

**UNIVERSIDADE FEDERAL DE PELOTAS**  
Centro de Desenvolvimento Tecnológico  
Programa de Pós-Graduação em Computação



Dissertação

**EXEHDA-HM: Uma Abordagem Composicional para Processamento de  
Informações Contextuais em Modelos Híbridos de Contexto**

**Roger da Silva Machado**

Pelotas, 2016

**Roger da Silva Machado**

**EXEHDA-HM: Uma Abordagem Composicional para Processamento de  
Informações Contextuais em Modelos Híbridos de Contexto**

Dissertação apresentada ao Programa de  
Pós-Graduação em Computação da Universi-  
dade Federal de Pelotas, como requisito par-  
cial à obtenção do título de Mestre em Ciência  
da Computação

Orientadora: Prof<sup>a</sup>. Dr<sup>a</sup>. Ana Marilza Pernas  
Coorientador: Prof. Dr. Adenauer Corrêa Yamin

Pelotas, 2016

Universidade Federal de Pelotas / Sistema de Bibliotecas  
Catalogação na Publicação

M149e Machado, Roger da Silva

EXEHDA-HM : uma abordagem composicional para processamento de informações contextuais em modelos híbridos de contexto / Roger da Silva Machado ; Ana Marilza Pernas, orientadora ; Adenauer Corrêa Yamin, coorientador. — Pelotas, 2016.

92 f. : il.

Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2016.

1. Computação ubíqua. 2. Ciência de contexto. 3. Modelagem híbrida de contexto. 4. Processamento de contexto. I. Pernas, Ana Marilza, orient. II. Yamin, Adenauer Corrêa, coorient. III. Título.

CDD : 005



Universidade Federal de Pelotas  
Centro de Desenvolvimento Tecnológico  
Programa de Pós-Graduação em Computação

---

## ATESTADO

ATESTO para os devidos fins, que **ROGER DA SILVA MACHADO** foi aprovado pela Banca Examinadora da Defesa de Dissertação realizada em 2 de Março de 2016, intitulada: "EXEHDA-HM: Uma Abordagem Composicional para Processamento de Informações Contextuais em Modelos Híbridos de Contexto", sob a orientação do Prof. Dr. Ana Marilza Pernas. Realizadas as correções recomendadas pela Comissão Examinadora, o trabalho será encaminhado para homologação pelo Colegiado do Programa de Pós-Graduação em Computação. Igualmente, informamos que os demais requisitos necessários para a obtenção do grau de **Mestre em Ciência da Computação** pela Universidade Federal de Pelotas foram cumpridos.

Pelotas, 2 de Março de 2016

A handwritten signature in black ink, appearing to read 'Ricardo Matsumura de Araujo', written over a horizontal line.

Ricardo Matsumura de Araujo  
Coordenador do Programa de Pós-Graduação em Computação

## **AGRADECIMENTOS**

Primeiramente, quero agradecer a Deus, pois sei que sem ele nada poderia ter sido realizado.

Quero agradecer a minha família, meus pais que sempre me deram todo o apoio para me dedicar aos estudos e sempre deram todo o suporte financeiro de forma a nunca deixar faltar nada, a minha vó Geltrudes que me criou e sei que me tornei a pessoa que sou hoje graças a ela, aos meus irmãos pelo carinho e por todas as coisas que compartilhamos ao longo dos anos. Um agradecimento especial para minha namorada Fernanda, por todo o incentivo para me dedicar a este trabalho.

Agradeço aos meus orientadores Prof<sup>a</sup>. Dr<sup>a</sup>. Ana Marilza Pernas e Prof. Dr. Adenauer Correa Yamin, pela dedicação, paciência e grande incentivo na realização deste trabalho, pois tenho certeza que o trabalho desenvolvido não teria sido o mesmo sem as suas participações.

Agradeço a todos os colegas do mestrado, pois com certeza todos contribuíram para realização deste trabalho, pelas horas de estudos, conversas a respeito dos trabalhos e datas importantes a serem lembradas. Um agradecimento especial para o colega Ricardo Almeida, que participou durante todas as etapas de desenvolvimento do trabalho, principalmente o período de escritas de artigo e tentativas de publicações.

**Talvez não tenha conseguido fazer o melhor,  
mas lutei para que o melhor fosse feito.  
Não sou o que deveria ser, mas  
Graças a Deus, não sou o que era antes**  
— MARTIN LUTHER KING

## RESUMO

MACHADO, Roger da Silva. **EXEHDA-HM: Uma Abordagem Composicional para Processamento de Informações Contextuais em Modelos Híbridos de Contexto.** 2016. 91 f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2016.

Com a disseminação das redes de computadores e de dispositivos portáteis de elevado poder computacional e amplo espectro de uso, teve início a terceira era na cadeia evolutiva da computação moderna, a Computação Ubíqua. Uma das principais características da computação ubíqua é a ciência de contexto, a qual pode ser dividida em três etapas, sendo elas, aquisição, modelagem e processamento. A sinergia entre estas três etapas potencializa a utilização do contexto por parte das aplicações. Entre outros aspectos, o emprego de contexto proporciona uma melhor interação do usuário com a aplicação. Destaca-se que na etapa de modelagem de contexto, diversas abordagens podem ser utilizadas, sendo que ainda não foi encontrado um modelo que possa ser aplicado de maneira igualmente satisfatória em diferentes cenários de uso. Com isso têm surgido as abordagens baseadas na integração de modelos, modelagens híbridas, as quais combinam dois ou mais modelos com o intuito de aproveitar os pontos positivos de cada modelo. A utilização de modelos híbridos de contexto traz novos desafios para as aplicações cientes de contexto, pois se tornam necessários mecanismos que facilitem o acesso a estas informações, formas de manter a persistência das mesmas, como também estratégias para processar os diferentes contextos adquiridos. Pensando nos desafios encontrados pela utilização de modelos híbridos de contexto, o presente trabalho apresenta uma abordagem composicional para processamento de informações contextuais em modelos híbridos de contexto, intitulada EXEHDA-HM, onde é provida a utilização de um repositório com base em três modelos de armazenamento: relacional, não relacional, e de triplas. Além disso, a abordagem proposta possui como diferencial uma estratégia de processamento composicional baseada em regras, a qual possibilita à aplicação combinar as informações contextuais armazenadas nos diferentes modelos em uma mesma regra, contribuindo assim para uma melhor identificação das situações de interesse.

**Palavras-chave:** Computação Ubíqua, Ciência de Contexto, Modelagem Híbrida de Contexto, Processamento de contexto.

## ABSTRACT

MACHADO, Roger da Silva. **EXEHDA-HM: A Compositional Approach to Processing Contextual Information in Hybrid Context Models.** 2016. 91 f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2016.

With the dissemination of computer networks, and the increased usage of portable devices of high computing power and broad spectrum of use, began the third era in the evolutionary chain of modern computing, ubiquitous computing. A key feature of ubiquitous computing is the context awareness, which can be divided into three stages, being they, acquisition, modeling and processing. The synergy between these three stages boosts the use of context by the applications, among other aspects, the employment of context provides a better user interaction with the application. It is noteworthy that in the context modeling stage, various approaches can be used, and has not yet been found a model that can be applied of manner equally satisfactorily in different application scenarios. Thus, have arisen the approaches based on integration models, hybrid modeling, which combine two or more modeling approaches in order to take advantage of the strengths of each model. The use of hybrid context models brings new challenges for the applications context-aware as they become necessary mechanisms to facilitate access to this information, ways to maintain the persistence of the same, as well as strategies to process the acquired contexts. Thinking about the challenges faced by the use of hybrid context models, this work presents a compositional approach to processing contextual information in hybrid context models, named the EXEHDA-HM, where is provided the use of one repository based on three models: relational; non-relational; and triplestore. Moreover, the proposed approach has as highlights the compositional processing strategy based on rules, which enables an application to combine the contextual information stored in the different models in the same rule, thus contributing to a better identification of the situations of interest.

