com sistemas ubíquos em três formas distintas: (i) através da utilização das classes definidas no pacote *OO* para a realização de operações nas ontologias persistidas nas bases de dados; (ii) através da importação e exportação de dados no padrão OWL; e (iii) através da realização de consultas e inferências baseadas nas linguagens SWRL e SQWRL.

A partir do momento em que uma ontologia é criada ou importada, através de um arquivo OWL já definido, a arquitetura oferece a possibilidade de replicação das informações presentes na ontologia entre diferentes nós de uma rede. A configuração da replicação de informações pode ser feita tanto pelos utilitários de configuração do banco de dados CouchDB quanto pela API Ektor, que possui objetos de comunicação utilizados no pacote *Comm*.

A Figura 13 apresenta uma representação básica da arquitetura SemantiCouch, utilizada para a persistência distribuída de ontologias.

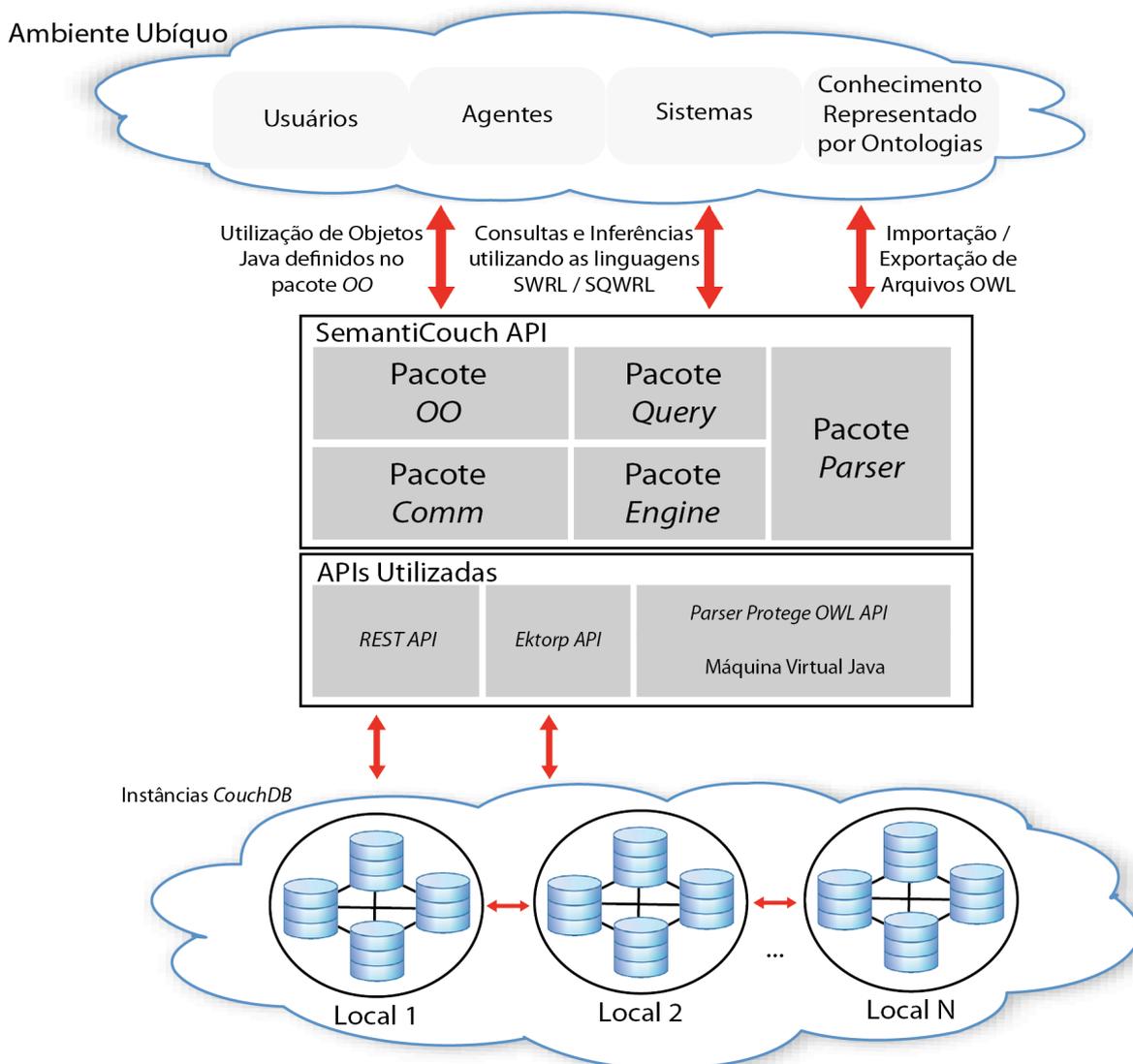


Figura 13 – Arquitetura básica da arquitetura SemantiCouch

A estrutura da arquitetura foi separada em pacotes, que são responsáveis por funções distintas no processo de persistência e consulta de ontologias. Os pacotes e suas respectivas funções são:

- **Pacote Comm.** Pacote responsável pela comunicação do pacote *Engine* e as APIs de comunicação Ektorp (EKTORP, 2011) e RESTful (ANDERSON et al 2010) com o banco de dados CouchDB. Controla os erros de comunicação com o banco de dados, o versionamento de documentos JSON, e gerencia a criação de documentos nas instâncias do banco de dados CouchDB. A Figura 14 apresenta o diagrama das classes que compõe o pacote *Comm*.

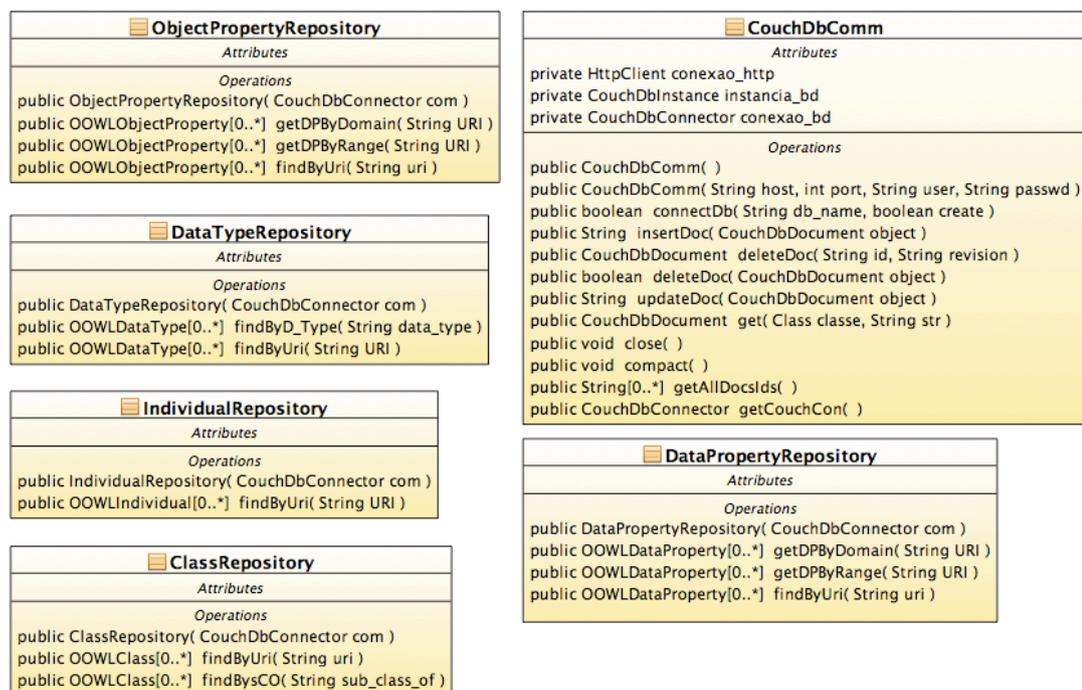


Figura 14 – Diagrama de classes do pacote *Comm*

- Pacote OO.** Pacote responsável pela definição de classes¹ que representam os componentes de ontologias persistidas no banco de dados. Essas classes podem ser utilizadas pelos desenvolvedores para a comunicação direta de classes Java de seus sistemas com os métodos de persistência e consulta na arquitetura de persistência. A classe *OOWLModel* é a classe principal para a modelagem de uma ontologia persistida no banco de dados, e a classe *OOWLOntology* é a classe que define a ontologia persistida em memória. A Figura 15 apresenta um diagrama das classes que compõe o pacote *OO* para a integração de programas que utilizam a linguagem Java com a arquitetura de persistência.

¹ Não há relacionamento direto entre as classes pois cada uma delas realiza operações específicas em tipos diferentes de entidades na ontologia.

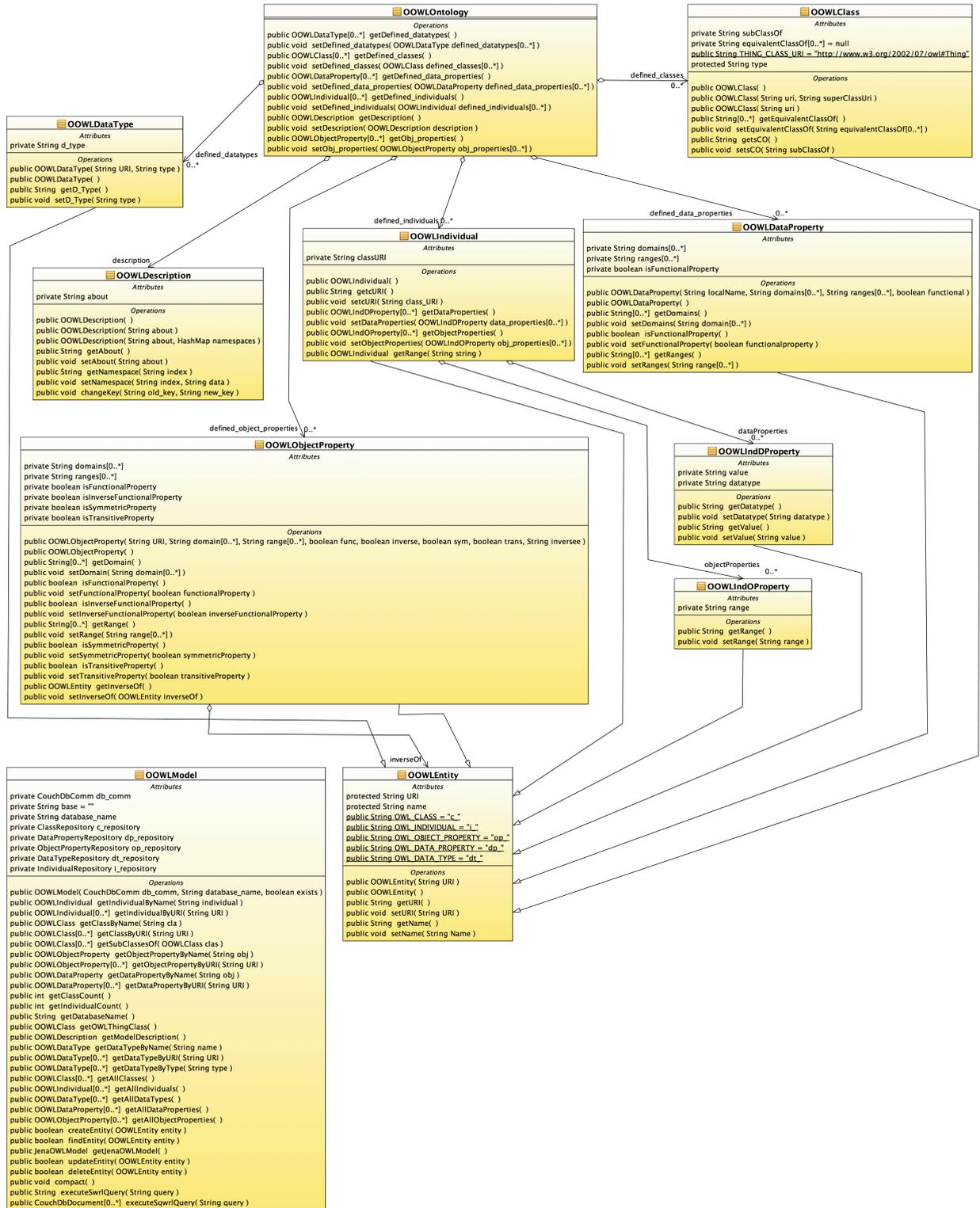


Figura 15 – Diagrama de classes do pacote OO

- Pacote Engine.** Pacote responsável pela interligação e controle entre os pacotes utilizados pela API. Além disso, permite a configuração da replicação de bases de dados entre nós – detalhes sobre o processo de replicação de informações são

discutidos na Seção 4.1.4;

- **Pacote Parser.** Pacote responsável pela conversão de formatos para a exportação ou importação de documentos no formato OWL ou para a conversão interna de formatos utilizados nas consultas, inferências e demais operações no banco de dados. Detalhes sobre o processo de conversão de formatos são apresentados na Seção 4.1.2;
- **Pacote Query.** Pacote responsável pelo controle de consultas e inferências na base de dados. Oferece o suporte às linguagens SWRL e SQWRL, fazendo a conversão destas para as linguagens utilizadas para a realização de consultas no banco de dados CouchDB. Detalhes sobre o processo de consulta são descritos na Seção 4.1.3;

Deste modo, a arquitetura de persistência SemantiCouch, em conjunto com o serviço de persistência de contexto e seleção contextualizada de documentos, permite o armazenamento e recuperação de informações de contexto modeladas em uma ontologia, distribuída entre os nós da rede pervasiva utilizada pela arquitetura ClinicSpace. Além disso, permite a persistência de documentos e exames clínicos de forma distribuída, associada à informações de contexto, permitindo a recuperação destas informações de forma personalizada.

Para garantir a compatibilidade com arquivos de ontologias comumente utilizados, a arquitetura oferece suporte à conversão de formatos RDF e OWL para o formato do banco de dados (documentos JSON), e vice-versa.

4.1.2 Processo de Conversão de Formatos de Arquivos

Além da estrutura de programação, a arquitetura mantém uma estrutura de armazenamento definida no banco de dados, reduzindo a necessidade de criação de dados extras, além dos já informados pela ontologia.

A estrutura de dados é baseada nas premissas de desempenho e escalabilidade definidas por Anderson et al (2010), que declara a importância da fragmentação de documentos no banco de dados CouchDB, caso o objetivo do utilizador do banco de dados seja a replicação de dados. Tal característica é necessária pois o banco de dados CouchDB realiza a replicação por documentos. Portanto, Anderson et al (2010) informa que documentos menores, porém em maiores quantidades, são replicados de forma mais rápida e consistente

que poucos documentos com maiores volumes de dados por documento.

Decidiu-se pela utilização de um documento por entidade da ontologia. Desta forma, obtem-se uma alta fragmentação quanto ao número de arquivos, e mantém-se a organização dos arquivos sem a necessidade de um controle complexo, pois cada documento no banco de dados se refere à uma entidade ou instância de entidade na ontologia – identificada, quanto ao seu tipo, no nome do documento.

A Figura 16 ilustra o processo de conversão de um arquivo de exemplo OWL para uma instância do banco de dados CouchDB.

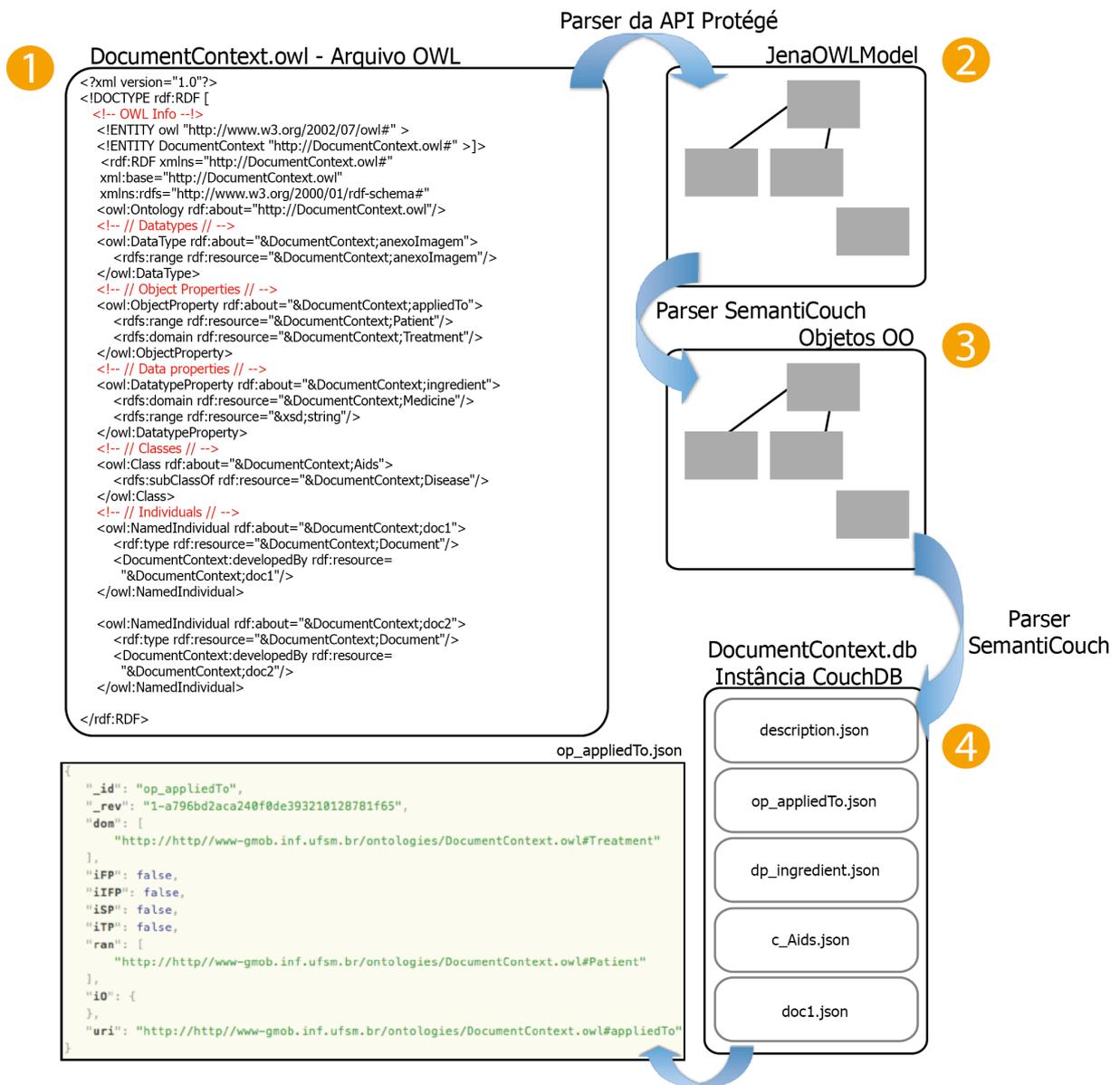


Figura 16 – Estrutura de armazenamento de um arquivo OWL no banco de dados

Para executar a conversão de formatos, são realizados os seguintes passos pela arquitetura:

- (1) Uma ontologia é definida e salva em um arquivo no formato OWL. Utilizam-se arquivos gerados automaticamente pelo software *Protégé*, seguindo o padrão OWL v1.0 (W3C, 2011);
- (2) Através da indicação de qual arquivo deseja ser convertido, a arquitetura utiliza o *Parser* da API *Protégé* (STANFORD, 2011) para realizar a conversão do arquivo OWL original para um objeto Java do tipo *JenaOWLModel*, definido na API do *Protégé*, o qual mantém as informações sobre uma ontologia;
- (3) O *parser* definido no SemantiCouch realiza uma varredura pelos objetos relacionados a entidades de uma ontologia no objeto *JenaOWLModel* e converte todas as informações destes objetos para instâncias das classes definidas no pacote *OO*;
- (4) O *parser* definido na arquitetura SemantiCouch cria uma nova instância no banco de dados CouchDB e armazena as informações dos objetos no banco de dados no formato JSON, sendo cada objeto armazenado em forma de documento,.

A divisão da estrutura do arquivo OWL original de uma ontologia permite que o sistema faça consultas carregando apenas os dados necessários para o mesmo.

Por exemplo, caso uma entidade externa queira consultar apenas os dados referentes à estrutura de classes da ontologia, o sistema irá fazer a pesquisa somente no arquivo correspondente, sem precisar consultar outros arquivos, que representam os indivíduos da ontologia.

Após o processo de conversão, uma instância do banco de dados armazena os documentos referentes as entidades da ontologia convertida. A Figura 17 apresenta um exemplo de visualização da instância do banco de dados CouchDB após a conversão de uma ontologia.

Cada entidade é representada no banco de dados como um documento único, que contém informações de acordo com cada tipo de entidade inserida. Cada documento possui um campo *_id* e um campo *_rev* únicos. O campo *_id* se refere ao nome único da entidade na estrutura do banco de dados, tendo como base uma *flag* determinando o tipo da entidade e o seu nome, baseado na *URI* da entidade. Já o campo *_rev* é gerado e utilizado automaticamente pelo banco de dados CouchDB para o controle de concorrência e a replicação de documentos entre diversos nós em uma rede.

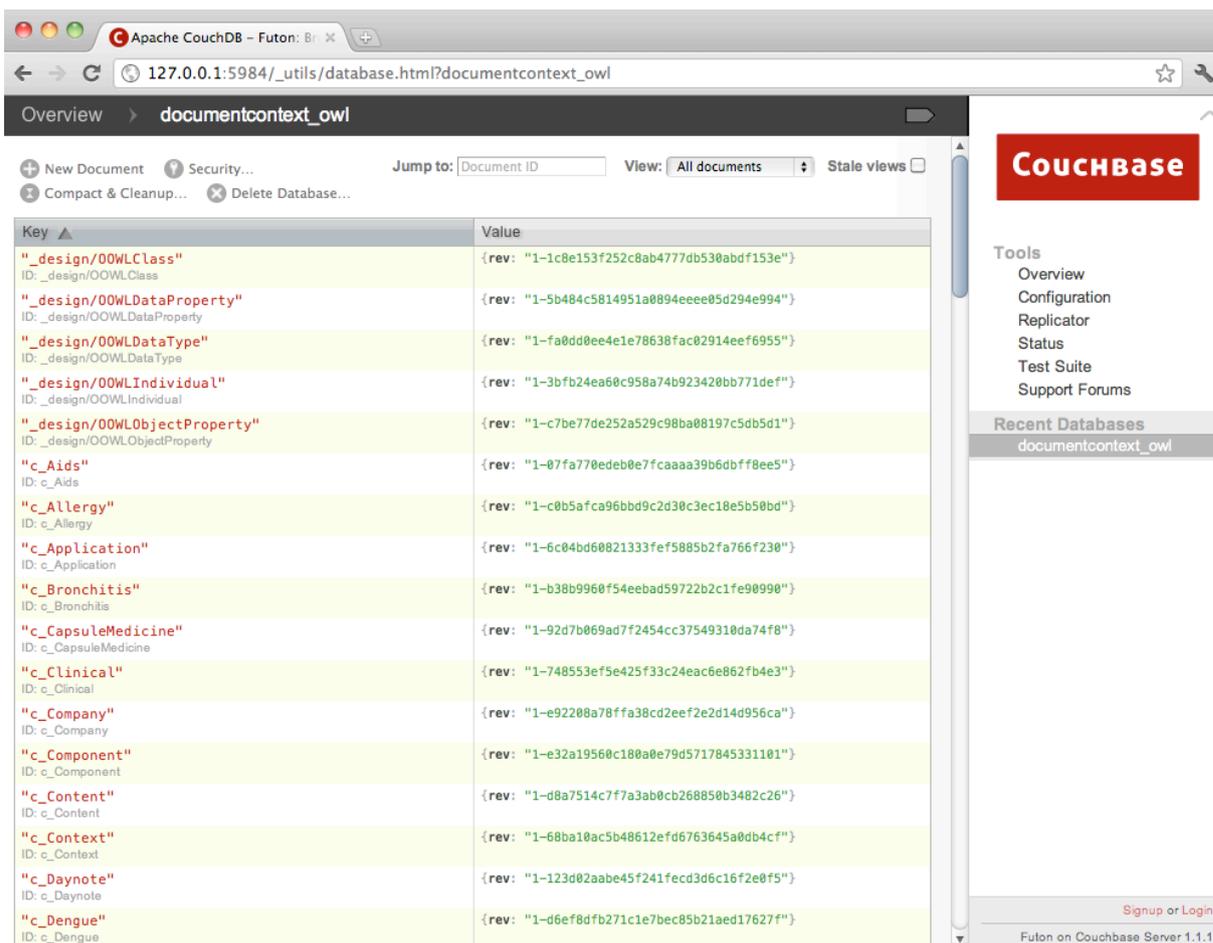


Figura 17 – Documentos em uma instância do banco de dados CouchDB.

A Tabela 5 apresenta as *flags* utilizadas nos nomes de documentos pela arquitetura.

Tabela 5 – Relação de *flags* utilizadas para nomear documentos no banco de dados

Flag	Representa	Exemplo de Nome de Arquivo
C_	Classe	C_Paciente
DT_	Tipo de Dado	DT_AnexoDeImagem
DP_	Propriedade de Dado	DP_Nome_Paciente
OP_	Propriedade de Objeto	OP_Possui_Exame
I_	Indivíduo	I_ViniciusMaran
description	Arquivo de Cabeçalho da Ontologia	description

Além da conversão de um arquivo OWL para uma instância do banco de dados, a arquitetura permite a conversão das informações armazenadas no banco de dados para um arquivo OWL.

A Figura 18 ilustra os passos da API para a conversão de uma instância do banco de dados para um arquivo OWL. Passo 1, verifica se a instância informada como fonte da conversão existe; Passo 2, lê todos os documentos presentes no banco de dados e converte cada documento para um objeto definido no pacote *OO* de acordo com o seu tipo; Passo 3, cria um novo objeto da classe *JenaOWLModel* e insere neste modelo todas as informações relacionadas a ontologia que estão definidas no objeto *OOWLModel*; Passo 4, através do *parser* definido na API *Protégé*, a estrutura do objeto *JenaOWLModel* é salva em um arquivo, representado por um objeto da classe *File*. Deste modo, a estrutura da ontologia é salva como um arquivo OWL e pode ser utilizada por outros sistemas que utilizam este formato.

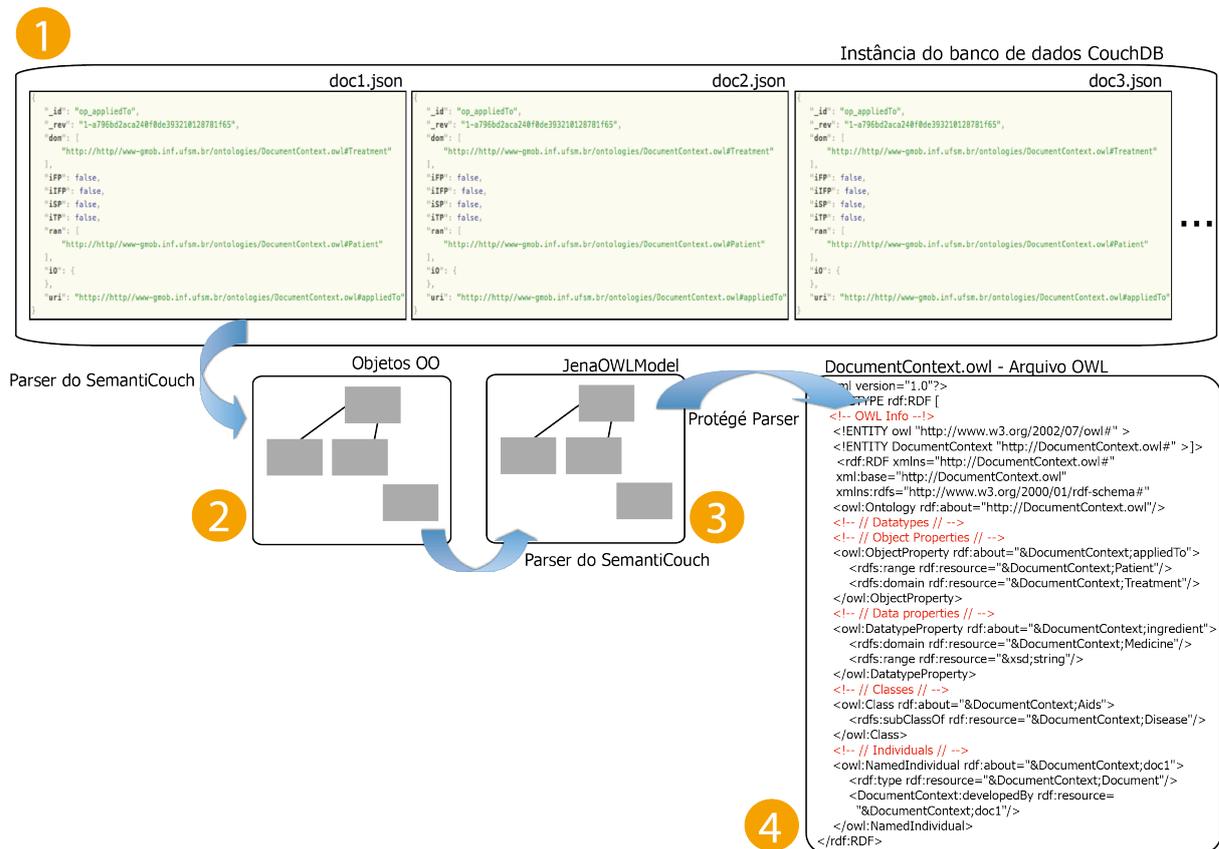


Figura 18 – Processo de conversão de uma instância do banco de dados para um arquivo OWL

O processo de conversão - importação de um arquivo OWL - é opcional. Pode-se criar uma instância do SemantiCouch de duas formas: (a) o desenvolvedor cria uma instância do objeto *OOWLModel*, que define uma ontologia no banco de dados e a partir deste realiza operações na ontologia, e (b) o desenvolvedor utiliza os métodos de conversão para converter uma ontologia já existente - em formato OWL ou RDF - para documentos *JSON*, que,

posteriormente, são inseridos no banco de dados e associados a um objeto *OOWLModel*, permitindo a sua modificação.

A partir do momento que o objeto *OOWLModel* é criado e associado a uma instância do banco de dados, a arquitetura proporciona a possibilidade da realização de inserções e consultas de dados nas ontologias persistidas no banco de dados. O processo de inserção e consulta é descrito na próxima seção.

4.1.3 Inserção e Consulta

A realização de operações na arquitetura SemantiCouch pode ser conduzida de duas formas: (a) através dos métodos pré-definidos na API, ou (b) através de consultas baseadas na linguagem SWRL e SQWRL.

Para a realização de operações diretamente no modelo - representado por um objeto *OOWLModel* pode-se utilizar os métodos pré-definidos na API. Desta forma, a arquitetura consulta as informações existentes no banco de dados e retorna as entidades requisitadas. Por exemplo, para realizar a inserção de uma entidade na ontologia, pode-se definir uma nova entidade através de um objeto *OOWLEntity* e, então, usar o método *insertEntity* da classe *OOWLModel* para inserir esta entidade na ontologia. Neste momento, a arquitetura verifica automaticamente se as informações da entidade estão em conformidade com a representação da ontologia, como se os *namespaces*¹ estão descritos na ontologia armazenada.

Os métodos pré-definidos na classe *OOWLModel* para as operações em entidades na ontologia fazem ligações entre o código Java executado para a operação e o banco de dados CouchDB através de *views*², que são constantemente atualizadas pelo CouchDB (ANDERSON et al, 2010). A Figura 19 apresenta a lista dos métodos pré-definidos na classe *OOWLModel* referentes à consulta de entidades em uma ontologia.

¹ *Namespace*: fornecem um método simples para a qualificação de nomes de elementos e atributos usados em documentos XML (W3C, 2011).

² *Views*: são a principal ferramenta usada para consultar bancos de dados e relatórios no banco de dados CouchDB. Eles são definidos em linguagem *javascript* (embora existam outras formas de representação) (ANDERSON et al, 2010).

```

○ getAllClasses() : ArrayList<OOWLClass>
○ getAllDataProperties() : ArrayList<OOWLDataProperty>
○ getAllDataTypes() : ArrayList<OOWLDataType>
○ getAllIndividuals() : ArrayList<OOWLIndividual>
○ getAllObjectProperties() : ArrayList<OOWLObjectProperty>
○ getClassByName(String cla) : OOWLClass
○ getClassByURI(String URI) : List<OOWLClass>
○ getClassCount() : int
○ getDataPropertyByName(String obj) : OOWLDataProperty
○ getDataPropertyByURI(String URI) : List<OOWLDataProperty>
○ getDataTypeByName(String name) : OOWLDataType
○ getDataTypeByType(String type) : List<OOWLDataType>
○ getDataTypeByURI(String URI) : List<OOWLDataType>
○ getDatabaseName() : String
○ getIndividualByName(String individual) : OOWLIndividual
○ getIndividualByURI(String URI) : List<OOWLIndividual>
○ getIndividualCount() : int
○ getJenaOWLModel() : JenaOWLModel
○ getModelDescription() : OOWLDescription
○ getOWLThingClass() : OOWLClass
○ getObjectPropertyByName(String obj) : OOWLObjectProperty
○ getObjectPropertyByURI(String URI) : List<OOWLObjectProperty>
○ getSubClassesOf(OOWLClass clas) : List<OOWLClass>

```

Figura 19 – Métodos definidos na classe *OOWLModel* para a consulta de informações da ontologia

Os métodos definidos na classe *OOWLModel* permitem que sejam consultadas informações sobre entidades específicas presentes no banco de dados. Estes métodos realizam chamadas ao banco de dados CouchDB, retornando as entidades desejadas. A Figura 20 apresenta:

(a) a definição de um método de consulta no banco de dados, com a definição da *view* utilizada no banco de dados CouchDB. Neste exemplo, apresenta-se o método que realiza a consulta de todas as propriedades de dados ligadas a um indivíduo em específico – consultado pelo seu nome no banco de dados; e

(b) um exemplo da chamada deste método por um programa que utiliza o SemantiCouch. No exemplo, o método retorna uma lista com todas as propriedades de dados do indivíduo de nome “*paciente_vinicius_maran*”.

```

a @View( name = "dpByDomain",
  map = "function(doc) { if (doc.type == 'OOWLDataProperty' && doc.dom) "
      + "emit( doc.dom, doc )}" )
  public List<OOWLDataProperty> getDPByDomain(String URI) {
      return queryView("dpByDomain", URI);
  }

b OOWLModel my_model = new OOWLModel(conexao_couchdb,
  "documentcontext_owl", true);
  my_model.getDPByDomain("paciente_vinicius_maran");

```

Figura 20 – (a) método de consulta de entidade; (b) exemplo de chamada do método

Para a realização da distribuição das informações da ontologia entre diversos nós de uma rede, sejam eles nós fixos ou móveis, a arquitetura utiliza os mecanismos de replicação e controle de versões do banco de dados CouchDB.

4.1.4 Distribuição de Informações

Para realizar o compartilhamento de informações entre nós de uma rede distribuída, a arquitetura utiliza a API Ektorp, que gerencia os nós do banco de dados CouchDB. A distribuição de dados é realizada através da configuração de um servidor para onde as informações devem ser replicadas. Este servidor, considerado como *master*, realiza a replicação de dados entre outros servidores com bases de dados CouchDB.