**Keywords:** Ubiquitous Computing, Context Awareness, Hybrid Context Modeling, Context Process.

## LISTA DE FIGURAS

Figura 1	Processamento de dados de uma aplicação ciente de contexto . . .	16
Figura 2	Visão geral das etapas da ciência de contexto . . . . .	20
Figura 3	Utilização das técnicas de processamento de contexto . . . . .	25
Figura 4	Exemplo de modelagem de perfis em CSCP . . . . .	28
Figura 5	Exemplo de representação do modelo ORM . . . . .	29
Figura 6	Exemplo de modelo orientado a objetos . . . . .	30
Figura 7	Exemplo de um modelo baseado em lógica para localização de usuário . . . . .	31
Figura 8	Exemplo de uma ontologia referente a informações de usuário . . .	32
Figura 9	Processo para manter a consistência no <i>middleware</i> MoCA . . . . .	38
Figura 10	Modelagem híbrida para o <i>framework</i> iTransIT . . . . .	39
Figura 11	Arquitetura da plataforma CoCA . . . . .	40
Figura 12	Arquitetura do repositório para sistemas SIEM . . . . .	43
Figura 13	Diagrama de Bloco Funcional do sistema ciente de contexto . . . .	44
Figura 14	Arquitetura de software do middleware EXEHDA . . . . .	48
Figura 15	Ambiente ubíquo provido pelo EXEHDA . . . . .	48
Figura 16	Arquitetura da EXEHDA-HM . . . . .	57
Figura 17	Arquitetura do servidor de contexto . . . . .	58
Figura 18	Diagrama de classes do mecanismo de gerenciamento . . . . .	60
Figura 19	Diagrama entidade relacionamento das regras composicionais . . .	64
Figura 20	Comunicação entre EXEHDA-HM e aplicação ciente de contexto . .	65
Figura 21	Fluxo de execução da regra composicional . . . . .	66
Figura 22	Estrutura resumida da ontologia que modela o tráfego da rede . . .	70
Figura 23	Funcionamento da regra para detecção de ataques a rede . . . . .	71
Figura 24	Consulta SQL referente à busca pela categoria de CPU utilizado .	72
Figura 25	Consulta SPARQL referente à busca pelo ataque DoS a um dispositivo	72
Figura 26	Hierarquia adicionada a ontologia da ISO 27002 . . . . .	74
Figura 27	Exemplo de log para identificação da situação de política de senha	74
Figura 28	Funcionamento da regra para o reforço de políticas de senha . . . .	75
Figura 29	Consulta no modelo não relacional pela situação de interesse . . .	76
Figura 30	Consulta SPARQL referente à busca pela diretriz complexity requirements . . . . .	76

## LISTA DE TABELAS

Tabela 1	Exemplo de modelagem chave-valor para informações de um usuário	27
Tabela 2	Comparação entre abordagens de modelagem de contexto . . . . .	33
Tabela 3	Comparação entre os trabalhos de modelagem híbrida de contexto	46
Tabela 4	Comparação entre as ferramentas para armazenamento de triplas .	52
Tabela 5	Comparação dos trabalhos de modelagem híbrida de contexto com a EXEHDA-HM . . . . .	68

## LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
APF	Administração Pública Federal
API	<i>Application Programming Interface</i>
CADDM	<i>Context Acquisition and Distribution Data Model</i>
CML	<i>Context Modelling Language</i>
CMP	<i>Context UML Profile</i>
CoCA	<i>Collaborative Context-Aware Service Platform</i>
ConDoR	<i>Context Distribution and Reasoning</i>
CRDM	<i>Context Reasoning Data Model</i>
CSCP	<i>Comprehensive Structured Context Profiles</i>
CT	<i>Context Tool</i>
DoS	<i>Denial of Service</i>
DSIC	Departamento de Segurança da Informação e Comunicações
EXEHDA	<i>Execution Environment for Highly Distributed Applications</i>
HCoM	<i>Hybrid Context Management</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IEC	<i>International Electrotechnical Commission</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
MoCA	<i>Mobile Collaboration Architecture</i>
NBR	Norma Brasileira
ORM	<i>Object Role Modeling</i>
OWL-DL	<i>Web Ontology Language-Description Logics</i>
PCM	<i>Primary-Context Model</i>
PCont	<i>Primary-Context Ontology</i>
POSIC	Política de Segurança da Informação e Comunicações

RAID	<i>Reasoning, Aggregation, Interpretation, Decision and Action</i>
RCDB	<i>Relational Context Database</i>
RDF	<i>Resource Description Framework</i>
RDFS	<i>Resource Description Framework Schema</i>
RHIC	Repositório Híbrido de Informações Contextuais
SIEM	<i>Security Information and Events Management</i>
SOA	<i>Service-Oriented Architecture</i>
SPARQL	<i>SPARQL Protocol and RDF Query Language</i>
SQL	<i>Structured Query Language</i>
UML	<i>Unified Modeling Language</i>
URI	<i>Uniform Resource Identifier</i>
Wi-Fi	<i>Wireless Fidelity</i>
XML	<i>eXtensible Markup Language</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	Tema	15
1.2	Motivação e Objetivos	16
1.3	Estrutura do Texto	17
<b>2</b>	<b>CIÊNCIA DE CONTEXTO</b>	<b>19</b>
2.1	Aquisição de Contexto	21
2.2	Modelagem de Contexto	22
2.3	Processamento de Contexto	23
2.4	Considerações Finais	25
<b>3</b>	<b>ABORDAGENS PARA MODELAGEM DE CONTEXTO</b>	<b>26</b>
3.1	Modelos Chave-Valor	26
3.2	Modelos Baseados em Linguagens de Marcação	26
3.3	Modelos Gráficos	27
3.4	Modelos Orientados a Objetos	29
3.5	Modelos Baseados em Lógica	30
3.6	Modelos Baseados em Ontologias	31
3.7	Análise das Abordagens para Modelagem de Contexto	32
3.8	Considerações Finais	34
<b>4</b>	<b>TRABALHOS RELACIONADOS A MODELAGEM HÍBRIDA DE CONTEXTO</b>	<b>35</b>
4.1	Modelo Gráfico e de Ontologias	35
4.2	Modelo Orientado a Objetos e de Ontologias	36
4.2.1	ConDoR	36
4.2.2	Modelo Híbrido do Middleware MoCA	37
4.2.3	Modelo Híbrido do Framework iTransIT	38
4.3	Modelo Relacional e de Ontologias	39
4.4	Modelo Relacional, Linguagem de Marcação e de Ontologias	42
4.5	Modelo Objeto Relacional e de Ontologias	43
4.6	Considerações Finais	45
<b>5</b>	<b>ESCOPO DO TRABALHO</b>	<b>47</b>
5.1	Middleware EXEHDA	47
5.2	Armazenamento de triplas: conceitos e ferramentas	49
5.3	Tecnologias Utilizadas	51
5.3.1	PostgreSQL	52
5.3.2	MongoDB	53

5.3.3	Virtuoso . . . . .	53
5.3.4	SPARQL . . . . .	53
5.3.5	Jena . . . . .	54
5.4	<b>Considerações Finais . . . . .</b>	<b>55</b>
6	<b>EXEHDA-HM: ARQUITETURA E PRINCIPAIS FUNCIONALIDADES . . . .</b>	<b>56</b>
6.1	<b>EXEHDA-HM: Camada de Serviço . . . . .</b>	<b>59</b>
6.2	<b>EXEHDA-HM: Camada de Armazenamento . . . . .</b>	<b>62</b>
6.3	<b>EXEHDA-HM: Camada de Processamento . . . . .</b>	<b>64</b>
6.4	<b>Considerações Finais . . . . .</b>	<b>67</b>
7	<b>CENÁRIOS DE USO . . . . .</b>	<b>69</b>
7.1	<b>Cenário 1 - Situação de Ataque a Rede . . . . .</b>	<b>69</b>
7.2	<b>Cenário 2 - Reforço de Políticas de Senhas . . . . .</b>	<b>73</b>
7.3	<b>Considerações Finais . . . . .</b>	<b>76</b>
8	<b>CONCLUSÃO . . . . .</b>	<b>77</b>
8.1	<b>Trabalhos Futuros . . . . .</b>	<b>79</b>
8.2	<b>Publicações . . . . .</b>	<b>80</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>82</b>

# 1 INTRODUÇÃO

Com o passar dos anos, os computadores tornaram-se cada vez mais presentes nas tarefas cotidianas e mais próximos dos usuários. Por volta da década de 40 teve início à primeira era da computação com o surgimento dos *mainframes*, computadores que geralmente ocupavam um grande espaço e possuíam acesso restrito. Nas décadas de 70 e 80 surgia a segunda era da computação com o advento dos primeiros computadores pessoais (PCs), possibilitando uma maior popularização dos computadores e aproximação com os usuários, além da realização de tarefas gerais como processamento de texto, navegação na Internet, jogos e programação.