Na arquitetura SemantiCouch, o processo de replicação é realizado através da classe *OOWLReplicator*, a qual possui um objeto associado a classe *OOWLModel*. Deste modo, são oferecidos dois métodos principais para a replicação de dados. O método apresentado na Figura 21 realiza a replicação da ontologia (inclusive de seus indivíduos) para um endereço de um servidor que possua o banco de dados CouchDB em execução.

```

OOWLModel.replicOntoToNode(String nodeLink, boolean
                           isContinuous)

```

Figura 21 - Replicação da ontologia pela arquitetura SemantiCouch

Além disso, é possível escolher se a replicação acontecerá apenas no momento em que

o método é chamado, ou se a replicação será contínua; ou seja, o banco de dados CouchDB manterá as informações sincronizadas automaticamente entre os nós utilizados para a replicação. Já o método apresentado na Figura 22 realiza apenas a replicação dos objetos *OOWLClass*, *OOWLObjectProperty*, *OOWLDataProperty*, *OOWLDescription* e *OOWLDataType*; ou seja, apenas a estrutura de representação da ontologia, sem as instâncias dos indivíduos.

```
OOWLModel.replicDefToNode(String nodeLink, boolean
                          isContinuous)
```

Figura 22 - Replicação das definições da ontologia pela arquitetura SemantiCouch

Desta forma, a mesma estrutura pode ser utilizada entre diversos nós em uma rede, mas cada nó terá as suas instâncias de indivíduos em particular, sem acesso pelos outros nós da rede.

Por padrão, o banco de dados CouchDB realiza a replicação de dados em formato de círculo, conforme ilustra a Figura 23.

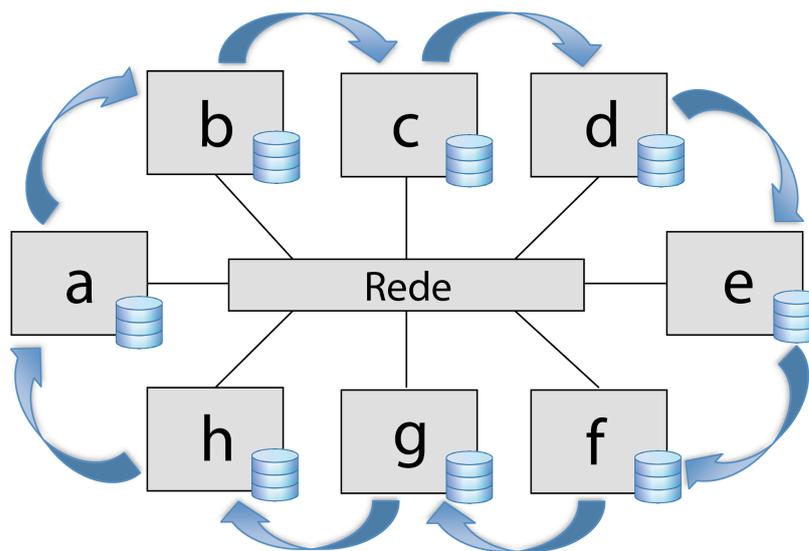


Figura 23 – Exemplo de replicação de documentos pelo banco de dados *CouchDB*

Caso um nó fique *offline* por um determinado motivo, as informações são automaticamente replicadas entre os outros nós e serão sincronizadas com este nó *offline* assim que ele estiver *online* novamente na rede. Este processo é realizado automaticamente

pelo banco de dados *CouchDB*.

A configuração em relação a replicação de dados pode ser feita de forma direta no banco de dados CouchDB através dos utilitários de configuração do banco de dados.

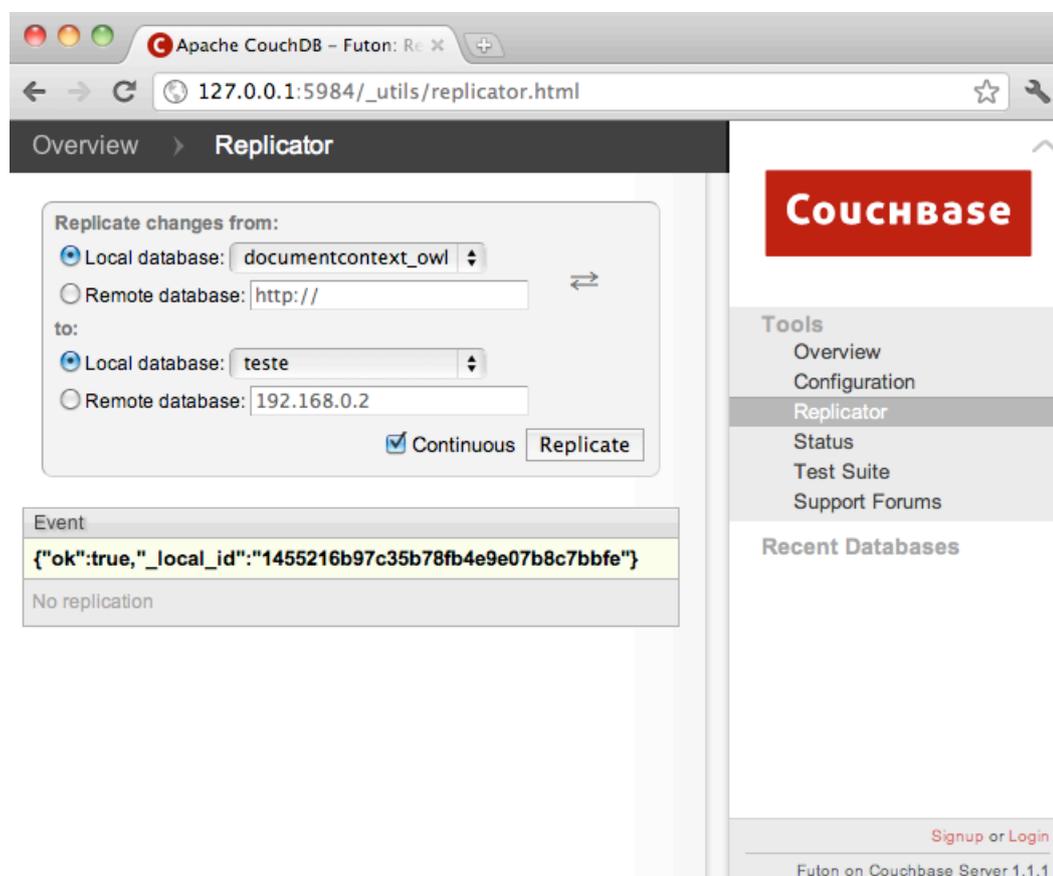


Figura 24 – Exemplo de configuração da replicação de instâncias pelo utilitário *Futon*.

A Figura 24 apresentou um exemplo de configuração de replicação contínua de uma instância do banco de dados CouchDB para outra instância local do banco de dados, utilizando o utilitário de configuração *Futon* (ANDERSON et al, 2010). Desta forma, todas as alterações realizadas em qualquer uma das instâncias do banco de dados serão sincronizadas com a outra instância do banco de dados.

4.2 Seleção Contextualizada de Documentos Clínicos

Após a definição de como as informações de contexto e informações clínicas são

representadas na arquitetura ClinicSpace e da definição da arquitetura para a persistência de ontologias, foi definido o serviço de persistência de contexto e seleção contextualizada de documentos clínicos, que: (i) realiza a ligação entre a arquitetura SemantiCouch e os serviços da arquitetura ClinicSpace para as requisições de persistência e recuperação de contexto, e (ii) realiza o processo de inserção e consulta de documentos baseado no contexto informado no momento da consulta pela arquitetura ClinicSpace.

A primeira etapa para a definição do módulo de seleção foi a modelagem das mudanças visuais na IETC, anteriormente definida na arquitetura ClinicSpace; para a inserção de documentos com associação de contexto e para a consulta de informações clínicas sobre pacientes.

4.2.1 Alterações na IETC

O primeiro passo na integração do serviço de consulta de documentos na arquitetura ClinicSpace foi modificar algumas características da IETC definida por Silva (2009) e modificada por Machado (2010).

Em relação à persistência de dados de contexto, nenhuma modificação foi realizada, pois a interface visual da IETC, modificada por Machado (2010), já apresentava a possibilidade de informar e consultar dados de contexto. Então, as mudanças para a integração com o serviço proposto foram realizadas apenas no código de execução. A Figura 25 apresenta a interface modificada por Machado (2010) e utilizada atualmente na arquitetura ClinicSpace.

Para a inserção de documentos no sistema, criou-se uma interface separada, para a inserção de documentos e a associação de elementos de contexto a este documento (até o momento, é realizada de forma manual). A interface prototipada permite com que o usuário associe informações de contexto ao documento e permite que o usuário salve as informações referentes a este documento.

Em termos de distribuição geográfica, o módulo da IETC está alocado em clínicas e laboratórios do ambiente hospitalar, para a inserção de informações clínicas. Porém, a associação de informações de contexto é gerada pela própria arquitetura ClinicSpace, através da captação de dados de contexto através de sensores.

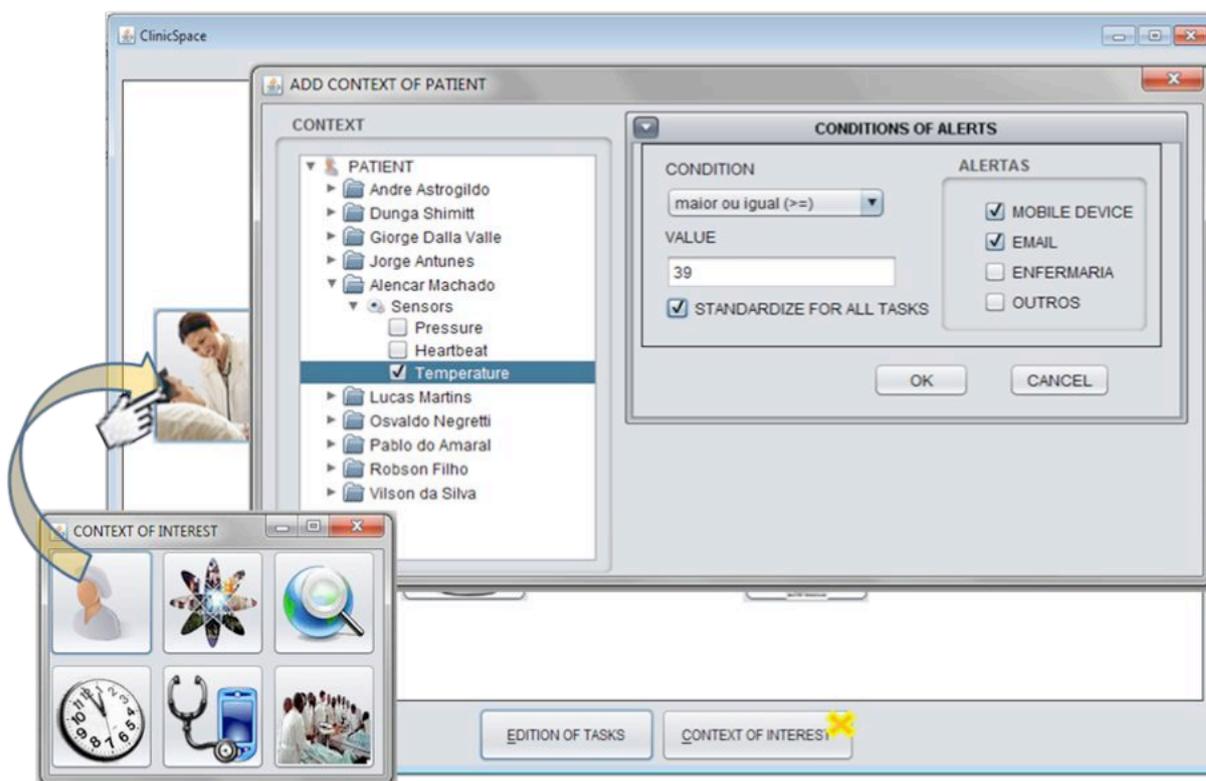


Figura 25 – Associação de informações de contexto ao paciente através da IETC
(MACHADO, 2010)

A Figura 26 apresenta o protótipo de inserção de documentos e associação de contexto aos documentos clínicos. Este protótipo se comunica diretamente com o serviço SGT, fazendo as requisições de inserção e associação de contexto a arquitetura ClinicSpace.

A partir deste módulo, o usuário pode realizar as seguintes operações: (1) selecionar qual tipo de documento deve ser criado; (2) selecionar quais elementos de contexto devem ser associados a este documento; (3) adicionar anexos como imagens ou textos ao documento; e (4) editar as partes de cada documento. Após a realização destas operações, o sistema converte as informações para instâncias da ontologia e armazena na arquitetura de persistência SemantiCouch.

A cada inserção de documento, uma nova versão é criada na ontologia, pois um mesmo documento pode ter mais de uma versão (formato). As versões se distinguem pelas informações de contexto associadas, por exemplo, um exame de sangue de um determinado paciente pode ter duas versões distintas, onde uma é associada a um determinado dispositivo e outra é associada a outro tipo de dispositivo.

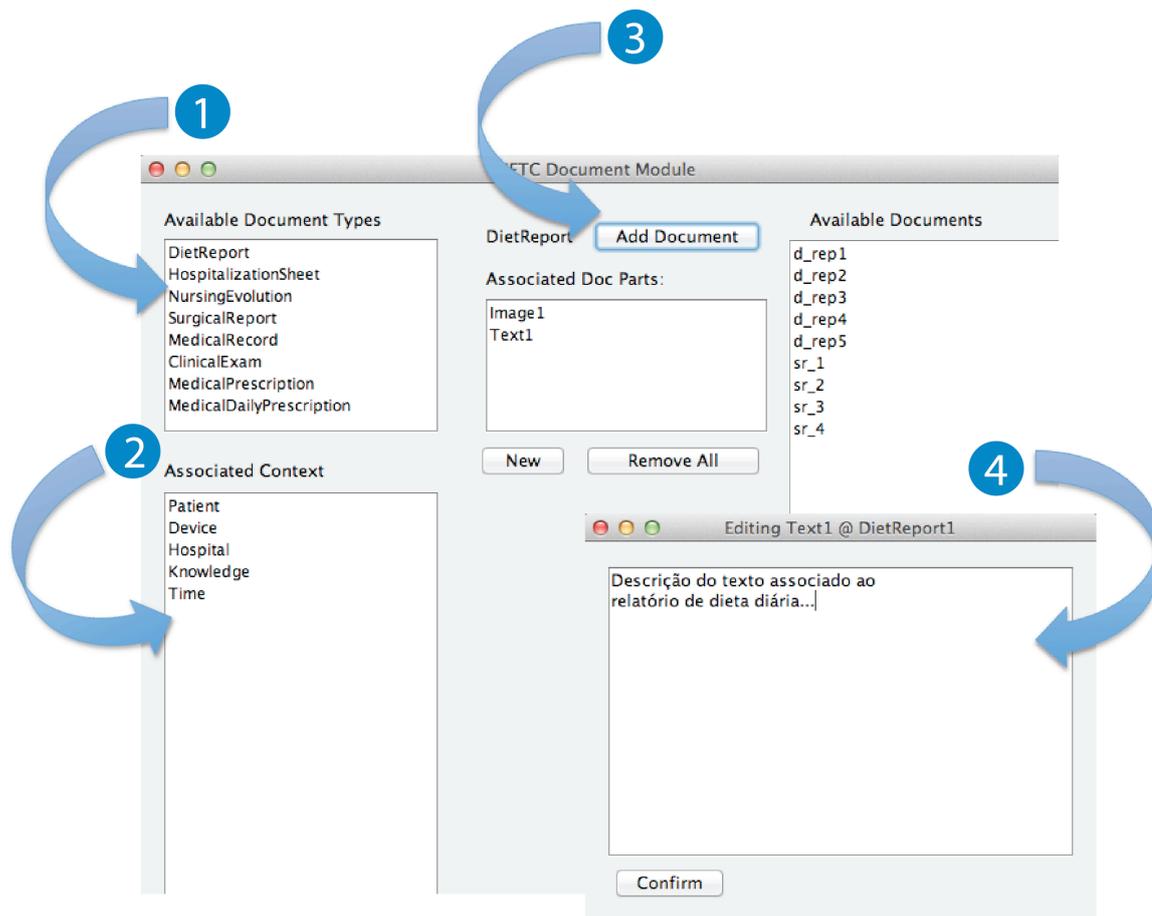


Figura 26 – Protótipo do módulo de inserção de documentos e associação de contexto

Um elemento foi adicionado à IETC para a consulta de documentos, onde o médico informa elementos de contexto referentes ao documento, e o sistema consulta documentos de acordo com o contexto informado.

4.2.2 Associação de Informações de Contexto

A cada versão de um documento clínico que é inserido no sistema, são associados os elementos de contexto relevantes¹ para futuras consultas realizadas pelo sistema. Tais informações de contexto são comparadas às informações da consulta, e o serviço decide se a versão consultada é a mais próxima do contexto informado².

¹ A associação de elementos de contexto é feita de forma manual pelo usuário.

² Até o momento, o serviço compara se há outras versões de documentos que possuam elementos de contexto da mesma classe associada. Ele não verifica se o elemento de contexto tem propriedades ou características próximas

Além desta característica, o serviço implementa mecanismos de permissões de acessos aos documentos. Este mecanismo de acesso tem como objetivo fazer com que apenas determinados exames sejam acessados por determinados médicos, onde o médico possui acesso aos exames relacionados à sua especialidade médica ou a especialidades que possuem alguma referencia com a sua especialidade médica. O mecanismo é implementado através de propriedades de classes definidas na ontologia, onde um documento só tem o seu acesso permitido se os elementos de contexto específicos - o médico e o paciente - possuem a propriedade *hasAccess* descrita na sua representação ontológica.