Entretanto, foi com a popularização da Internet e dos ambientes distribuídos que os computadores passaram a se interconectar, compartilhando recursos e informações, ampliando os serviços oferecidos e a complexidade das aplicações desenvolvidas, levando a computação para a terceira era da computação moderna, a era da Computação Ubíqua. Esta é a fase dos dias atuais, onde há uma variedade de dispositivos eletrônicos com diferentes perfis computacionais, que fornecem conjuntamente serviços para o usuário (RAZZAQUE et al., 2016) (KRUMM, 2009).

O paradigma da computação ubíqua tem como premissa prover computação de forma transparente, estando o modelo computacional integrado as demandas do usuário. Este paradigma foi introduzido no visionário artigo (WEISER, 1991), e contempla diversas frentes de pesquisa em aberto até os dias de hoje. Ao se construir e executar aplicações ubíquas cientes de contexto há uma série de funcionalidades que devem ser providas, envolvendo desde a aquisição de informações contextuais, a partir do conjunto de fontes heterogêneas e distribuídas, até a representação dessas informações, seu processamento, armazenamento e a realização de inferências para seu uso em tomadas de decisões (BELLAVISTA et al., 2012).

Conhecer o contexto em que ocorre uma interação, embora constitua uma tarefa usual para os seres humanos, mostra-se uma tarefa complexa em sistemas computacionais. A construção de sistemas cientes de contexto exige inicialmente, a definição do que considerar como contexto, onde este se aplica e que informações são necessárias para descrevê-lo. É preciso viabilizar formas de realizar a aquisição do

contexto de forma mais automática possível. Após a coleta de dados brutos a partir do sensoriamento, tornam-se necessários mecanismos de processamento de contexto que tratem as informações coletadas, produzindo informações contextualizadas (BAUER; NEWMAN; KIENTZ, 2014).

O processamento das informações contextuais coletadas exige que as mesmas sejam organizadas segundo um modelo. Há diversas estratégias para modelagem de contexto presentes na literatura, cada uma considerando cenários específicos de uso (PATEL et al., 2009) (KOTENKO et al., 2013). Assim, a compreensão da natureza da aplicação é essencial para escolher ou projetar o modelo mais apropriado para realizar a etapa de modelagem de contexto. Por outro lado, a existência de modelos bem definidos é oportuna para a verificação da consistência dos dados contextuais coletados (BIBRI, 2015).

## 1.1 Tema

O tema central desta dissertação é o processamento de informações contextuais em modelos híbridos de contexto. Atualmente as aplicações cientes de contexto tendem a tratar com dados de diferentes naturezas, onde os mesmos podem ser modelados utilizando diversas abordagens. Nesta perspectiva se observa uma tendência na utilização de repositórios com diferentes modelos de armazenamento, pois muitas vezes a utilização de somente um modelo para armazenamento das informações contextuais acaba não sendo o ideal, por questões ligadas a desempenho, utilização de disco entre outras (CARVALHO, 2014).

A figura 1 ilustra um cenário enfrentado por aplicações de segurança da informação, onde para identificação de uma situação de interesse podem estar presentes Dados Semiestruturados, os quais possuem uma estrutura irregular, onde não existe um esquema padrão para os dados, este tipo de dado pode ser obtido de logs de aplicações. Dados Sintáticos, os quais representam o contexto sem associar semântica ao mesmo, como por exemplo, dados referentes ao status do dispositivo computacional (quantidade de processador e memória utilizada, entre outras). Além destes, Dados Semânticos, os quais agregam significado à representação do contexto, e podem ser providos por normas de segurança que devem ser empregadas no ambiente computacional.

Pensando na necessidade de processamento de dados provenientes de diferentes modelos, a proposta desenvolvida tem como premissa o desenvolvimento de uma estratégia de processamento das informações contextuais em ambientes que utilizam modelos híbridos de contexto, possuindo como um diferencial a capacidade de combinação das informações presentes nos diferentes modelos de contexto utilizados.

Na proposta desenvolvida as aplicações podem utilizar os modelos tanto de forma

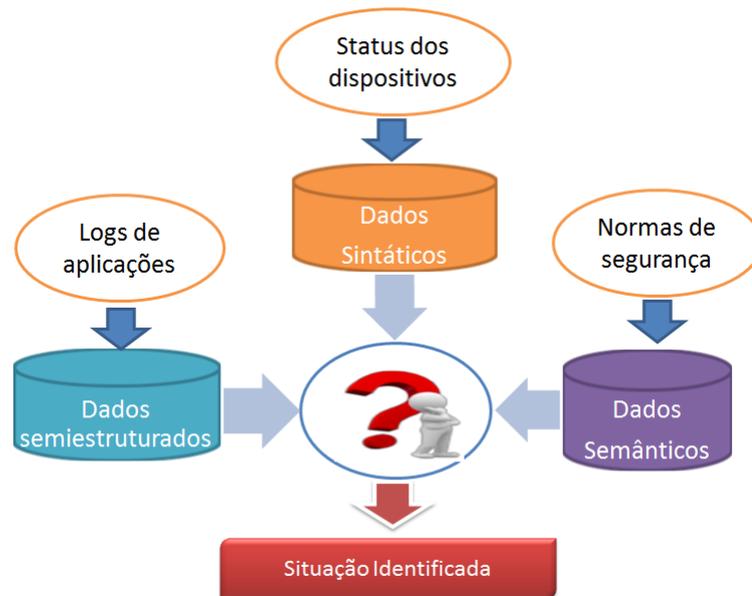


Figura 1: Processamento de dados de uma aplicação ciente de contexto

individual, quanto combinada, de tal modo que esta flexibilidade contribua para uma melhor identificação das situações de interesse.

## 1.2 Motivação e Objetivos

Aplicações cientes de contexto podem tornar a vida do usuário mais confortável e, por consequência, melhorar a sua qualidade de vida, podendo fazer com que suas necessidades sejam atendidas com mais facilidade pelo sistema computacional (ASENSIO, 2011).

Uma das principais etapas do desenvolvimento de aplicações cientes de contexto diz respeito à forma de estruturação e armazenamento dos dados de contexto. A estratégia adotada irá refletir diretamente nas possibilidades a serem empregadas na etapa seguinte, de processamento de contexto (BETTINI et al., 2010).

Há diversas abordagens de modelagem de contexto presentes na literatura, as quais diferem por aspectos como expressividade, formalismo, aplicabilidade a diferentes domínios, dentre outros aspectos. Consta-se que não há uma proposta de modelagem de contexto perfeita, e que a compreensão do problema em si é essencial para escolher ou projetar o modelo apropriado de modelagem de contexto (BOLCHINI et al., 2007).

Em função de não existir uma proposta de modelagem de contexto que atenda às aplicações de diferentes naturezas, as abordagens híbridas têm sido apontadas como possível solução a este problema. Os modelos híbridos vêm sendo considerados promissores, pois ao combinar duas ou mais técnicas de modelagem, dotadas de diferentes níveis de interpretação, têm o seu espectro de uso potencialmente ampliado

(PERERA et al., 2014).

A utilização de modelagens híbridas traz um novo desafio que é como realizar o gerenciamento do contexto, já que são combinados diferentes modelos e/ou várias formas de armazenamento das informações contextuais. Com isso, são necessários mecanismos que garantam o acesso às informações contextuais já capturadas, e que sejam disponibilizadas às aplicações interessadas, de forma atualizada, porém, sempre preservando a consistência destas informações (BELLAVISTA et al., 2012).

Esta dissertação tem por objetivo geral a concepção de uma abordagem denominada EXEHDA-HM (*Execution Environment for Highly Distributed Applications-Hybrid Modelling*) que oferece suporte a utilização de modelagem híbrida de contexto, possibilitando o armazenamento em diferentes modelos e um mecanismo de processamento composicional de contexto. A abordagem proposta é integrada ao *middleware* EXEHDA (YAMIN, 2004) (LOPES et al., 2014), contribuindo com o seu Subsistema de Adaptação e Reconhecimento de Contexto.

Como objetivos específicos do trabalho, são destacadas as seguintes frentes:

- revisar as premissas de concepção do *middleware* EXEHDA, em particular do Subsistema de Adaptação e Reconhecimento de Contexto;
- conceber uma abordagem com suporte a modelagem híbrida de contexto, onde será provida a integração entre diferentes modelos de contexto, de forma a se beneficiar das vantagens de cada modelo;
- prover um mecanismo para persistência das informações contextuais sintáticas e semânticas, permitindo o acesso e manipulação destas informações;
- conceber uma estratégia composicional para processamento de contexto, a qual possua como diferencial a possibilidade de combinar dados semânticos e sintáticos;
- desenvolver um estudo de caso, que demonstre a utilização da abordagem EXEHDA-HM;
- divulgar ante a comunidade científica os resultados atingidos pela pesquisa através de publicações em eventos e/ou jornais especializados da área.

### 1.3 Estrutura do Texto

Este trabalho está dividido em oito capítulos.

O segundo capítulo apresenta os conceitos de Ciência de Contexto, visando apresentar suas principais características e etapas presentes no gerenciamento das

informações contextuais. Na sequência do capítulo são destacadas as etapas de aquisição, modelagem e processamento de contexto.

No terceiro capítulo é realizada uma sumarização das abordagens de modelagem de contexto presentes na literatura, onde são apresentadas suas principais características, bem como as vantagens e desvantagens em sua utilização.

O quarto capítulo apresenta alguns trabalhos que utilizam abordagens híbridas de modelagem de contexto, realizando uma descrição destes, onde são destacadas as arquiteturas de cada trabalho, as áreas de aplicação e os modelos de contexto utilizados.

O quinto capítulo explora o escopo do trabalho desenvolvido, sendo apresentado o *middleware* EXEHDA, um levantamento sobre o armazenamento em triplas, sendo apresentadas algumas das principais ferramentas utilizadas e as principais características das ferramentas escolhidas para implementação da abordagem proposta.

No sexto capítulo é apresentada a abordagem EXEHDA-HM, mostrando o repositório de informações contextuais proposto, caracterizando o mecanismo para realizar o gerenciamento das informações contextuais, com o intuito de manter a consistência dos dados entre os modelos de contexto e as diferentes formas de armazenamento utilizadas para manter a persistência das informações contextuais. Ainda, é apresentada a estratégia composicional proposta para o processamento das informações contextuais, a qual possui como diferencial a possibilidade de combinar as informações presentes nos diferentes modelos de armazenamento.

O sétimo capítulo apresenta o estudo de caso desenvolvido para demonstrar a utilização da abordagem EXEHDA-HM, destacando os principais benefícios em sua utilização.

O trabalho finaliza no oitavo capítulo, onde são apresentadas as principais conclusões, os trabalhos futuros, bem como as publicações realizadas.

## 2 CIÊNCIA DE CONTEXTO

Contexto é qualquer informação que pode ser usada para caracterizar a situação de uma entidade (pessoa, local ou objeto) considerada relevante para a interação entre o usuário e a aplicação, incluindo o próprio usuário e a aplicação (DEY, 2001) (PERNAS, 2012).

Também pode ser considerada como contexto uma descrição complexa de conhecimento compartilhado sobre circunstâncias físicas, sociais, históricas, entre outras, onde ações ou eventos ocorrem. É o que restringe a interpretação de uma ação ou evento, sem, no entanto, ser parte dessa ação/evento. Portanto o contexto é uma coleção de condições relevantes e influências que tornam uma situação única e compreensível (BRÉZILLON, 1999).

A ciência de contexto é a capacidade de um sistema em usar o contexto para prover serviços e/ou informações relevantes para o usuário (PERNAS, 2012). Os sistemas cientes de contexto devem ser flexíveis, adaptativos, e capazes de atuar automaticamente para ajudar o usuário na realização de suas atividades (MACHADO, 2010).

As áreas da Computação Ubíqua e Inteligência Artificial foram as pioneiras em estudar e utilizar o conceito de contexto, demonstrando o potencial da aplicação desse conceito nos sistemas computacionais. Pesquisas recentes vêm utilizando o conceito de contexto para beneficiar sistemas ligados a outras áreas tais como:

- sistemas colaborativos: utilização de contexto para melhorar a interação e produtividade do grupo, aplicado por exemplo, na área médica;
- hipermídia adaptativa: possibilidade de personalização e adaptação do conteúdo de sites Web a partir de contextos;
- integração de dados: facilitando a resolução de conflitos semânticos com a utilização de contextos;
- interação humano-computador: o contexto é utilizado para adaptar as interfaces dos sistemas tornando mais intuitiva a sua interação com os usuários.

Algumas motivações para aplicação de ciência de contexto nos sistemas computacionais são:

- auxiliar na compreensão da realidade;
- contribuir na adaptação de sistemas;
- auxiliar no processo de transformação dos dados em informação;
- apoiar a compreensão de eventos;
- ajudar a identificar situações de interesse.

De forma simplificada, a construção de um sistema ciente de contexto se dá por meio de três etapas, sendo elas: aquisição, modelagem e processamento. A figura 2 apresenta uma visão geral das etapas relacionadas à obtenção da ciência do contexto, onde, na parte inferior da figura, estão os sensores, utilizados pela etapa de aquisição para capturar os dados contextuais de interesse. Após, estes dados são repassados para a camada de armazenamento, a qual é responsável por armazenar os dados coletados. Na sequência, os dados contextuais são repassados para a etapa processamento, a qual irá processar os dados com o intuito de realizar alguma inferência nos mesmos, de forma a identificar determinadas situações que podem gerar uma ação a ser realizada. A figura 2 ainda é apresentada a etapa de modelagem, a qual é considerada transversal as demais já que ela está diretamente ligada às operações das outras etapas da ciência de contexto. Destaca-se que o termo modelagem contexto também é referido como representação de contexto (PERERA et al., 2014), sendo tratados, por este trabalho, como sinônimos. No topo da figura 2 se dá a abstração das aplicações que utilizam os dados contextuais. Estas três etapas são apresentadas a seguir com maiores detalhes.

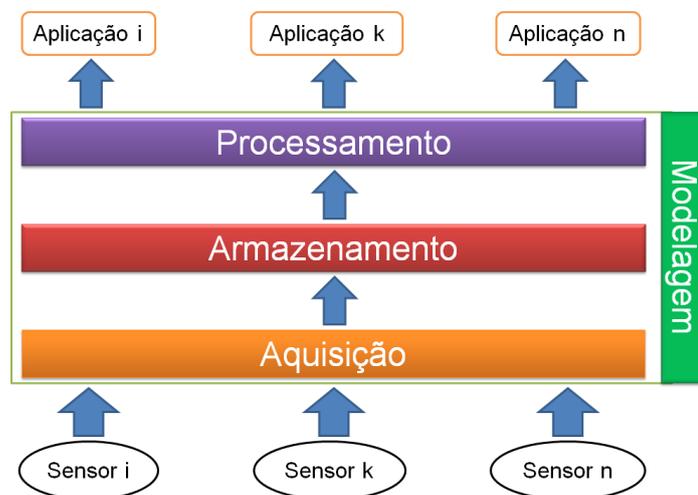


Figura 2: Visão geral das etapas da ciência de contexto

## 2.1 Aquisição de Contexto

A aquisição do contexto refere-se ao processo de monitorar, capturar e/ou obter informações contextuais. A informação contextual pode ser adquirida a partir de diversas fontes, tais como (VIEIRA et al., 2006):