Caso o médico queira acessar o resto do histórico de exames e documentos relacionados ao paciente, ele apenas consulta o histórico completo pelo IETC, pois o mecanismo de acesso apenas prioriza o acesso às informações de acordo com a especialidade.

4.2.3 Versionamento de Documentos

Para realizar o controle de como as informações de contexto são armazenadas com documentos, optou-se pela utilização de versionamento de documentos. Desta forma, cada documento inserido no sistema possui diferentes versões para a mesma informação, variando de acordo com elementos de contexto.

Todos os documentos armazenam informações de contexto, de acordo com os elementos de contexto informados pelo ClinicSpace, ao serviço no momento da inclusão de um documento no sistema. Porém, novas versões de documentos são criadas apenas quando os elementos de contexto Paciente – representado pela classe *Patient* na ontologia - e Dispositivo – representado pela classe *Device* - são novos se comparados aos armazenados no banco de dados, variando em relação aos outros elementos de contexto.

Esta característica se deve ao fato de que estes dois elementos de contexto influenciam diretamente na modelagem da informação. O elemento de contexto Paciente define para qual paciente o documento foi criado; desta forma, todas as informações dos documentos são baseadas em um paciente. Já o elemento de contexto Dispositivo modela a capacidade de representação e utilização de dispositivos para o armazenamento e visualização de documentos clínicos. Por exemplo, um mesmo exame pode ser representado em forma de

imagens em um dispositivo que possui um *display* com esta capacidade, e em forma de texto em outro dispositivo que não possui *display* colorido.

Nesta primeira versão de implementação, as versões de documentos são criadas manualmente pelo usuário, que identifica o paciente a quem o documento se refere e a qual dispositivo específico a informação é adaptada. Assim, o serviço associa estas informações ao documento que é armazenado na arquitetura ClinicSpace. Estas informações são utilizadas posteriormente na comparação realizada com elementos de contexto informados no momento da consulta aos documentos.

Para as outras informações de contexto (*Application, Time, Disease, Localization, Medicine, MaterialForClinicExamination, Knowledge e Prognostic*), o serviço realiza operações sobre as informações clínicas. Como por exemplo avisa o médico caso ele esteja consultando um exame com informações críticas e outras pessoas estão na mesma sala. Porém não precisa armazenar novas versões de documentos uma vez que essas são efetuadas automaticamente, em tempo de execução, pelo serviço. Com isso, evita-se a redundância de dados gerada por diferentes versões de um mesmo documento clínico.

4.2.4 Processo de Inserção de Documentos

A primeira etapa na seleção contextualizada de documentos acontece na inserção destes documentos no sistema, conforme ilustra a Figura 27. Quando o documento é inserido no sistema, alguns dos elementos de contexto – como a especialidade médica envolvida em cada parte do documento – são adicionados explicitamente (1), já outras informações são associadas diretamente pelo ambiente ClinicSpace (2) – como a localização onde o exame foi realizado e o formato das informações representadas no documento. Após a inserção do documento no sistema, o serviço associa este documento aos elementos de contexto e armazena estas informações na ontologia (3) e insere os dados na arquitetura de persistência, replicando as informações entre as instituições (4).

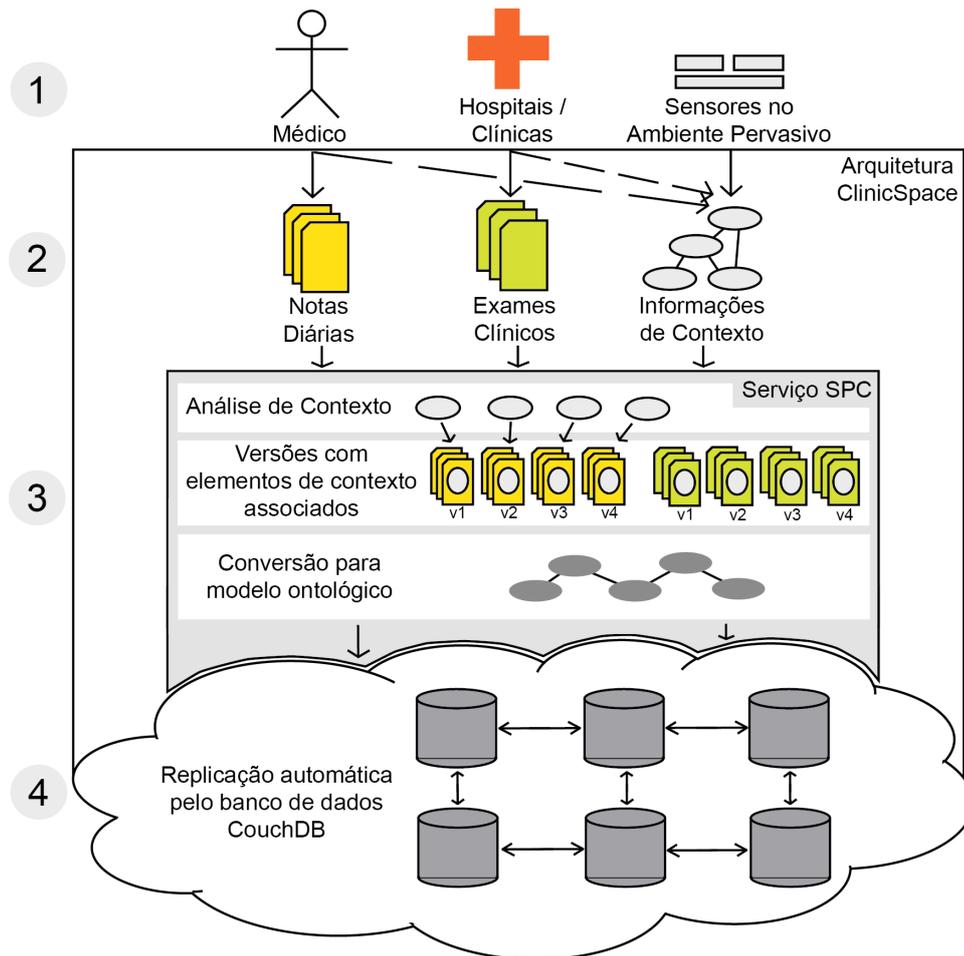


Figura 27 – Processo de Inserção de um documento no sistema

A partir da inserção destas informações na arquitetura, elas estarão disponíveis em diversos locais, de acordo com as configurações de replicação da arquitetura SemantiCouch¹. Portanto, estas informações podem ser acessadas em outras instituições hospitalares e em dispositivos de diversos tipos. A partir deste momento, as informações podem ser modificadas ou consultadas de acordo com as necessidades da arquitetura ClinicSpace.

4.2.5 Processo de Consulta de Documentos

Para realizar a seleção contextualizada de documentos, o serviço recebe informações de contexto da consulta, e qual documento é consultado. Estas informações de consulta são

¹ O controle da replicação de documentos é feito pelo banco de dados CouchDB.

modeladas como indivíduos de uma ontologia, que tem suas informações comparadas às persistidas no banco de dados. A partir destas informações, o serviço escolhe qual versão (dentre as versões geradas automaticamente) se encaixa nos elementos de contexto informados. Se uma versão específica deste contexto não é encontrada, o serviço procura por outras versões que pertençam a mesma classe da versão consultada, realizando uma busca e comparação entre as classes definidas na ontologia. A Figura 28 ilustra este processo.

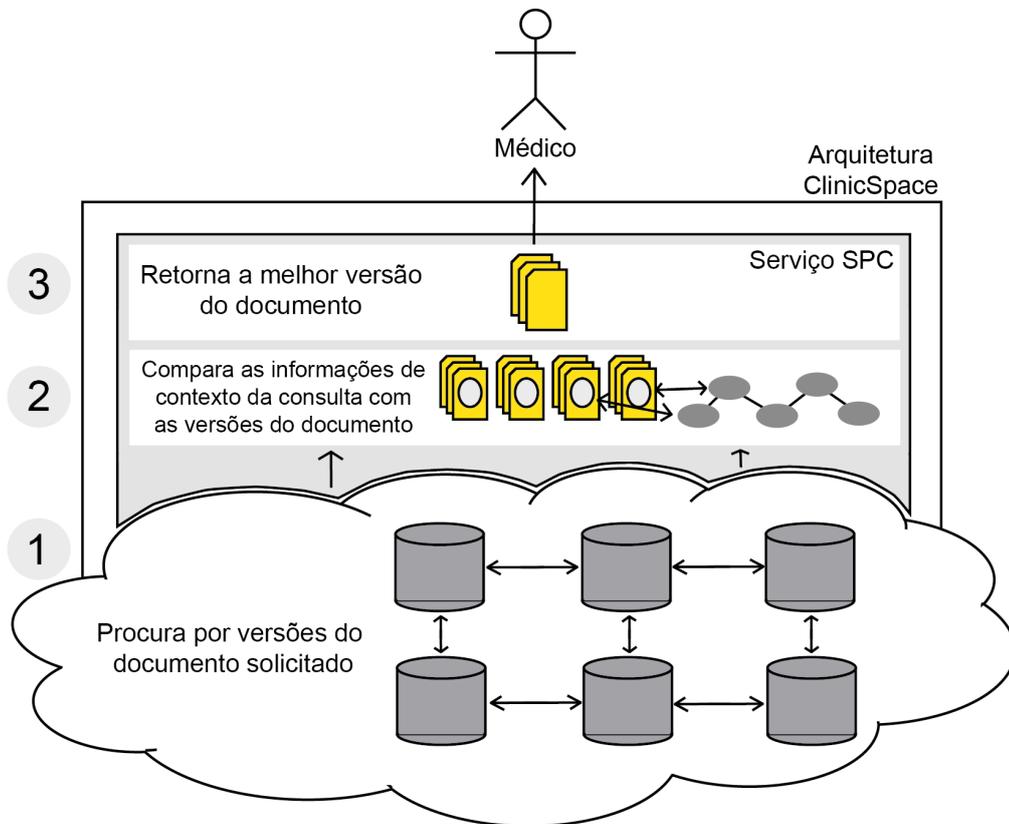


Figura 28 – Processo de consulta de um documento no sistema

Como apresentado na Figura 28, quando uma requisição de consulta é feita, o serviço procura por versões de documentos que possuam informações de contexto associadas que condizem com o contexto informado na consulta (1). Após a realização desta etapa, o serviço compara as informações de contexto dos documentos (2) e retorna a versão que possui elementos de contexto associados semelhantes aos elementos informados na consulta¹ (3).

¹ Foi implementada a comparação as mesmas instâncias de contexto. A comparação entre classes é sugerida como trabalho futuro na Seção 6.2.

5 TESTES REALIZADOS E CASO DE USO

Após a definição e implementação do serviço, realizaram-se testes com os seguintes objetivos:

- Avaliar se as consultas realizadas retornam resultados esperados, desta forma validando a representação das informações na ontologia;
- Avaliar a variação do tamanho dos dados em arquivos OWL e persistidos na arquitetura SemantiCouch;
- Comparar as informações relacionadas a tempo de inserção e tamanho dos dados após as inserções em relação ao trabalho de Machado (2010).

Além destes testes, definiu-se um cenário de uso para validar o serviço em conjunto com a arquitetura ClinicSpace.

5.1 Testes de Representação de Contexto

A primeira etapa na realização dos testes diz respeito à capacidade de representação da ontologia descrita na Seção 3.1. Para testar a capacidade de representação da ontologia proposta, realizou-se um conjunto de consultas e inferências que se considera essenciais para a utilização da ontologia em um ambiente hospitalar.

Para a realização de consultas, utilizou-se a linguagem SQWRL e a ferramenta *Protégé*. Inicialmente, definiu-se um conjunto de consultas e regras de inferência baseadas em um estudo realizado em parceria com um hospital universitário do Rio Grande do Sul, onde foram elencadas as tarefas e consultas mais comuns realizadas em sistemas hospitalares.

Após a definição das consultas e regras de inferência, povoou-se a ontologia desenvolvida com indivíduos de classes variadas. Estes indivíduos possuem relações entre si que foram descritas utilizando propriedades.

Em uma primeira etapa de testes, realizou-se consultas simples na ontologia, buscando obter informações sobre um determinado grupo de indivíduos. A seguir, apresentam-se as consultas realizadas na ontologia para testes, o código SQWRL referente às consultas realizadas e os resultados obtidos nestas consultas.

- **Consulta 1.** Deseja-se saber o número de pacientes adultos que estão presentes no hospital no momento da consulta (considerando que pacientes adultos possuem mais de 17 anos). O trecho de código SQWRL escrito para realizar esta consulta na ontologia é apresentado na Figura 29.

```
Patient(?p) ^ Hospital(?h) ^
isIn(?p, ?h) ^ personAge(?p, ?age) ^
swrlb:greaterThan(?age, 17) ^ sqwrl:makeSet(?s, ?p) ^
sqwrl:size(?size, ?s) -> sqwrl:select(?size)
```

Figura 29 - Código SQWRL da Consulta 1

Como resultado, a consulta retornou o número de pacientes que estavam no hospital e que tinham a idade maior que 17 anos.

A Consulta 2 representa um exemplo de consulta baseada em um indivíduo específico da ontologia, consultando dados e indivíduos que possuam alguma ligação específica com este indivíduo.

- **Consulta 2.** Deseja-se obter os nomes dos pacientes que estão sendo tratados atualmente pelo indivíduo da classe *Doctor* com nome “*Doutor_Carlos*”. O trecho de código SQWRL escrito para realizar esta consulta na ontologia (a) e o resultado apresentado pelo *software Protégé* (b) são apresentados na Figura 30.

(a)

```
Patient(?p) ^ Doctor(Doutor_Carlos) ^
treatedBy(?p, Doutor_Carlos) ^
personName(?p, ?name) ->
sqwrl:select(?name)
```

(b)

?name
paciente dois
paciente tres
paciente um

Figura 30 – Consulta 2

Na consulta 3, testa-se a obtenção de diversos dados relacionados a indivíduos de uma mesma classe.

- **Consulta 3.** Deseja-se saber o número de identificação único (representado pelo campo *id*), o nome e a idade dos pacientes que estão nas dependências do hospital. O trecho de código SQWRL escrito para realizar esta consulta na ontologia (a) e o resultado da consulta (b) são apresentados na Figura 31.

<p>(a)</p> <pre>Patient(?p) ^ Hospital(?h) ^ personName(?p, ?name) ^ isIn(?p, ?h) ^ personAge(?p, ?age) ^ personID(?p, ?id) -> sqwrl:select(?id, ?name, ?age)</pre>	<p>(b)</p> <table border="1"> <thead> <tr> <th>?id</th> <th>?name</th> <th>?age</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>paciente dois</td> <td>30</td> </tr> <tr> <td>5</td> <td>paciente cinco</td> <td>50</td> </tr> <tr> <td>1</td> <td>paciente um</td> <td>17</td> </tr> <tr> <td>3</td> <td>paciente tres</td> <td>45</td> </tr> </tbody> </table>	?id	?name	?age	2	paciente dois	30	5	paciente cinco	50	1	paciente um	17	3	paciente tres	45
?id	?name	?age														
2	paciente dois	30														
5	paciente cinco	50														
1	paciente um	17														
3	paciente tres	45														

Figura 31 – Resultados da Consulta 3

Apesar de proporcionar a obtenção de dados da ontologia, as consultas não contemplam a inserção ou modificação do conhecimento representado na ontologia. Desta forma, torna-se necessária a utilização de inferências, que permitem a adição ou alteração de um conhecimento presente na ontologia.

Para a realização de inferências sobre a ontologia proposta, elaborou-se algumas constatações para testar se a ontologia, através de regras de inferência, era capaz de realizar a validação destas constatações.

Em seguida, montou-se um conjunto de regras, referentes a estas constatações, utilizando a linguagem SWRL, e realizaram-se os testes de inferência, utilizando o software *Protégé* versão 3.4.4 e o plugin *Jess* na versão 7.1 (JESS, 2010).

- **Inferência 1.** Definiu-se uma regra que estabelece que todos os parentes de pacientes que estão no mesmo ambiente hospitalar que o paciente tem o direito de acessar as informações de diagnóstico deste paciente. O trecho de código apresentado Na Figura 32 descreve o código SWRL referente à regra definida nesta inferência.

```
Hospital(?h1) ^ Hospital(?h2) ^ Patient(?p) ^
PatientsParent(?pp) ^ isIn(?p, ?h1) ^
isIn(?pp, ?h2) ^ isParentOf(?pp, ?p) ^
sameAs(?h1, ?h2) ->
haveInfoOf(?pp, ?p)
```

Figura 32 - Código SWRL da inferência 1

Como resultado para esta inferência, alterações foram realizadas na ontologia, criando a ligação *haveInfoOf* entre os parentes de um paciente e o paciente, permitindo assim que estes parentes tenham permissão de acesso às informações sobre o paciente no ambiente hospitalar pelo sistema pervasivo.

- **Inferência 2.** Definiu-se uma regra que diz respeito ao tratamento diferenciado do sistema a uma pessoa que está no ambiente hospitalar e está sendo monitorada por um sensor. A partir desta premissa, a pessoa passa a ser tratada como paciente pelo

sistema hospitalar. O trecho de código apresentado na Figura 33 descreve o código SWRL referente a regra definida nesta inferência.

```
Person(?p) ^ Hospital(?h) ^
isIn(?p, ?h) ^ SensorComponent(?s) ^
monitoredBy(?p, ?s) -> Patient(?p)
```

Figura 33 - Código SWRL da inferência 2

Após a inferência utilizando a regra apresentada, todas as pessoas que estavam no hospital e estavam sendo monitoradas por um sensor passaram a pertencer a classe *Patient* e, conseqüentemente, tratadas pelo sistema hospitalar como pacientes.

Com a realização dos testes de representação foi comprovado que a ontologia criada representa os principais elementos de contexto e informações clínicas frequentemente utilizadas por médicos.

A próxima etapa de testes se refere ao armazenamento de contexto utilizando a arquitetura SemantiCouch, onde (i) se utilizou o arquivo OWL da ontologia criada para a conversão em uma instância de banco de dados, (ii) replicou-se esta instância entre os nós de uma rede e (iii) realizou-se consultas diversas na ontologia presente no banco de dados para simular situações de uso real do sistema.

5.2 Testes de Armazenamento de Contexto

A partir da confirmação de que a ontologia desenvolvida atende aos requisitos de representação de contexto, realizaram-se testes voltados à inserção, consulta e replicação de dados no serviço de persistência proposto. Inicialmente, realizaram-se testes de inserção da ontologia variando o número de classes, indivíduos e relações definidas na ontologia.

Todos os testes realizados foram feitos em um computador *Apple Macbook Pro*, processador *Core i5 2.3ghz*, 4GB de memória RAM e HD de 500GB, utilizando a versão 1.1.1 do banco de dados CouchDB com as configurações *default*.

Para realizar a inserção de ontologias no sistema de persistência, criou-se uma interface prototipada para o auxílio nas tarefas sobre ontologias persistidas na arquitetura SemantiCouch.

A Figura 34 apresenta a interface de operação prototipada.

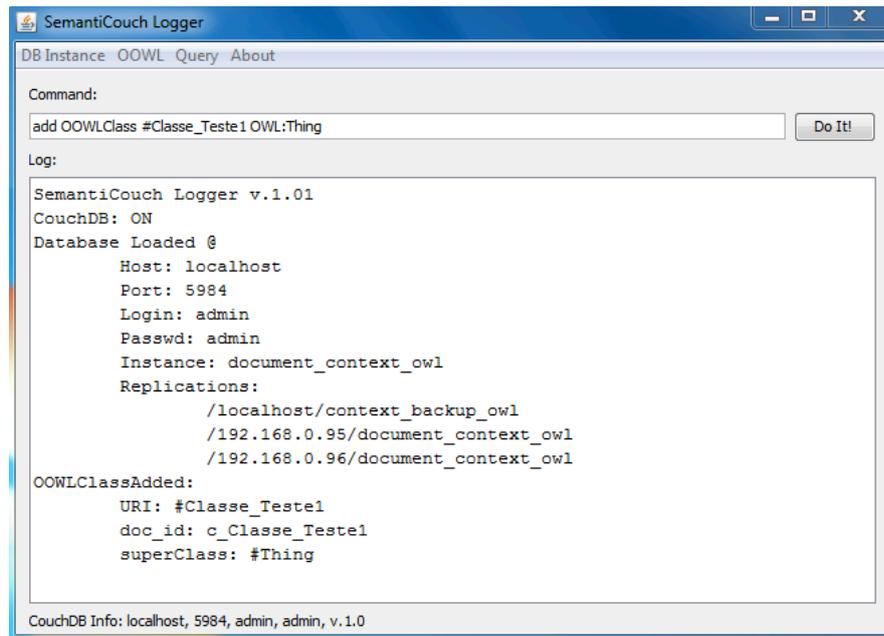


Figura 34 – Interface de testes da arquitetura *SemantiCouch*

Na primeira etapa de testes, realizou-se a conversão e inseriu-se a ontologia proposta na Seção 3.1, variando o número de indivíduos descritos na ontologia. O resultado dos testes de conversão e inserção são apresentados na Figura 35.

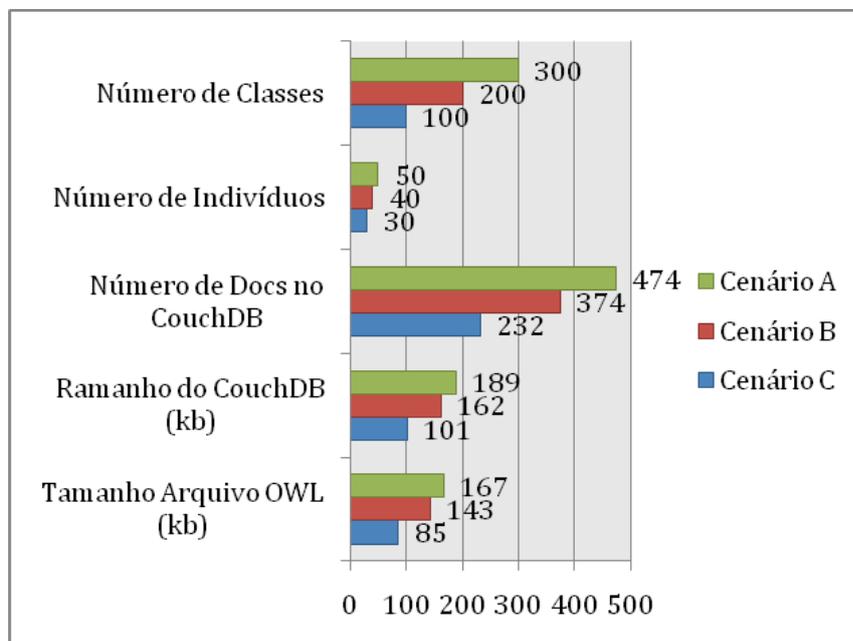


Figura 35 – Resultados dos testes de inserção de ontologias

Como se pode observar nos resultados, a inserção dos arquivos OWL no banco de dados gerou um pequeno aumento de tamanho dos arquivos de ontologias originais (aproximadamente 13%). Este aumento está diretamente relacionado às indexações realizadas automaticamente pelo CouchDB para o aumento de velocidade em consultas posteriores. Além disso, pode-se observar que o número de documentos no banco de dados se manteve alto, mostrando alta fragmentação, uma das características desejáveis para auxiliar a replicação automática de documentos realizada pelo CouchDB.

Após esta primeira etapa, o sistema torna-se capaz de realizar consultas e outras operações nesta ontologia. A arquitetura do banco de dados CouchDB permite que os dados da ontologia sejam replicados em outros nós e, em caso de falha de um destes, a ontologia pode ser acessada e modificada nos outros nós, com a replicação feita posteriormente de forma automática pelo banco de dados.

Para a realização dos testes de replicação, configurou-se a replicação de uma instância do banco de dados que contem os dados de uma ontologia entre diversos nós de uma mesma rede, e mediu-se o tempo necessário para a replicação destes dados.

A replicação dos dados foi realizada de forma correta¹ nos nós que continham instâncias do banco de dados CouchDB. Em relação aos tempos de replicação dos dados, não se conseguiu medi-los de forma precisa e sem interferências, pois esta etapa é realizada diretamente pelo banco de dados CouchDB, sem interferência da aplicação gerenciada pela arquitetura SemantiCouch. Realizando consultas manuais nos servidores, não se presenciou erros de versionamento gerados pelo banco de dados CouchDB, o que indica que a replicação de dados foi feita de forma correta e rápida.

Após a realização dos testes de replicação de dados, realizaram-se testes em relação às consultas de dados, representados na ontologia, no banco de dados.

O processo de consulta é realizado pelo sistema *Query for Objects* da API Ektorp (EKTORP, 2011), onde se consulta por chaves específicas de objetos persistidos. Desta forma, realizou-se a consulta de um exame de sangue relacionado a um determinado paciente utilizando o código apresentado na Figura 36.

¹ O sistema não apresentou erros nas consultas realizadas após a replicação dos dados.

```

OOWLIndividual vm =
my_model.getindividualByName("ViniciusMaran");
OOWLIndividual bloodExam = null;
if (vm.getObjectProperties().contains("http://www-
gmob.inf.ufsm.br"+
"/ontologies/DocumentContext.owl#has")) {
bloodExam = vm.getRange(("http://www-gmob.inf.ufsm.br"+
"/ontologies/DocumentContext.owl#has")
}

```

Figura 36 - Consulta de dados na arquitetura SemantiCouch

A partir deste momento, a variável *bloodExam* contém as referências do indivíduo relacionado a um exame - propriedade *has* ao indivíduo com nome “*ViniciusMaran*”. Portanto, os desenvolvedores podem realizar alterações diretamente no objeto relacionado e depois executar o processo de atualização deste objeto no banco de dados, o qual ocorre com uma simples chamada de método.

O código da Figura 37 apresenta um exemplo de carregamento de um indivíduo, a troca de nome deste indivíduo e a sua atualização no banco de dados.

```

OOWLIndividual bExam =
my_model.getindividualByName("bloodExamVM");
if (bExam != null) {
    bExam.setName("bloodExamVMM");
    bExam.update(); }

```

Figura 37 - Operações realizadas em um indivíduo na ontologia

Como resultado dos testes de persistência e consulta de dados de contexto, o sistema retornou as informações requisitadas e inseriu novas informações quando solicitado nos testes.

5.3 Caso de Uso

Para validar a utilização da arquitetura ClinicSpace em conjunto com o serviço proposto neste trabalho, definiu-se um cenário de uso, baseado no estudo de caso proposto por Machado (2010) e considerando a seguinte situação: a arquitetura ClinicSpace é utilizada em um hospital, tanto em nós fixos, nos computadores dos profissionais clínicos, quanto em seus

dispositivos móveis. Assim, os médicos realizam suas atividades diárias com informações acessíveis em qualquer dispositivo dentro do ambiente hospitalar.

O hospital possui uma arquitetura de armazenamento baseada no SemantiCouch para armazenar informações clínicas do ambiente hospitalar e informações de contexto que são gerenciadas pela arquitetura ClinicSpace. Os pacientes possuem todos os seus exames e documentos armazenados e compartilhados com outras instituições hospitalares que também utilizam esta arquitetura.

Um médico possui uma rotina diária de tarefas definida na arquitetura ClinicSpace. Desta forma, a arquitetura realiza as operações baseando-se no fluxo de tarefas definidas pelo médico. Em um determinado momento, um médico especialista em neurologia é designado a atender um determinado paciente que, ao entrar no hospital, é identificado automaticamente pela leitura de contexto realizada pela arquitetura ClinicSpace e possui exames realizados recentemente na área de neurologia.

A partir do momento em que o paciente é atendido pelo médico, a arquitetura faz uma leitura geral do contexto. Através desta leitura, a arquitetura realiza a identificação do paciente (que neste cenário possui 16 anos), do médico, e das pessoas que estão presentes com o paciente no momento da consulta (neste cenário, a mãe do paciente). O médico então associa outras informações de contexto ao paciente, tais como a leitura de seus sinais vitais através da IETC (Figura 38).

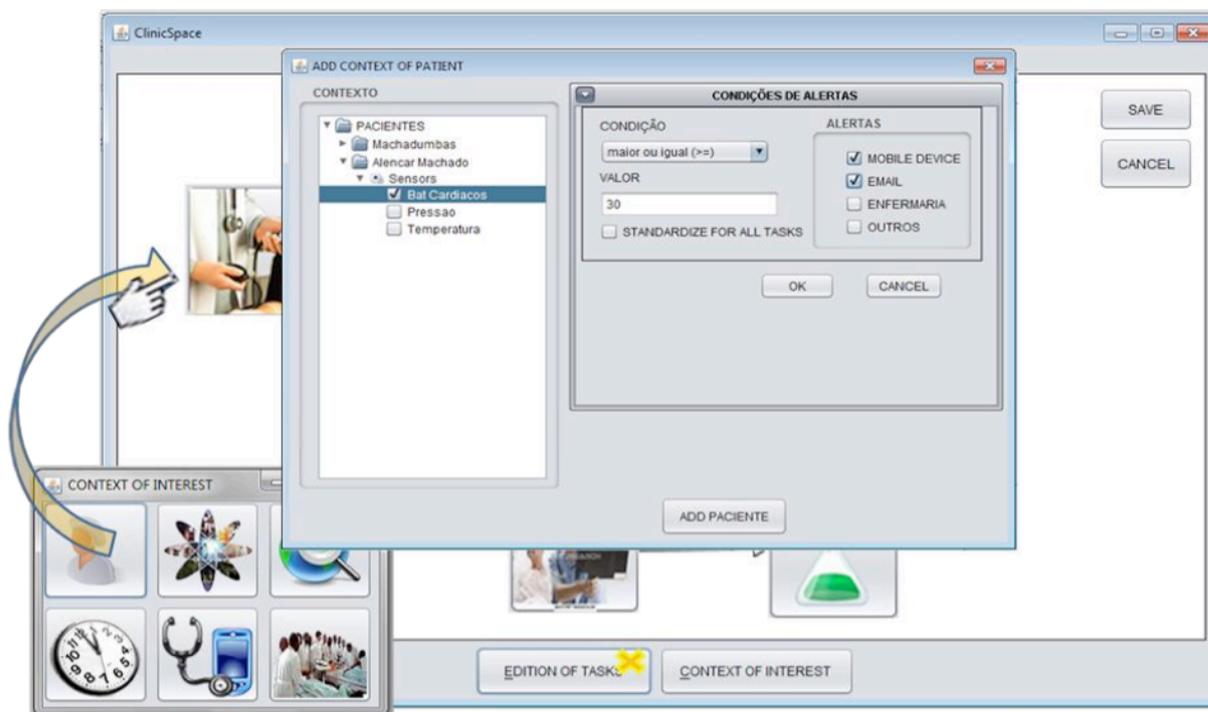


Figura 38 – Associação de informações de contexto ao paciente através da IETC
(MACHADO, 2010)

A arquitetura possui todas as informações clínicas dos pacientes armazenadas e acessíveis em qualquer dispositivo dentro do hospital, porém o médico deseja consultar apenas:

- as informações referentes aos exames neurológicos do paciente;
- as informações de uma forma adaptada ao dispositivo que ele está utilizando no momento da consulta (no caso deste cenário de uso, um *smartphone*);
- as informações de qualquer ambiente hospitalar onde o paciente já realizou exames, mesmo sem saber onde este paciente já realizou exames.

Para que isto aconteça, o médico utiliza a interface IETC e realiza os seguintes procedimentos:

- **Modela a tarefa desejada (a).** Neste caso de uso, o médico dividiu o atendimento ao paciente em cinco subtarefas: (i) verificação dos sinais vitais do paciente; (ii) verificação do histórico de saúde do paciente; (iii) análise de exames laboratoriais realizados pelo paciente; (iv) diagnóstico do paciente; e (v) prescrição de receita médica ao paciente;
- **Associa as informações de contexto (b).** O médico associa informações de contexto

necessárias na realização das tarefas, tais como pressão arterial e batimentos cardíacos do paciente.

No momento em que o médico faz a consulta aos documentos, na subtarefa análise dos exames laboratoriais realizados pelo paciente, ele insere uma nota referente a um exame em específico a insere no sistema através da IETC. Após estas etapas, ele realiza o atendimento ao paciente.

Para representar este caso de uso, populou-se uma ontologia com diversas informações relacionadas a pacientes, médicos, especialidades e documentos gerados de forma manual (Figura 39).

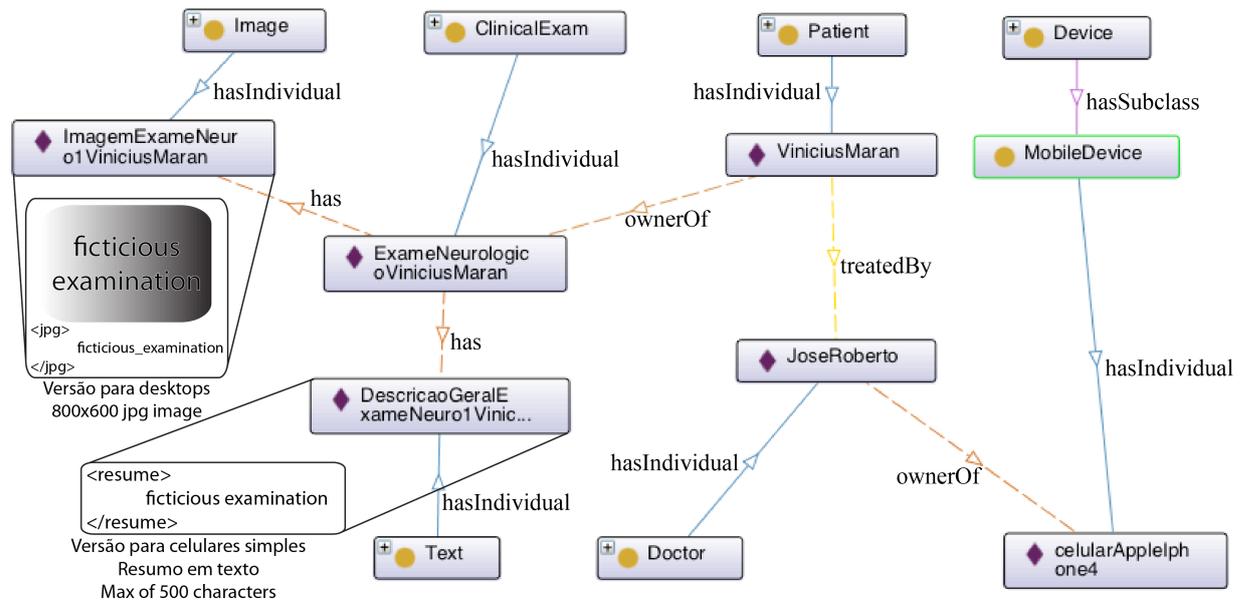


Figura 39 – Indivíduos criados para o teste de seleção de documentos

Além da inserção de dados de contexto e dados de pacientes e médicos, criaram-se exames com imagens fictícias para inserção na base de dados, representando exames realizados por determinados pacientes¹.