- sensores físicos: são o tipo mais comum de sensores, tangíveis e geram dados por si só. Os dados obtidos a partir destes sensores são chamados de contexto de baixo nível. Possuem menor significância e apresentam vulnerabilidade a pequenas mudanças. São responsáveis pela captura de informações sobre o ambiente físico, tais como, luz, temperatura, umidade, gás. Alguns pontos positivos são: a detecção de erros e a possibilidade de identificar falta de valores de forma relativamente fácil; ser eficiente, devido a ter acesso às configurações do sensor de baixo nível. Como pontos negativos, destacam-se: implantação e manutenção de hardware, que pode ser custosa; tratar com sensores e programação de baixo nível.
- sensores virtuais: estes sensores recuperam dados de outras fontes e publicam como dados de sensores. Podem ser utilizados para coletar informações que não podem ser medidas fisicamente, tais como detalhes de calendário, e-mail, bate-papo, mapas, dados de redes sociais, preferências do usuário. Alguns pontos positivos são: fornecimento de dados significativos; informações de contexto de alto nível; dados fornecidos não precisam de processamento; não é necessário lidar com tarefas de nível de hardware. Como pontos negativos destacam-se: dificuldade de encontrar erros nos dados; preenchimento de valores ausentes não é uma tarefa fácil, já que em sua maioria são dados não numéricos e imprevisíveis.
- sensores lógicos: combinam sensores físicos e sensores virtuais, a fim de produzir informações mais significativas. Estes sensores são utilizados para coletar informações que não podem ser coletados diretamente através de um único sensor físico e para realiza a fusão de dados de vários sensores, como por exemplo, informações sobre o tempo, o reconhecimento de atividade, reconhecimento de localização. Alguns pontos positivos são: fornecimento de dados altamente significativos; provimento de informações de contexto de alto nível; normalmente informações mais precisas; não é necessário lidar com tarefas de nível de hardware. Como pontos negativos destacam-se: dificuldade de encontrar erros em dados; não se tem controle sobre processo de produção de dados.

As informações contextuais podem ser classificadas das seguintes formas (HENRICKSEN; INDULSKA, 2005):

- informações percebidas: são provenientes de sensores físicos ou lógicos, possuem uma baixa persistência. As informações podem ser imprecisas, desconhecidas ou caducas, e suas fontes de imperfeição são erros na percepção, falhas do sensor ou desconexões da rede;
- informações provenientes de perfil: são fornecidas pelo próprio usuário, possuem uma persistência moderada. As informações tendem a caducar, e suas fontes de imperfeição são a omissão do usuário em atualizar mudanças ocorridas;
- informações derivadas: são aquelas obtidas por meio de mecanismos de derivação, possuem uma persistência variável. As informações estão sujeitas a erros e imperfeições, e suas fontes de imperfeição são entradas imprecisas, mecanismo de derivação imaturo ou simplificado.

É importante que a tarefa de aquisição de contexto permaneça em constante execução, e é desejável que seja implementada de forma independente das aplicações que a utilizem, possibilitando assim que diversas aplicações possam fazer uso das mesmas informações contextuais.

## 2.2 Modelagem de Contexto

Um sistema ciente de contexto requer que informações contextuais sejam trocadas e utilizadas por diferentes entidades, como agentes humanos e de software, dispositivos e serviços. A quantidade de informação capturada e acessada num modelo de representação do contexto é significativa, com isso, os estudos de técnicas para representação de informações contextuais vêm se tornando cada vez mais importantes.

O processo de modelagem de contexto consiste na concepção de um modelo de entidades do mundo real, suas propriedades, estado de seu ambiente e situações que podem ser usados como referência para a aquisição, interpretação e raciocínio de informações contextuais (KNAPPMEYER et al., 2013). Um bom formalismo para modelagem de contexto reduz a complexidade das aplicações cientes de contexto, facilita o acesso às informações realizando buscas de forma eficiente, melhora a capacidade de manutenção e de evolução da aplicação (STRANG; LINNHOFF-POPIEN, 2004). Alguns requisitos que devem ser levados em consideração durante a modelagem das informações de contexto são: heterogeneidade, mobilidade, dependências e relações entre os dados.

Existem diversos modelos para representação do contexto, tais como, chave-valor, linguagem de marcação, gráfico, orientado a objetos, lógico, ontológico, entre outros.

Devido à relevância deste tópico para o presente trabalho, no capítulo 3 são apresentadas as principais estratégias presentes na literatura para realizar a etapa de modelagem de contexto, onde são apresentadas as principais características de cada modelo.

## 2.3 Processamento de Contexto

Processamento de Contexto pode ser definido como um mecanismo de raciocínio para inferir novos conhecimentos e melhorar a compreensão dos contextos adquiridos (PERERA et al., 2014). Também pode ser apresentado como um processo de realizar deduções de contexto de alto nível a um conjunto de contextos de baixo nível.

Um número significativo de mecanismos provenientes dos campos da inteligência artificial e com base em sistemas de conhecimento pode ser adotado para realizar o processo de raciocínio de contexto. A seguir são apresentadas algumas das técnicas utilizadas para realizar o processamento sobre a informação contextual.

- Baseadas em regras: este é um dos métodos mais simples para realizar raciocínio sobre contexto. Permite a geração de informações de contexto de alto nível utilizando contextos de baixo nível, é uma técnica simples para se definir e estender e não necessita de uma utilização intensiva de recursos computacionais. Como mostrado na figura 3, é a técnica mais utilizada em trabalhos científicos de contexto. O raciocínio baseado em regras possui alguns pontos negativos, tais como: quando se utiliza uma grande base de regras, facilmente se torna confuso e intratável; as regras devem ser definidas manualmente, o que é propenso a erros devido ao trabalho manual; não possui mecanismo para realizar a validação e verificação de qualidade. O raciocínio baseado em regras só pode ser aplicado em sistemas de suporte de contexto com propagação baseado em eventos, e não suporta imprecisão, somente sendo aplicada para respostas do tipo booleana (KNAPPMAYER et al., 2013).
- Baseadas em aprendizagem supervisionada: nesta categoria estão as técnicas que utilizam um conjunto de treinamento, onde, neste conjunto, os dados se encontram categorizados, ou seja, está presente o resultado esperado para cada caso que será utilizado para treinamento e, em seguida, é possível classificar novos eventos com base no conjunto que foi utilizado durante o treinamento. Dentre as técnicas desta categoria destacam-se: as Árvores de Decisão, as quais consistem de uma técnica de aprendizagem supervisionada onde é construída uma árvore a partir de um conjunto de dados que podem ser utilizados para classificar os novos eventos capturados, e as Máquinas de Vetores de Suporte, as quais são utilizadas para reconhecimento de padrões em computação ciente de

contexto. Os pontos positivos destas técnicas são: costumam alcançar um alto grau de precisão; disponibilidade de modelos alternativos; possuem base matemática e estatística. Dentre as dificuldades encontradas na utilização destas técnicas, pode-se citar: a exigência de quantidade significativa de dados para treinamento; pode ser necessária uma maior utilização de recursos computacionais, como processamento e armazenamento; seleção de dados que serão utilizados, de forma a não tornar o processamento muito custoso (PERERA et al., 2014).

- Baseada em aprendizagem não supervisionada: estas técnicas utilizam um conjunto de treinamento para aprender, mas estes dados não estão categorizados, ou seja, eles não possuem o resultado esperado. Nesta categoria destacam-se as técnicas de agrupamento, como a técnica k Vizinhos Mais Próximo. As técnicas utilizadas nesta categoria costumam ser utilizadas em redes de sensores, para realizar tarefas como de roteamento, e também nas operações de posicionamento e localização. Possuem destaque por conseguirem aprender sem precisar de um conjunto de treinamento com as respostas esperadas. Como pontos negativos, destacam-se: dificuldade em realizar a validação; complexidade que os modelos podem alcançar; imprevisibilidade dos resultados; utilização de recursos computacionais podem se tornar intensivos (PERERA et al., 2014).
- Baseada em lógica descritiva: é aplicado em conjunto com a representação de contexto ontológico. A modelagem semântica de conceitos (classes), papéis (propriedades, relações) e indivíduos, permite que o conhecimento a ser especificado seja interpretável por máquina. O raciocínio baseado em ontologias é computacionalmente intensivo e o tempo de resposta dependerá em grande parte do tamanho do conjunto de dados e do conjunto de regras presentes na ontologia. Além da complexidade de raciocínio, a concepção de uma ontologia é dita ser uma tarefa complexa, que exige conhecimentos de domínio. Como pontos positivos do raciocínio baseado em lógica descritiva destacam-se: a possibilidade de raciocínio complexo; representação complexa; resultados significativos; possibilidade de validação e verificação da qualidade. Dentre os pontos negativos na sua utilização destacam-se: o baixo desempenho, já que podem exigir maior tempo para realizar o processamento; a necessidade de maior poder computacional (KNAPPMEYER et al., 2013).
- Baseada em lógica probabilística: estas técnicas permitem que as decisões sejam tomadas com base em probabilidades associadas aos eventos. Podem ser utilizadas para combinar os dados dos sensores a partir de diversas fontes. Além

disso, podem ser usadas para identificar as resoluções de conflitos entre contextos. Na maioria das vezes, estas técnicas são usadas para entender ocorrência de eventos. O raciocínio probabilístico é especialmente aplicável em ambientes cientes de contexto devido as potenciais falhas temporárias na comunicação com os sensores e a possibilidade de medições imprecisas de sensores físicos. As principais vantagens desta estratégia são: a possibilidade de combinar eventos; lidar com a incerteza; fornecer resultados moderadamente significativos. Como pontos negativos destacam-se: a dificuldade de só trabalhar com valores numéricos; a necessidade de saber as probabilidades dos eventos. Dentre as técnicas utilizadas nesta estratégia pode-se citar: Naïve Bayes, que é considerada a estratégia generativa mais simples para a classificação de variáveis de classe com uma única dependência de valores, e os *Hidden Markov Models*, que representa dados estruturados sequencialmente, permitindo que o estado seja representado usando evidências observáveis sem ler diretamente o estado (PERERA et al., 2014).

A figura 3 apresenta uma distribuição das técnicas de processamento de contexto utilizadas por trabalhos encontrados na literatura, onde destacam-se as técnicas baseadas em regras, aprendizagem e em lógica probabilística.

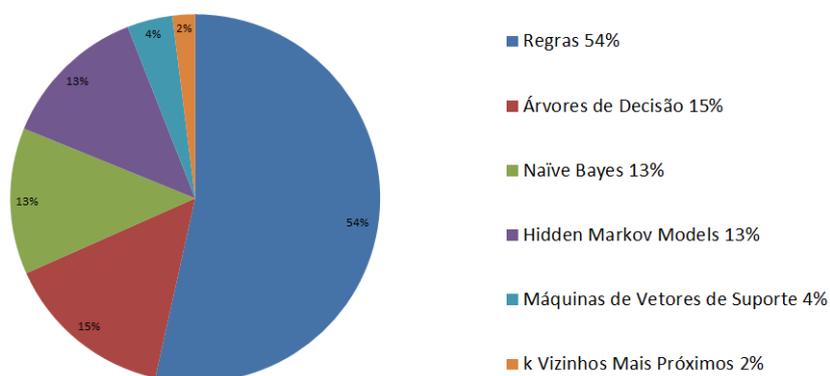


Figura 3: Utilização das técnicas de processamento de contexto  
Fonte: PERERA et al., 2014.

## 2.4 Considerações Finais

Neste capítulo foram abordados aspectos sobre a ciência de contexto, apresentando os benefícios em sua utilização, e foram apresentadas as três etapas básicas presentes para a sua construção: aquisição, modelagem e processamento. Na etapa de aquisição foram abordadas as possíveis fontes utilizadas para adquirir a informação contextual e as características de cada tipo. Ainda, foram apresentadas as principais técnicas utilizadas para realizar a etapa de processamento de contexto, onde destaca-se a técnica baseada em regras, por ser a técnica mais utilizada.

## **3 ABORDAGENS PARA MODELAGEM DE CONTEXTO**

Atualmente, diversas abordagens para modelagem de contexto são encontradas na literatura, sendo que cada uma possui vantagens e desvantagens em relação a sua utilização, desta forma, ainda não se tem uma técnica que seja considerada ideal para representar qualquer tipo de contexto. A seguir são apresentadas algumas das abordagens utilizadas para realizar a modelagem de contexto.

### **3.1 Modelos Chave-Valor**

Esta abordagem de modelagem é a que utiliza a estrutura de dados mais simples para representar a informação contextual. O contexto é representado através de pares compostos por uma chave, responsável por identificar o atributo de contexto, e por um valor associado a essa chave. Os dados podem ser representados em diferentes formatos, como por exemplo, arquivos textos e binários. É um modelo simples de utilizar e manipular, não permitindo estruturas mais sofisticadas que habilitem algoritmos eficientes de recuperação de contexto. Esta é a forma mais simples para representação do contexto, mas a modelagem deste tipo não é escalável e não é adequada para armazenar estruturas de dados complexas. Além disso, não é possível modelar estruturas ou relações hierárquicas (PERERA et al., 2014).

Pode ser utilizada para modelar quantidade limitada de dados, como as preferências do usuário e configurações de aplicativos. Em sua maioria, dados que não estão relacionados e são independentes entre si. Um exemplo de modelagem baseada em pares chave-valor é apresentada na tabela 1, onde, são descritas informações de um usuário. Alguns exemplos de trabalhos que utilizam a modelagem de chave-valor são: (PETRELLI et al., 2000), (VIRGILIO; TORLONE, 2005), (WILSON; LONG; ATKESON, 2005), (HOU et al., 2005), (AHN; 0001, 2006) e (PATEL et al., 2009).

### **3.2 Modelos Baseados em Linguagens de Marcação**

Esta estratégia de modelagem utiliza estruturas de dados hierárquicas, consistindo de *tags* com atributos e conteúdo para armazenar o contexto, sendo considerada uma

Tabela 1: Exemplo de modelagem chave-valor para informações de um usuário

Chave	Valor
Localização	Pelotas, RS
Idade	20 anos
Estado Civil	Solteiro

melhoria sobre a estratégia de chave-valor. A vantagem de usar *tags* de marcação é que estas permitem a recuperação de dados de forma eficiente. Além disso, a validação é suportada através de definições de esquema, e ferramentas de validação estão disponíveis para técnicas populares, tais como, a marcação XML (*eXtensible Markup Language*). Em contraste, as linguagens de marcação não oferecem recursos avançados de expressão que permitam raciocínio sobre as informações contextuais (PERERA et al., 2014). Uma aplicação comum de modelagem baseada em linguagens de marcação é a modelagem de perfis.

Esta abordagem pode ser utilizada como um formato intermediário de organização de dados, bem como um modo de transferência de dados através da rede. Pode também ser usada para separar as estruturas de dados utilizadas por dois componentes em um sistema. Um exemplo é o CSCP (*Comprehensive Structured Context Profiles*) (HELD; BUCHHOLZ; SCHILL, 2002). O CSCP é um modelo que representa o contexto como perfis de sessão, a figura 4 apresenta um exemplo desta estratégia, onde, é realizada a descrição do perfil da entidade de contexto *DeviceProfile*, por meio da informação de contexto *Memory*. Alguns exemplos de trabalhos que utilizam a modelagem baseada em linguagens de marcação são: (SANCHEZ et al., 2006), (HERBERT; O'DONOGHUE; CHEN, 2008), (REETZ; TÖNJES; BAKER, 2010), (CORRADI; FANELLI; FOSCHINI, 2009), (SHAEIB; CAPPELLARI; ROANTREE, 2010), (LI; SEHIC; DUSTDAR, 2010).

### 3.3 Modelos Gráficos

Esta abordagem utiliza elementos gráficos para a representação de contexto, tais como grafos contextuais, ORM (*Object Role Modeling*) e UML (*Unified Modeling Language*). A modelagem utilizando grafos contextuais (BRÉZILLON, 2005) é baseada em redes semânticas e suporta a representação de modelos de tarefas. As estratégias baseadas em ORM (HENRICKSEN; INDULSKA, 2004) envolvem a identificação de fatos e papéis executados pelas entidades. A figura 5 apresenta um exemplo de representação utilizando o modelo ORM, onde o contexto (ou fato) pode ser estático (*static fact type*) ou dinâmico, é considerado estático se permanece inalterado enquanto a entidade que o descreve persistir.