A Figura 40 apresenta um exemplo de exame fictício em diferentes versões de imagem.

¹ As versões de imagens para diferentes resoluções de *display* foram criadas manualmente e inseridas no sistema.

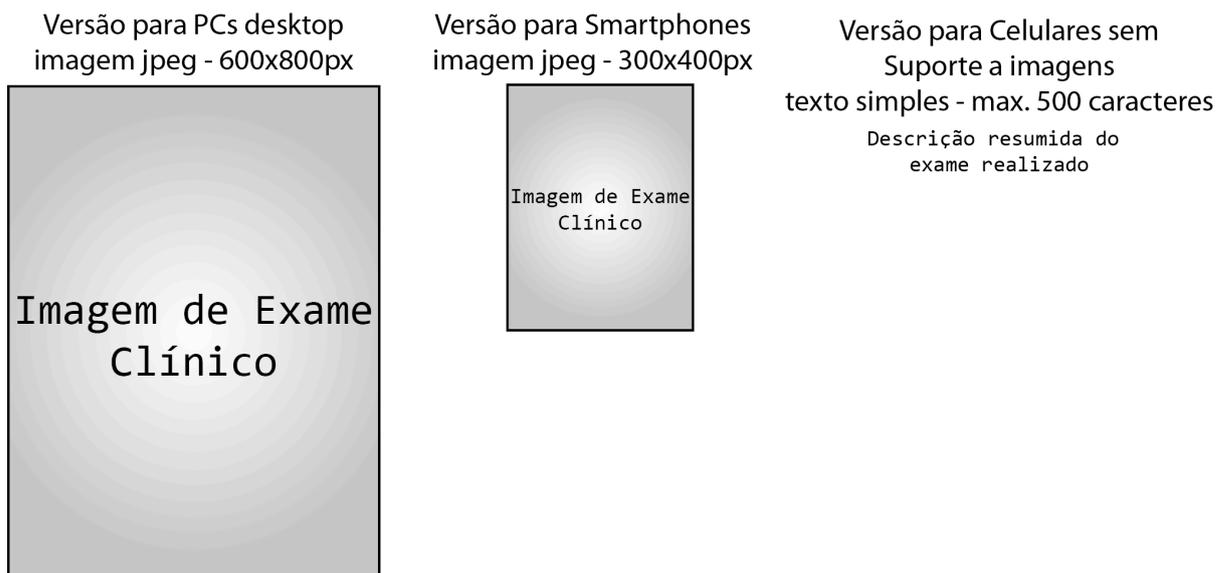


Figura 40 – Exemplo de exame clínico fictício criado em diferentes versões em relação a resolução de tela

Após a criação destas informações, criaram-se instâncias da arquitetura *SemantiCouch* e replicaram-se as informações das instâncias do banco de dados em três servidores distintos, que replicavam as informações entre si continuamente, conforme ocorriam mudanças nos documentos no banco de dados.

Comparou-se o tempo de carregamento das aplicações necessárias para a execução das tarefas na arquitetura ClinicSpace utilizando a persistência de contexto e com a última solução utilizada na arquitetura ClinicSpace (MACHADO, 2010). Neste teste, foram sendo criadas instâncias de médicos e estas eram inseridas gradativamente na arquitetura. O resultado está mapeado na Figura 41.

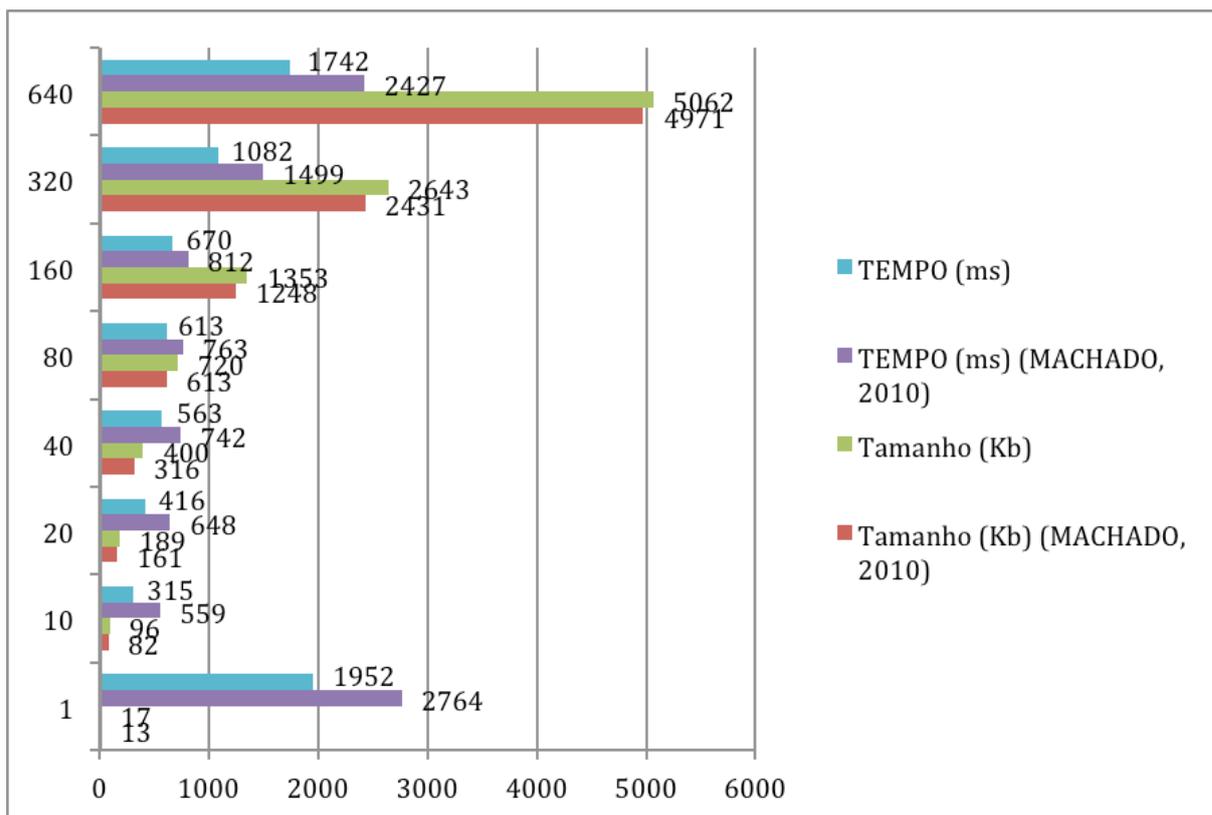


Figura 41 – Comparação entre os tempos de carregamento do sistema com as alterações realizadas

Como se pode observar, o tempo de carregamento do sistema reduziu aproximadamente 28% com a inserção do serviço proposto neste trabalho. Esta mudança se deve principalmente pela persistência dos dados da ontologia em um banco de dados. O tempo de carregamento com 1 único médico é maior porque nesta etapa os serviços do *middleware* EXEHDA são iniciados. Após o carregamento do EXEHDA, as instâncias dos médicos foram sendo adicionadas gradativamente.

Desta forma, (i) os dados não precisam ser carregados em memória toda vez que o sistema é iniciado, e (ii) os dados de contexto e dados clínicos dos pacientes podem ser utilizados de forma distribuída e acessível em diversos tipos de dispositivos, mantendo-se sincronizados automaticamente.

6 CONCLUSÕES

A utilização de sistemas ubíquos na área de *healthcare* torna o ambiente hospitalar mais inteligente e centrado no profissional clínico, podendo contribuir para diminuir o fator de rejeição dos sistemas hospitalar por profissionais clínicos. A proposta do projeto ClinicSpace é prover uma infra-estrutura para que aplicações e serviços possam auxiliar os profissionais clínicos na realização personalizada de tarefas em sistemas hospitalares.

Dentro do escopo do projeto ClinicSpace, o serviço de persistência de contexto e seleção contextualizada de documentos teve como principal objetivo oferecer meios de armazenamento e recuperação de dados, auxiliando no processo de adaptação de aplicações e de conteúdo, conforme o contexto, e apresentado de forma personalizada ao usuário.

A integração entre as definições ontológicas de contexto e de conhecimento clínico com as vantagens da utilização de um banco de dados facilmente escalável, permite com que a arquitetura ClinicSpace utilize informações compartilhadas entre uma rede pervasiva.

A criação de uma nova abordagem para a persistência de ontologias (arquitetura SemantiCouch) foi necessária graças as necessidades de sistemas ubíquos para a persistência de contexto, as quais não foram encontradas em outras ferramentas de persistência.

6.1 Contribuições

As contribuições deste trabalho para a arquitetura ClinicSpace são:

- Modelagem ontológica de elementos de contexto. Para uma identificação mais completa sobre elementos de contexto que podem interferir na realização de tarefas e seleção de documentos, expandiu-se a capacidade de representação de elementos de contexto, originalmente definida por Machado (2010);
- Modelagem ontológica de conhecimento clínico representado em documentos. Criou-se uma definição ontológica que representa os documentos clínicos mais frequentemente utilizados na realização de atividades por profissionais clínicos. Deste modo, o conhecimento representado nesta ontologia pode ser utilizado entre diferentes sistemas computacionais;

- Interface gráfica de associação dos elementos de contexto e consulta de documentos clínicos. Desenvolveu-se módulos que, acoplados na interface já existente do ClinicSpace, permitem aos profissionais clínicos consultar documentos e os alterarem de acordo com o contexto do momento da consulta;

- Arquitetura de persistência distribuída SemantiCouch. Através do desenvolvimento da arquitetura SemantiCouch, conseguiu-se persistir os dados representados de forma ontológica em um banco de dados facilmente escalável e distribuído. Desta forma, os dados clínicos de pacientes são replicados entre instituições hospitalares e podem ser acessados em diferentes contextos;

- Serviço de seleção contextualizada de documentos para a arquitetura ClinicSpace. Através da consulta do contexto informado na consulta de documentos, o serviço de seleção contextualizada seleciona a versão de documentos adequados ao contexto para a visualização e retorna o documento ao profissional clínico.

Cada item acima resultou em uma reestruturação nos serviços existentes do ClinicSpace, que antes realizavam as operações baseados em uma ontologia utilizada em memória de forma não compartilhada.

6.2 Trabalhos Futuros

Os trabalhos futuros propostos na arquitetura abrangem:

- expansão da ontologia definida para representar mais elementos de contexto relevantes e mais informações clínicas de acordo com pesquisas realizadas no ambiente hospitalar, como a representação do contexto emocional de pacientes e médicos, sendo capaz de medir o nível de stress de pessoas no ambiente clínico;

- definição de um módulo para a utilização da arquitetura SemantiCouch em sistemas distribuídos utilizando a tecnologia de *webservices* e utilização da API em dispositivos móveis, tornando-a independente da arquitetura ClinicSpace, podendo ser utilizada por outros sistemas de saúde;

- elaboração de regras de inferência específicas para a sua utilização em ambientes reais;

- definição de um *Reasoner* capaz de realizar inferências e consultas nos formatos SWRL e SQWRL, convertendo a linguagem utilizadas nas consultas para as operações já disponíveis na arquitetura SemantiCouch;
- melhoria na integração do serviço com a interface IETC e teste da usabilidade da arquitetura ClinicSpace em conjunto com o serviço proposto em um ambiente real de testes.

6.3 Publicações

As publicações até o momento relativas ao trabalho são:

CLEI 2011: Maran, V., Machado, A., Brum Saccol, D., Augustin, I. *"Um serviço de persistência de contexto e seleção contextualizada de documentos para a arquitetura ClinicSpace"*. Em: XXXVII Conferencia Latinoamericana de Informática (XXXVII CLEI). Quito – Equador.

IADIS WWW/INTERNET 2011: Maran, V., Machado, A., Brum Saccol, D., Augustin, I. *"A Software Architecture to Provide Persistence and Retrieve of Context Data Based on Ontological Models"*. Em: IADIS International Conference WWW/Internet 2011. Rio de Janeiro – Brasil.

IADIS WWW/INTERNET 2011: Maran, V., Machado, A., Brum Saccol, D., Augustin, I. *"Adding Context Data Persistence Based on Ontologies in an Ubiquitous Healthcare Architecture"*. Em: IADIS International Conference WWW/Internet 2011. Rio de Janeiro – Brasil.

Revista Perspectivas da Ciência da Informação: Maran, V., Machado, A., Brum Saccol, D., Augustin, I. *"Uma Representação Ontológica de Informações e Documentos Clínicos Relevantes na Realização de Atividades em um Ambiente Hospitalar"*. Brasil. (em processo de revisão).

RITA – Revista de Informática Teórica Aplicada: Maran, V., Machado, A., Brum Saccol, D., Augustin, I. *"Uma Arquitetura de Armazenamento Distribuído de Dados de Contexto Baseados em Modelos Ontológicos"*. Brasil (em processo de revisão).

Referências Bibliográficas

ANDERSON, J. C.; LEHNARDT, J.; SLATER, N. **CouchDB: The Definitive Guide**. Livro. Ed O'Reilly Media. ISBN: 978-0-596-15589-6. 272 pag., 2010

AMERICAN SOCIETY FOR TESTING AND MATERIALS **Standard Specification for Continuity of Care Record (CCR)** ASTM Standard ASTM E2369 – 05e1 –. DOI: 10.1520/E2369-05E01, 2011.

AUGUSTIN, I.; LIMA, J. C. D.; YAMIN, A. C. **Computação Pervasiva: como Programar Aplicações**. Em: X Simposio Brasileiro de linguagens de programação (SBLP), 2006, Itatiaia, RJ. SBLP, 2006.

AUGUSTIN, I.; YAMIN, A.; GEYER, C. F. R. **Managing the follow-me semantics to build large-scale pervasive applications**. Em: Proceedings of the 3rd International Workshop on Middleware for Pervasive and Ad-Hoc Computing, 2005, France, p. 1-8.

AURA AURA WEBSITE. Website. Disponível em: www.cs.cmu.edu/~aura. Acessado em Março de 2011.

BARDRAM, J. E. **Hospitals of the Future Ubiquitous Computing Support for Medical Work in Hospitals**. Em: Proceedings of the 5th International Conference in Ubiquitous Computing, 2003, USA.

BARDRAM, J. E.; CHRISTENSEN, H. B. **Pervasive Computing Support for Hospitals: An overview of the Activity-Based Computing Project**. Em: IEEE Pervasive Computing, vol. 6, issue 1, p. 44-51, 2007.

BATZIOS, A.; MITKO, P.A **db4OWL: An Alternative Approach to Organizing and Storing Semantic Data**. Em: IEEE Internet Computing, vol. 13, no. 6, pp. 48-55, Nov./ Dec, 2009.

BETTINIA, C.; BRDICZKAB, O.; HENRICKSEN, K.; INDULSKAD, J.; NICKLASE, D.; RANGANATHAN, A.; RIBONI, D. **A survey of Context Modelling and Reasoning Techniques**. Em: *Pervasive and Mobile Computing*, 6(2), 161-180, 2010.

BOUQUET, P.; GIUNCHIGLIA, F.; HARMELEN, V.L.; SERAFIINI, H. **Contextualizing Ontologies**. Em: *Journal of Web Semantics* 1 (4), 2004.

CHEN, H.; PERICH, F.; FININ, T. W.; JOSHI, A. **Standard Ontology for Ubiquitous and Pervasive Applications**. Em: 1st Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous 2004), IEEE Computer Society, 2004.

COUCHBASE **CouchBase Mobile**. Website. Disponível em: <http://www.couchbase.com/products-and-services/couchbase-mobile>. Acessado em Março de 2011.

DEY, A.; ABOWD, G. **The Context Toolkit: Aiding the Development of Context-Aware Applications**, Em: *Proceedings of Human Factors in Computing Systems: CHI 99*, Pittsburgh, PA: ACM Press, pp.434-441. 2006.

EKTORP **EKTORP Website**. Website. Disponível em: <http://www.ektorp.org/>. Acessado em Fevereiro de 2011.

FERREIRA, G. L. **Adicionando ao Middleware Exehda o Suporte a Aplicações Orientadas a Atividades Humanas**. Monografia (Mestrado em Computação) Universidade Federal de Santa Maria, Santa Maria, RS, 2009.

FERREIRA, G.; LIBRELOTTO, G.; SILVA, L.; YAMIN, A. **Middleware for management of end-user programming of clinical activities in a pervasive environment** Em: *Workshop on Middleware for Ubiquitous and Pervasive Systems*, Vol.389, pp.07-12. 2009.

GAIA **THE GAIA PROJECT**. Website. Disponível em: <http://www.cs.uiuc.edu/gaia>. Acessado em Março de 2011.

GAMMA, E. et al., **Design Patterns: Elements of Reusable Object-Oriented Software**. Em: *Addison Wesley, USA*, 1995.

GMOB Grupo de Sistemas de Computação Móvel. Website. Disponível em: <http://www-gmob.inf.ufsm.br/>. Acessado em Março de 2011.

GRUBER, T. **What is an Ontology?**. Website, Disponível em: <http://www-ksl.stanford.edu/kst/what-is-anontology.html>. Acessado em Dezembro de 2010.

HENRICKSEN, K.; INDULSKA, J. **Developing context-aware pervasive computing applications: Models and approach.** Em: *Journal of Pervasive and Mobile Computing* 2(1), 37–64, 2006.

HENRICKSEN, K.; INDULSKA, J.; RAKOTONIRAINY A. **Modeling Context Information in Pervasive Computing Systems.** Em: 1st International Conference on Pervasive Computing. In: *Lecture Notes in Computer Science*, Volume 2414, pp. 167-180. Springer, 2002.

HL7 Health Level Seven International Page. Website. Disponível em: <http://www.hl7.org/>. Acessado em Março de 2011.

JESS Jess Website. Website. Disponível em: <http://www.jessrules.com/>. Acessado em Fevereiro de 2011.