Ainda, analisando a figura 5, caracteriza-se que o contexto dinâmico pode ser pro-

```

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cscp="context-aware.org/CSCP/CSCPProfileSyntax#"
  xmlns:dev="context-aware.org/CSCP/DeviceProfileSyntax#"
  xmlns:net="context-aware.org/CSCP/NetworkProfileSyntax#"
  xmlns="context-aware.org/CSCP/SessionProfileSyntax#"
  <SessionProfile rdf:ID="Session">
    <cscp:default rdf:resource=
      http://localContext/CSCPProfile/previous#Session"/>
    <device><dev:DeviceProfile>
      <dev:hardware><dev:Hardware>
        <dev:memory>9216</dev:memory>
      </dev:Hardware></dev:hardware></dev:DeviceProfile>
    </device>
  </SessionProfile>
</rdf:RDF>

```

Figura 4: Exemplo de modelagem de perfis em CSCP

Fonte: STRANG; LINNHOFF-POPIEN, 2004.

veniente de perfil (*profiled fact type*), percebido (*sensed fact type*) ou derivado (*derived fact type*). Um indicador de qualidade introduzido é um tipo de fato histórico para cobrir o aspecto temporal do contexto (*temporal fact type*). Por fim, destaca-se as dependências entre fatos (*fact dependency*), que indica que uma mudança em um fato conduz a uma mudança automática em um outro fato, representado pela relação *dependsOn*.

As estratégias baseadas em UML utilizam extensões dessa linguagem, como perfis e estereótipos, para representar informações de contexto. Um exemplo desta estratégia é CMP (*Context UML Profile*) (SIMONS, 2007). A seguir são apresentados alguns diagramas da UML e suas finalidades na modelagem de contexto.

- Diagrama de Casos de Uso: modelar os atores envolvidos, relacionados ou influenciados pelo contexto e suas possibilidades de interação.
- Diagrama de Componentes: modelar os sistemas envolvidos que contêm informações relacionadas ao contexto, como bancos de dados, fontes de contexto, entre outros.
- Diagrama de Classes: representar a informação estrutural do domínio e modelar como o contexto está estruturado.
- Diagrama de Sequência: modelar cenários de ativação do contexto. O diagrama detalha o fluxo da informação entre os sistemas envolvidos e exibe a sequência de disseminação da informação.

Os modelos gráficos possibilitam um melhor entendimento da estrutura das informações contextuais, mas possuem um baixo grau de formalismo (STRANG;

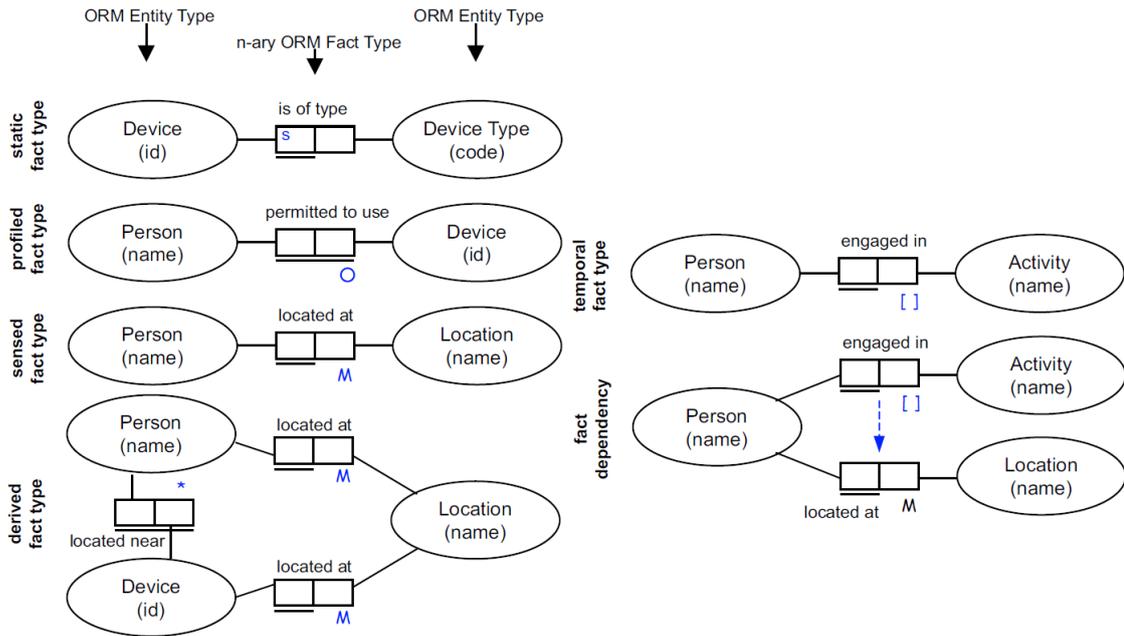


Figura 5: Exemplo de representação do modelo ORM  
 Fonte: STRANG; LINNHOFF-POPIEN, 2004.

LINNHOFF-POPIEN, 2004). Podem ser utilizados para armazenamento em longo prazo de grandes volumes de dados, e os contextos históricos podem ser armazenados em bancos de dados. Alguns exemplos de trabalhos que utilizam modelos gráficos são: (HENRICKSEN; INDULSKA; RAKOTONIRAINY, 2002), (KAENAMPORNAN; O'NEILL, 2004), (COUTAZ et al., 2005).

### 3.4 Modelos Orientados a Objetos

As abordagens orientadas a objetos modelam os dados utilizando hierarquias e relações entre classes e exploram os benefícios do paradigma de programação orientado a objetos, como encapsulamento e reusabilidade. Os detalhes de processamento de contexto são encapsulados no nível de objetos e o acesso às informações de contexto é realizado apenas por meio de interfaces.

Os modelos orientados a objetos são fortes em relação à possibilidade de composição distribuída, pois novos tipos de informações de contexto podem ser adicionados e instâncias podem ser atualizadas de forma distribuída. Porém, normalmente, as infraestruturas de execução que suportam esses modelos exigem muitos recursos dos dispositivos computacionais, o que nem sempre pode ser atendido (STRANG; LINNHOFF-POPIEN, 2004). Além disso, esta estratégia não fornece capacidade de raciocínio sobre os contextos e a avaliação de projetos é dificultada devido à falta de normas e especificações (PERERA et al., 2014).

Esta abordagem pode ser utilizada para representar contexto no nível de código

de programação, permitindo a manipulação do contexto em tempo de execução. Um exemplo dessa estratégia é apresentado na figura 6, onde as informações de contexto são representadas por um conjunto de entidades, que, por sua vez, descrevem objetos físicos ou conceituais, como um canal de comunicação, dispositivo ou uma pessoa. As propriedades das entidades, como o nome da pessoa ou o identificador do canal de comunicação, são representadas por atributos. Uma entidade está relacionada com os seus atributos e outras entidades por meio de relacionamentos de associação. Alguns exemplos de trabalhos que utilizam modelos orientados a objetos são: (CHEVERST; MITCHELL; DAVIES, 1999), (BARDRAM, 2005), (CONAN; ROUVOY; SEINTURIER, 2007), (PAWAR et al., 2009), (WIBISONO; ZASLAVSKY; LING, 2010).