JHA, A.; DESROCHES, C.; CAMPBELL, E.; DONELAN, K.; RAO, S.; FERRIS, T; SHIELDS, A.; ROSENBAUM, S.; BLUMENTHAL, D. (2009) **The Use of Electronic Health Records in U.S. Hospitals.** *In: New England Journal of Medicine.* 2009, 360:1628-1638

JSON Introducing JSON. Website. Disponível em: <http://www.json.org/>. Acessado em Março de 2011.

KROTH, M. L. **Serviço de Colaboração na Arquitetura ClinicSpace.** Dissertação de mestrado. Universidade Federal da Santa Maria, Santa Maria, RS, Brasil, 2011.

LAERUM H.; FAXVAAG, A. **Task-oriented evaluation of electronic medical records systems: development and validation of a questionnaire for physicians.** Em: *BMC Medical Informatics and Decision Making*, 1. p. 1-16.3, 2004.

MACHADO, A. **Associação do contexto de interess do usuário às atividades clínicas na arquitetura clinicspace** Monografia (Mestrado em Computação) Universidade Federal de Santa Maria, Santa Maria, RS, 2010.

MACHADO, A.; AUGUSTIN, I. **Associando Contexto as Tarefas Clínicas na Arquitetura ClinicSpace** Em: Simposio Brasileiro de Sistemas de Informação (SBSI), 2011, Salvador – BA.

MACHADO, A.; AUGUSTIN, I.; LIBRELOTTO, G. **Ciência do Contexto para Tarefas Clínicas em um Sistema de Saúde Pervasivo**. Em: Conferencia Latinoamericana de Informática, 2010.

NOSQL **NoSQL – Non Relational Universe**. Website. Disponível em: <http://nosql-database.org>. Acessado em Maio de 2011.

QUEIROZ, R. J. B. **O Uso Significativo do Prontuário Eletrônico de Saúde e o Acesso à Informação**. Website. Disponível em: <http://ciberdominiopublico.blogspot.com/2010/07/o-uso-significativo-do-prontuario.html>. Acessado em Fevereiro de 2011.

RANGANATHAN, A.; CAMPBELL, R. H. **Supporting Tasks in a Programmable Smart Home**. Em: From Smart Homes to Smart Care, vol. 15. Amsterdam: IOS Press, 2005. p. 3-10.

RAUTENBERG, S.; TODESCO, J.; STEIL, A.; GAUTHIER, F. **Uma Metodologia para o Desenvolvimento de Ontologias** Em: Revista Ciências Exatas e Naturais - RECEN -ISSN 2175-5620, Universidade Estadual do Centro-Oeste - UNICENTRO/PR, BRASIL. Vol. 10 No. 2, 2008.

RIZZETTI, T. **Framework para gerenciamento e personalização de contexto orientado a tarefas**. Monografia (Mestrado em Computação) Universidade Federal de Santa Maria, Santa Maria, RS, 2009.

ROMAN, M. et al., **Gaia: a Middleware Infrastructure to Enable Active Spaces**. Em: IEEE *Pervasive Computing*, New York, v.1, n. 4, dezembro de 2002.

SOUSA, J., GARLAN, D. **Aura: An architectural framework for user mobility in ubiquitous computing environments**. Em: The Working IEEE/IFIP Conference on Software Architecture (WICSA), 2002.

SOUZA, M. V. B. **Inferência de Atividades Clínicas na Arquitetura ClinicSpace a partir de Propriedades do Contexto**. 103f. Monografia (Mestrado em Computação) Universidade Federal de Santa Maria, Santa Maria, RS, 2010.

STANFORD UNIVERSITY **Protege WebSite**. Website. Disponível em: <http://protege.stanford.edu/>. Acessado em Março de 2011.

STRANG, T.; POPIEN, C. **A Context modelling survey**. Em: Proceedings of the Workshop on Advanced Context Modelling, Reasoning and Management as Part of UbiComp, pp.33-40, 2005.

SWRL **SWRL: A Semantic web rule language combining owl and ruleml**. Website. Disponível em: <http://www.w3.org/Submission/SWRL>. Acessado em Novembro de 2010.

UBIHEALTH **5th International Workshop on Ubiquitous Health and Wellness part os UbiComp 2010**. Setember 26th 2010, Copenhagen, Denmark. Disponível em: <http://www.create-net.org/ubint/ubihealth/index.htm>. Acessado em Março de 2011

VICENTINI, C. **PEHS: Arquitetura de um Sistema Pervasivo de Informação em Saúde Orientado às Atividades Personalizadas pelo Usuário Clínico**. Monografia (Mestrado em Computação) Universidade Federal de Santa Maria, Santa Maria, RS, 2010.

WEISER, M. **The Computer of 21st Century**, Em: Scientific American, 1991, p 3-11.

WORLD WIDE WEB CONSORTIUM W3C **OWL Web Ontology Language Guide: W3C Recommendation 10 February 2004**. Website, Disponível em: <http://www.w3.org/TR/owl-guide/>. Acessado em Dezembro de 2010.

WORLD WIDE WEB CONSORTIUM W3C **SWRL: A Semantic Web Rule Language Combining OWL and RuleML**. Website. Disponível em: <http://www.w3.org/Submission/SWRL/>. Acessado em Novembro de 2010.

WORLD WIDE WEB CONSORTIUM W3C **SemanticWeb Standards by W3c**. Website. Disponível em: <http://www.w3c.org/standards/semanticweb>. Acessado em Março de 2011.

YAMIN, A. C. **Arquitetura para um Ambiente de Grade Computacional Direcionado às Aplicações Distribuídas, Móveis e Conscientes do Contexto da Computação Pervasiva**. Tese de Doutorado. Porto Alegre, RS, Brasil, 2004.

ZHANG, D., GU T., WANG, X. **Context-aware Smart Home with Semantic Technology CONON**. Em: International Journal of Human-friendly Welfare Robotic Systems 6 (4). 12–20, 2005.

APÊNDICE A – ARQUITETURA CLINICSPACE

O projeto ClinicSpace (FERREIRA, 2009) (MACHADO, 2010) (RIZZETI, 2009) (SILVA, 2009) (SOUZA, 2010) (VICENTINI, 2010), atualmente em fase de desenvolvimento pelo grupo de pesquisas GMob na Universidade Federal de Santa Maria, possui como principal objetivo auxiliar os profissionais clínicos na realização das suas atividades em ambientes hospitalares. Para tal, o projeto visa a utilização de tecnologias provenientes das áreas de computação ubíqua e pervasiva e *End-user Programming*, permitindo com que os profissionais modelem suas tarefas de uma maneira fácil através de interfaces intuitivas. A partir destas modelagens, o sistema aloca os recursos computacionais necessários, gerenciados por um middleware específico para este ambiente pervasivo.

Motivação

Atualmente, os Sistemas Eletrônicos de Saúde – *EHS*; utilizados em ambientes hospitalares são modelados e desenvolvidos com o foco no controle administrativo e financeiro das instituições hospitalares.

Algumas pesquisas (BARDRAM; CHRISTENSEN, 2007) (QUEIROZ, 2011) demonstram que o foco em questões de administração e finanças faz com que exista uma alta taxa de rejeição ao uso desses sistemas por profissionais da saúde. Os clínicos, além de utilizarem formulários em papel paralelamente ao uso dos sistemas devem alimentar manualmente os sistemas com as informações já preenchidas em papel. Outro aspecto é que esses sistemas não oferecem suporte à interrupção e colaboração na realização de tarefas clínicas, características predominantes no ambiente hospitalar (BARDRAM; CHRISTENSEN, 2007).

Além disso, estes sistemas não permitem o compartilhamento de informações do histórico de pacientes entre diferentes instituições,. Isto dificulta o acesso a informações em caso de tratamento do paciente fora da área onde este reside, e implica na demora na coleta de informações do paciente e na redundância de dados – pois os mesmos exames realizados anteriormente pelo paciente devem ser refeitos em localidades diferentes (BARDRAM; CHRISTENSEN, 2007).

Para minimizar os efeitos causados por estes fatores de rejeição, os *EHS* devem ser modelados e construídos com o foco centrado no profissional clínico – no caso do ClinicSpace, o médico. Os profissionais devem utilizar os sistemas de forma completa, obtendo todas as informações clínicas disponíveis, através da leitura do contexto e das informações inseridas nos sistemas pelos profissionais clínicos (SILVA, 2009).

As atividades realizadas em ambientes hospitalares possuem como principais características: (i) a mobilidade – os profissionais clínicos realizam suas tarefas em diversos locais dentro da instituição hospitalar; e (ii) a colaboração no trabalho – além de realizarem atividades individuais, os profissionais atuam de forma colaborativa em muitas atividades hospitalares (BARDRAM; CHRISTENSEN, 2007).

Estas características fazem com que as atividades dos clínicos sofram constantes interrupções, para a realização de outras atividades com maior urgência ou no auxílio na a outros profissionais clínicos. Desta forma, os *EHS* devem ter uma semântica “siga-me” (AUGUSTIN; YAMIN; GEYER, 2005), fazendo com que a computação se desloque juntamente com o profissional, ou seja, disponibilizando o acesso às informações referentes ao profissional e paciente, em qualquer local que ele esteja.

Desta forma, faz-se necessária a utilização de informações de contexto, gerenciadas por sistemas ubíquos para garantir o acesso à informação, em qualquer lugar, a qualquer hora, em qualquer dispositivo e em qualquer rede disponível ao usuário (AUGUSTIN; YAMIN; GEYER, 2005).

Para auxiliar os profissionais clínicos na realização das suas atividades diárias, o projeto ClinicSpace fornece uma infraestrutura computacional, combinando a pró-atividade proporcionada pelas tecnologias da computação ubíqua e sensível ao contexto e a personalização das tarefas realizadas pelos profissionais clínicos.

Através da personalização de tarefas, os médicos podem modelar as suas atividades facilmente, sem a necessidade de conhecimento específico para a operação do sistema. Mesmo que o fluxo de trabalho mude por questões do ambiente – como por exemplo, uma interrupção no fluxo de trabalho para a realização de um novo atendimento - o sistema se adapta aos novos contextos das tarefas.

Esta adaptabilidade é equilibrada entre a pró-atividade (decisões automáticas baseadas em contexto) e a personalização (feita de forma individual por cada profissional clínico que utiliza o sistema). Todas as atividades realizadas pela arquitetura ClinicSpace são baseadas nas informações de contexto e na análise das tarefas clínicas realizadas pelos médicos no

ambiente hospitalar (SILVA, 2009).

Tarefas Clínicas

Para se integrar completamente ao fluxo de trabalho dos profissionais clínicos, a arquitetura ClinicSpace é fortemente baseada em atividades, ou seja, os processos realizados internamente no sistema levam em consideração as atividades que estão sendo realizadas no momento pelos profissionais clínicos (SILVA, 2009).

A arquitetura ClinicSpace foi projetada tendo como base a teoria da atividade (RANGANATHAN; CAMPBELL, 2005), onde uma atividade é definida como um conjunto de processos realizados por pessoas de forma colaborativa, coordenada, distribuída, em um espaço determinado. Além da definição de atividade, a teoria definida por Ranganathan & Campbell (2005) define seis componentes que relacionados criam e executam atividades, são eles: sujeito, objeto, ferramentas, regras, comunidade e colaboração.

No ambiente clínico, onde o projeto ClinicSpace atua, a teoria da atividade é aplicada possuindo como sujeito o profissional clínico, que por sua vez possui como principal objetivo tratar e realizar procedimentos nos pacientes, que atuam como objetivo da atividade (MACHADO, 2010).

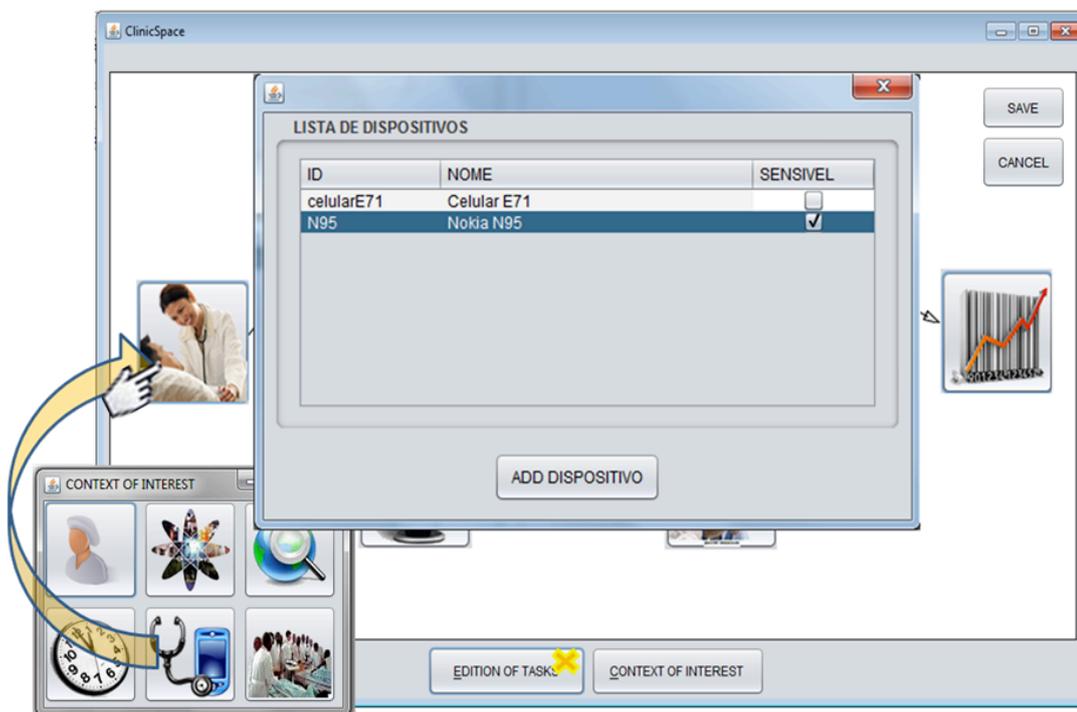
Para realizar a atividade, o médico utiliza ferramentas de mediação, as quais consistem em registros, procedimentos, equipamentos e recursos. Essa mediação, geralmente, segue regras especificadas através de guias clínicos. A atividade é executada no ambiente clínico, sendo esta a sua comunidade, onde há uma diversidade de profissionais que, normalmente, fazem uma divisão do trabalho para a sua realização (MACHADO, 2010).

Ao modelar essas noções na arquitetura ClinicSpace, a atividade clínica foi decomposta em um conjunto de tarefas e subtarefas – ações que tem o auxílio de recursos computacionais; e seguem a forma particular de cada indivíduo de realizá-la – personalização de tarefas.

Todas as tarefas realizadas pelos profissionais clínicos são compostas de outras subtarefas, que agrupadas formam uma tarefa ou um conjunto de tarefas que representam o fluxo de trabalho (*workflow*) do profissional clínico (FERREIRA, 2009). As subtarefas associadas a uma tarefa podem ser processos realizados automaticamente pela arquitetura ClinicSpace (pró-atividade do sistema) ou tarefas realizadas pelos profissionais clínicos

(personalização de tarefas).

Conforme o médico vai interagindo com o sistema, um histórico é montado e as atividades que o profissional já realizou são associadas ao seu perfil no ClinicSpace. Desta forma, cada médico possui um conjunto de tarefas associadas ao seu perfil e que podem ser acessadas facilmente quando este profissional personaliza o seu fluxo de trabalho no ClinicSpace através do IETC (Figura 1).



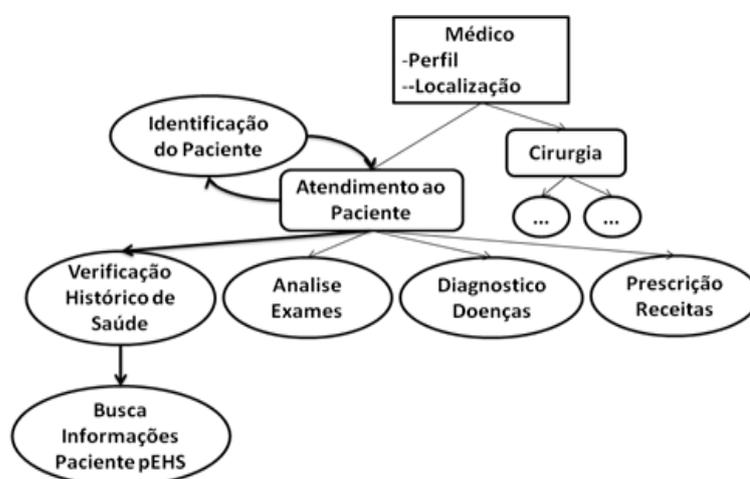
Apêndice A. Figura 1 - Demonstração da IETC para a edição do fluxo de trabalho
(MACHADO, 2010)

Como base para a modelagem das atividades clínicas na arquitetura ClinicSpace, Silva (2009) elencou um conjunto de 11 tarefas realizadas frequentemente em ambientes hospitalares. E elas foram determinadas a partir do estudo de Laerum and Faxvaag (2004), que realizaram uma pesquisa em diversas instituições hospitalares para obter uma lista das tarefas mais realizadas em ambientes clínicos e o tempo de duração destas tarefas. A Tabela 1 apresenta as tarefas mais realizadas nos ambientes hospitalares, definidas por Silva (2009).

Apêndice A. Tabela 1 - Conjunto de tarefas clínicas mínimas (SILVA, 2009)

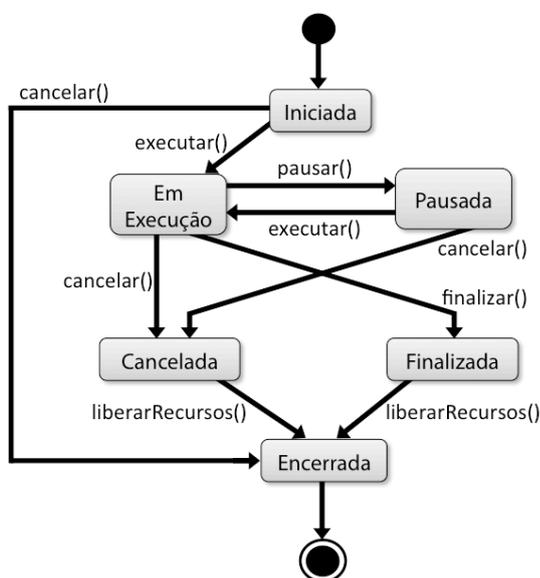
ID	Descrição
1	Revisar os problemas do paciente
2	Procurar informações específicas em registros do paciente
3	Obter os resultados de novos testes ou investigações
4	Adicionar notas diárias sobre as condições do paciente
5	Requisitar análises clínicas em laboratórios bioquímicos
6	Obter resultados de exames clínicos
7	Requisitar raios-X, ultra-som e tomografias computadorizadas
8	Obter resultados de raios-X, ultra-som e tomografias computadorizadas
9	Requisitar tratamentos
10	Escrever Prescrições
11	Registrar códigos para diagnósticos ou procedimentos executados

Na arquitetura, as tarefas são modeladas de uma forma geral, porém são dinâmicas, ou seja, uma tarefa pode conter subtarefas diferentes se realizada em diferentes momentos (SILVA, 2009). Na modelagem genérica de tarefas na arquitetura, tarefas podem conter: (i) subtarefas; ou (ii) uma tarefa inteira composta por subtarefas, e contida em outra tarefa de propósito mais abrangente. Assim, forma uma representação em grafo, onde um nó pode conter subtarefas embutidas, como ilustra a Figura 2.



Apêndice A. Figura 2 - Representação-exemplo do relacionamento entre as tarefas e subtarefas (MACHADO, 2010)

Para que a execução computacional das tarefas possa ser gerenciada pela arquitetura ClinicSpace, foi incorporado um ciclo de vida às tarefas (Figura 3), as quais são gerenciadas pelo Serviço Gerenciador de Tarefas (SGT).



Apêndice A. Figura 3 - Diagrama de possíveis estados de uma tarefa

Como pode ser observado na Figura 3, as tarefas possuem seis estados possíveis em relação a sua execução. Segundo Ferreira (2009), as tarefas após instanciadas, obrigatoriamente, devem estar em um desses seis estados:

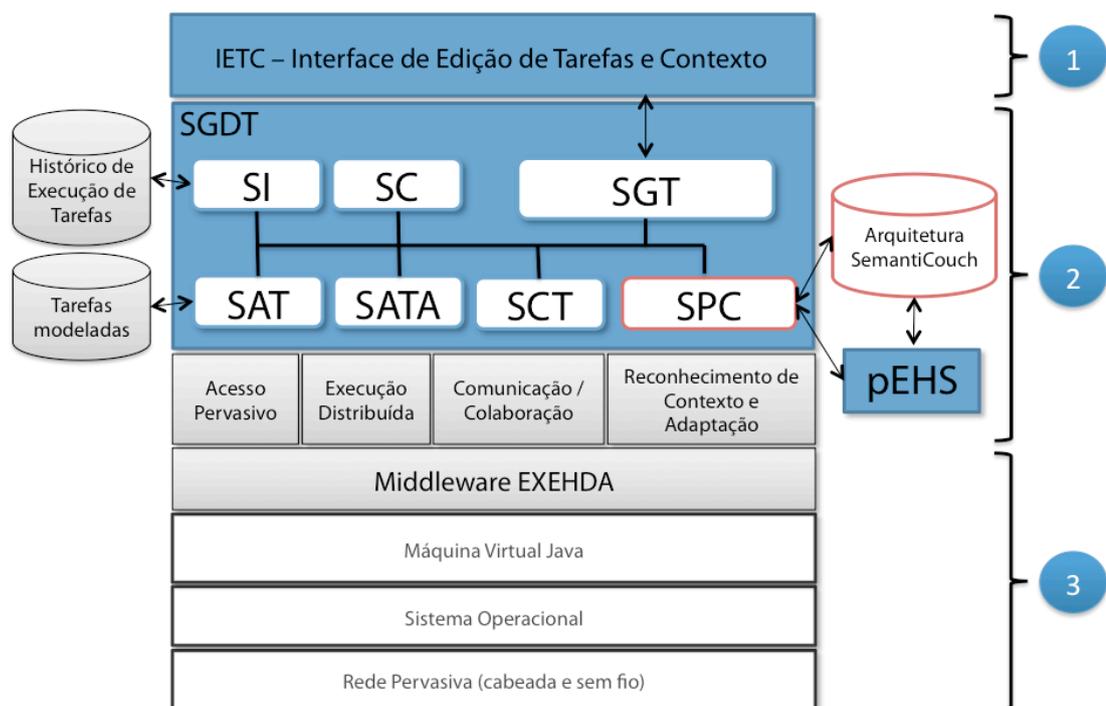
- **Iniciada (*Initialized*):** Neste estado a tarefa foi criada e inicializada recentemente e está pronta para ser executada pelo sistema;
- **Em Execução (*Running*):** Neste estado, a tarefa está ativa e sendo executada normalmente;
- **Pausada (*Paused*):** Neste estado, a tarefa foi interrompida e pode ser retomada a qualquer momento pelo sistema ou pelo usuário;
- **Finalizada (*Terminated*):** Neste estado a tarefa foi finalizada por ter encerrado a sua operação – terminou corretamente.
- **Cancelada (*Canceled*):** Neste estado a tarefa teve sua execução cancelada – não terminou corretamente.
- **Encerrada (*Closed*):** Neste estado, além de finalizada ou cancelada, a tarefa teve seus recursos (previamente alocados pelo sistema) liberados para a utilização por outras tarefas.

Baseado na definição de tarefas e sua importância para a definição de sistemas clínicos, a arquitetura ClinicSpace foi definida.

Arquitetura

Para prover o suporte computacional necessário para o auxílio na execução de tarefas, a arquitetura ClinicSpace é composta por camadas que integram um conjunto de serviços. Todas as camadas de serviços são gerenciadas pelo *middleware* EXEHDA (YAMIN, 2004), responsável pela integração dos componentes da arquitetura ClinicSpace.

Como apresentada na Figura 4, a arquitetura é separada por níveis, sendo eles: (i) nível superior, responsável pelas interfaces de interação do sistema com os usuários (médicos), composta pela IETC; (ii) nível intermediário, responsável pela modelagem de tarefas e interligação entre a camada superior (de interfaces) e a camada inferior; (iii) nível inferior, composto pelo *middleware* EXEHDA, que gerencia uma gama de recursos para a utilização pela arquitetura ClinicSpace (SILVA, 2009).



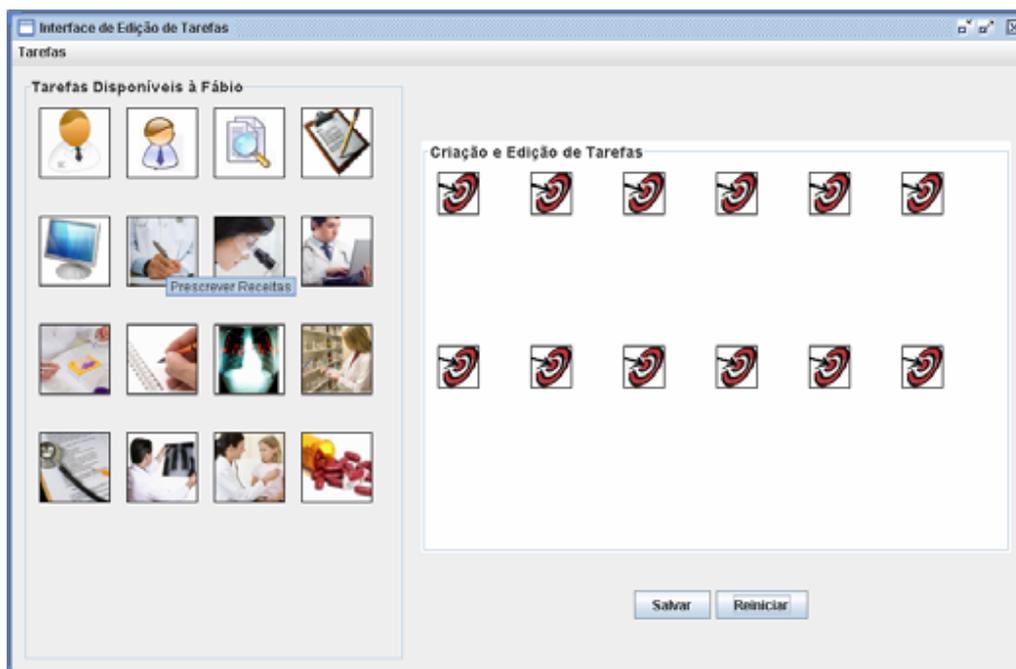
Apêndice A. Figura 4 - Arquitetura ClinicSpace

O *middleware* EXEHDA tem como principal característica a adaptação de recursos conforme o contexto de tarefas, a sua arquitetura é baseada em serviços, que em conjunto gerenciam um ambiente ubíquo (AUGUSTIN; YAMIN; GEYER, 2005).

Para realizar tarefas específicas, a arquitetura ClinicSpace é dividida em serviços, que se comunicam entre si constantemente. Esta comunicação entre os serviços varia de acordo com a tarefa que está sendo realizada no momento (SILVA, 2009).

A seguir, descreve-se brevemente as principais funções e características de cada um dos serviços implementados até o momento e os serviços SPC e SemantiCouch, propostos neste trabalho.

- **IETC (Interface de Edição de Tarefas e Contexto):** Para permitir o acesso e a personalização de informações pelos profissionais clínicos, foi desenvolvida a interface de edição de tarefas e contexto (SILVA, 2009) (MACHADO, 2010). Inicialmente, a IET (Interface de edição de tarefas), definida por Silva (2009) (Figura 5), permitia que o fluxo de trabalho fosse personalizado, mas não permitia a associação de elementos de contexto a estas tarefas. A definição de elementos de contexto na interface da arquitetura ClinicSpace foi realizada por Machado (2010), que adicionou a possibilidade de associação de um contexto ao fluxo de trabalho definido pelo usuário na própria interface.



Apêndice A. Figura 5 - Interface de Edição de Tarefas (SILVA, 2009)

Para realizar a personalização do seu fluxo de trabalho, o médico arrasta os ícones relacionados a cada atividade médica para a direita – que representa o seu fluxo de trabalho particular.

A cada mudança realizada na interface, a arquitetura salva a mudança no fluxo de tarefas do médico em uma ontologia que representa as informações relacionadas à atividade e ao contexto associado a tarefa (MACHADO, 2010).

- **Serviço de Gerenciamento de Tarefas (SGT):** Este serviço funciona como núcleo do serviço SGDT (Serviço de Gerenciamento Distribuído de Tarefas),. Ele é responsável pelo gerenciamento do ciclo de vida das tarefas na arquitetura ClinicSpace. Atua como uma camada mediadora entre a aplicação e os demais serviços, sendo construído com base no *design patterns Facade* (GAMMA, 1995), realizando a abstração de todos os serviços para a interface gráfica (Composta pela IETC) (SILVA, 2009).
- **Serviço de Acesso as Tarefas (SAT):** Este serviço é responsável por realizar o carregamento das tarefas do usuário, utilizando o *middleware* EXEHDA para carregar as tarefas do usuário com possibilidades móveis e execução distribuída (FERREIRA, 2009).

Para realizar esta tarefa, o serviço faz conversões e interpretações dos arquivos XML e objetos da linguagem Java para montar a estrutura de dados de um objeto *Tarefa* na memória

da aplicação (FERREIRA, 2009).

- **Serviço de Acesso a Tarefas Ativas (SATA):** Este serviço fornece a possibilidade de interrupção e retomada de execução das tarefas, bem como a migração de tarefas entre diferentes dispositivos utilizados pelo usuário. Este serviço mantém a informação do estado da tarefa (FERREIRA, 2009).
- **Serviço de Contexto para Tarefas (SCT):** Este serviço é responsável pelo relacionamento das tarefas referentes ao usuário com o contexto configurado no momento da modelagem. Além disso, o serviço 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).
- **Serviço de Inferência de Tarefas (SI):** Este serviço realiza a inferência das tarefas que um usuário deseja executar. Para realizar esta tarefa, ele monta uma árvore de decisão (SOUZA, 2010) baseada em informações do contexto providas pelo *middleware* EXEHDA e o histórico de execução das tarefas do usuário. O histórico de execução de tarefas do usuário é uma base de dados relacional onde são armazenadas todas as informações relacionadas as atividades já realizadas pelos usuários (SOUZA, 2010).
- **Serviço de Colaboração (SC):** Este serviço foi projetado para atender a necessidade de colaboração entre os profissionais em um ambiente clínico. Segundo Bardram & Christensen (2007), a colaboração é um aspecto fundamental no trabalho de médicos, enfermeiros, e outros profissionais da saúde. Características como colaboração entre uma atividade médica, como diagnóstico, ou a delegação de uma atividade não terminada por um médico devido à troca de turno de trabalho, estão sendo identificadas para que o Serviço de Colaboração possa ser utilizado em tempo de execução (KROTH, 2011).
- **Serviço de Persistência de Contexto e Seleção Contextualizada de Documentos (SPC) (Definido neste trabalho):** Este serviço realiza a intermediação entre os demais serviços da arquitetura ClinicSpace e a arquitetura de persistência SemantiCouch. Tanto em relação a persistência de informações de contexto e consulta destas informações pelos serviços do ClinicSpace quando necessário quanto em relação a consulta de documentos baseada em contexto realizada pelo *pEHS* e pelo serviço SCT – mais detalhes são apresentados na seção 4.3.
- **Arquitetura de Persistência *SemantiCouch* (Definida neste trabalho):** Esta

arquitetura é responsável pela distribuição de informações clínicas dos pacientes entre as diversas instituições hospitalares e é responsável pela persistência e consulta de dados de contexto utilizados e gerados pela arquitetura ClinicSpace. Mais detalhes são apresentados na seção 4.2.

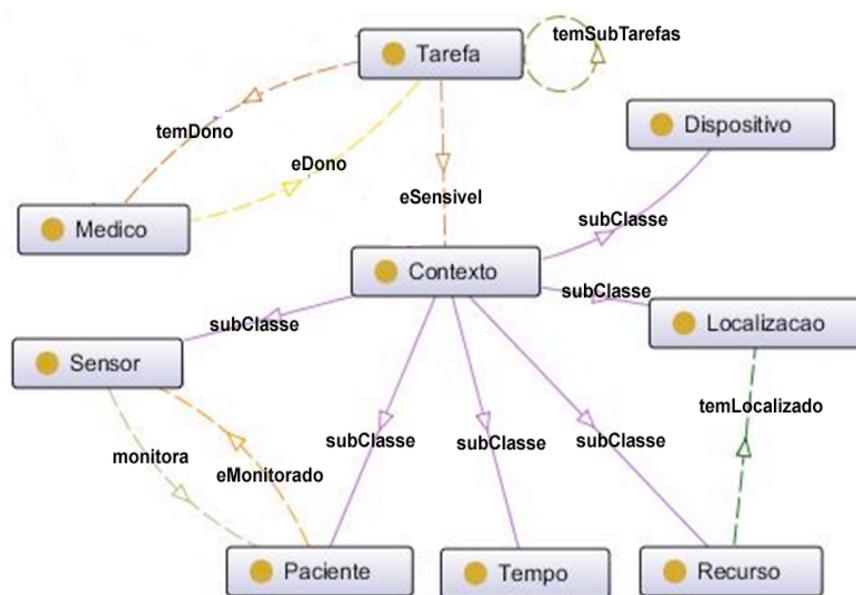
- **Sistema Eletrônico de Saúde Pervasivo (pEHS):** O pEHS é um sistema voltado ao gerenciamento de informações de saúde do paciente, bem como informações dos clínicos. É denominado pervasivo pois utiliza uma base de dados acessível em qualquer lugar, a todo tempo, com qualquer dispositivo – provida pela arquitetura *SemantiCouch*; sendo assim, o histórico de saúde do paciente é acessível de diferentes instituições de saúde, sem restrições de domínio de rede (VICENTINI, 2010).
- **Middleware de Gerenciamento para Ambientes Pervasivos – EXEHDA:** O *Middleware* EXEHDA, provê os serviços de mais baixo nível da arquitetura ClinicSpace, abstraindo do programador de subtarefas e aplicações pEHS a dinamicidade e heterogeneidade do ambiente pervasivo. Dentre eles, 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 (AUGUSTIN; YAMIN; GEYER, 2005).

Um dos fatores mais importantes para o funcionamento correto da arquitetura é como o contexto é representado e utilizado na arquitetura ClinicSpace. A próxima seção apresenta como o contexto é modelado e utilizado na arquitetura ClinicSpace até o momento.

Contexto na Arquitetura ClinicSpace

Para a modelagem de contexto, Machado (2010) propôs a utilização de ontologias, que são utilizadas pelos serviços da arquitetura ClinicSpace para a realização de serviços baseados em contexto.

A ontologia para a definição de contexto foi desenvolvida para representar elementos de contexto relacionados com as atividades clínicas realizadas por profissionais clínicos em ambientes hospitalares (MACHADO, 2010). A Figura 6 apresenta uma representação resumida da ontologia proposta para a arquitetura ClinicSpace.



Apêndice A. Figura 6 - Ontologia utilizada para a representação de contexto (MACHADO, 2010)