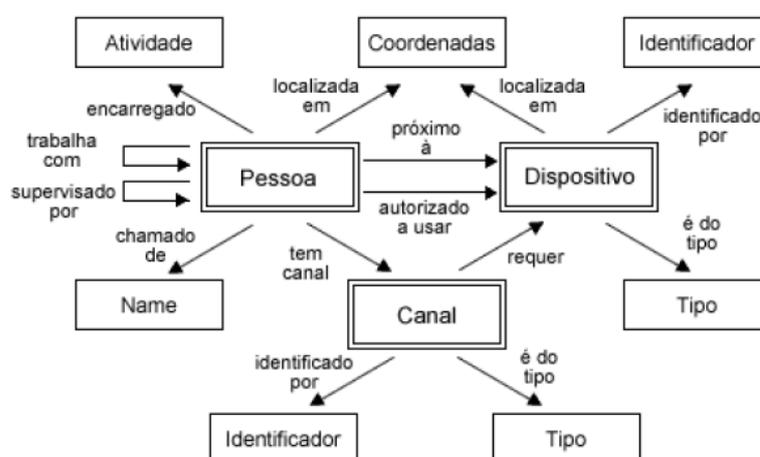


Figura 6: Exemplo de modelo orientado a objetos  
Fonte: FERNANDES, 2009.

### 3.5 Modelos Baseados em Lógica

Nas abordagens baseadas em lógica a informação contextual é modelada como fatos, expressões e regras. Além disso, um processo de inferência pode ser utilizado para derivar novos fatos com base nas regras modeladas, com isso a informação contextual é adicionada, atualizada e removida em termos de fatos ou inferências a partir de regras. Essa estratégia possui um alto grau de formalismo, no entanto, a falta de normalização reduz a possibilidade de reutilização e aplicabilidade, podendo implicar em uma maior dificuldade de manutenção e compreensão (PERERA et al., 2014). A modelagem baseada em lógica permite que novas informações de contexto de alto nível possam ser extraídas utilizando informações de contexto de baixo nível, adicionando assim a capacidade de melhorar outras técnicas de modelagem de contexto.

Esta abordagem pode ser utilizada para gerar eventos, ações no modelo, e definir restrições ou limitações. A figura 7 ilustra parte de um modelo baseado em lógica relacionado à localização do usuário de uma determinada aplicação. Alguns traba-

lhos que utilizam a modelagem baseadas em lógica são: (AKMAN; SURAV, 1997), (BACON; BATES; HALLS, 1997), (GRAY; SALBER, 2001).

```
(H_UserAtLocation (uid ?uid) (rid ?rid)
  (start-time ?start-time))
(H_UserColocation (uid-list ?uid1 ... ?uidn)
  (rid ?region-id) (start-time ?time-value))
(H_UserIsPresent (uid ?uid) (start-time ?time-value))
(L_UserAtLocation (uid ?uid) (x ?x) (y ?y) (z ?z))
```

Figura 7: Exemplo de um modelo baseado em lógica para localização de usuário  
Fonte: KATSIRI; MYCROFT, 2003.

### 3.6 Modelos Baseados em Ontologias

Nesta abordagem, o contexto é organizado em ontologias. Uma ontologia é uma especificação explícita, formal, de uma conceitualização compartilhada em que objetos, conceitos, entidades e relacionamentos do mundo real são definidos em uma determinada área de interesse ou domínio de conhecimento (GRUBER, 1993). Um dos grandes interesses na construção e uso de ontologias é tornar o conhecimento sobre o mundo real processável por máquinas. As ontologias consistem de vários componentes chaves, tais como, indivíduos, classes, atributos, relações, termos de funções, restrições, regras, axiomas e eventos. A figura 8 ilustra parte de uma ontologia para modelagem de contexto, a qual modela informações referentes a classe contexto, onde, a classe *Context* representa todas as informações relacionadas ao contexto do usuário, e toda informação de contexto é monitorado pela classe *Context.listener*.

Uma das vantagens da utilização das ontologias é a possibilidade de raciocínio lógico, o qual pode ser utilizado pelas aplicações para inferir contextos de alto nível a partir de contextos de baixo nível e para checar e resolver inconsistências no conhecimento contextual. O raciocínio é realizado por um motor de inferência o qual permite: juntar diferentes fragmentos ontológicos; deduzir conhecimento a partir de axiomas codificados simbolicamente; consultar instâncias e seus valores; consultar nomes de conceitos e atributos baseados em ontologias conhecidas; validar a consistência de uma ou mais ontologias; atribuir relacionamentos inter-ontológicos; e completar as ontologias através do processamento de hierarquia implícita e relacionamentos baseados em regras existentes (STRANG; LINNHOFF-POPIEN; FRANK, 2003).

Como desvantagens, pode-se destacar a complexidade que a representação pode alcançar e a grande utilização de recursos computacionais necessários para realizar

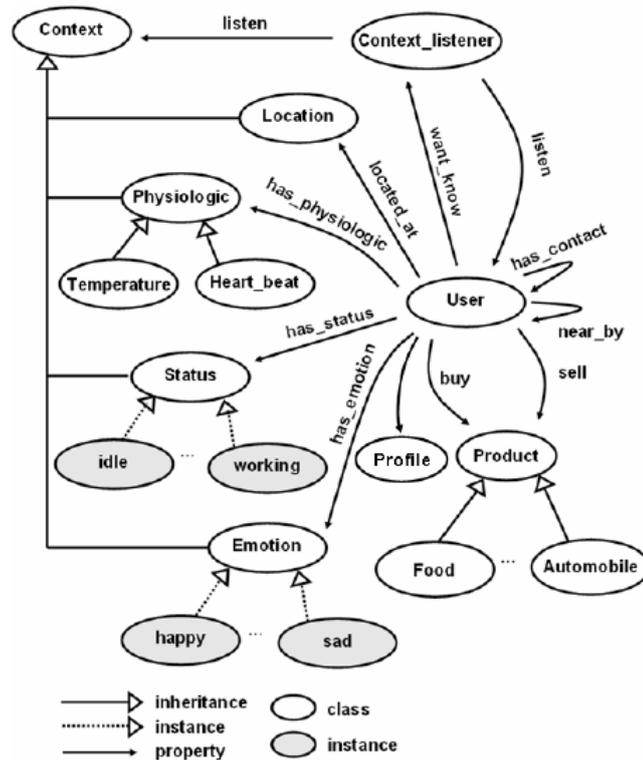


Figura 8: Exemplo de uma ontologia referente a informações de usuário  
 Fonte: ALMEIDA; SOUZA BAPTISTA; ANDRADE, 2006.

a recuperação das informações.

Esta abordagem pode ser utilizada para modelar o conhecimento do domínio e a estrutura do contexto baseada nos relacionamentos definidos pela ontologia. Alguns exemplos de trabalhos que utilizam a modelagem baseada em ontologias são: (STRANG; LINNHOF-POPIEN; FRANK, 2003), (JACOB et al., 2006), (RAMPARANY et al., 2007), (PIETSCHMANN et al., 2008), (RIBONI; BETTINI, 2009), (ZHOU et al., 2009).

### 3.7 Análise das Abordagens para Modelagem de Contexto

Visando realizar uma comparação entre os modelos de contexto mais utilizados, os autores em (STRANG; LINNHOF-POPIEN, 2004) propuseram uma serie de métricas que podem ser avaliadas:

- composição distribuída (dc): a composição e administração dos modelos de contexto devem ser extremamente dinâmicas em termos do tempo, topologia da rede e recursos;
- validação parcial (pv): capacidade para validar conhecimento parcial. Em determinados momentos, devido a composição distribuída, não é possível validar todo o conhecimento de contexto;

- qualidade da informação (qua): a qualidade da informação muda de acordo com o sensor utilizado, entre outros fatores. Os métodos devem suportar o tratamento de informação com distintos níveis de qualidade;
- incompletude e ambiguidade (inc): o método deve ser capaz de tratar informações incompletas e ambíguas;
- nível de formalidade (for): modelos com sintaxe e semântica bem definidas;
- aplicabilidade em ambientes já existentes (app): utilização de modelos em aplicações já existentes.

A tabela 2 apresentada uma comparação entre alguns modelos de contexto, onde o sinal de (-) significa limitação do modelo, (- -) limitação ainda maior, o sinal de (+) o atendimento da métrica e o (++) o atendimento de uma forma mais satisfatória.

Tabela 2: Comparação entre abordagens de modelagem de contexto

Modelo	dc	pv	qua	inc	for	app
Chave-Valor	-	-	--	--	--	+
Linguagem de Marcação	+	++	-	-	+	++
Gráfico	--	-	+	-	+	+
Orientado a Objetos	++	+	+	+	+	+
Lógico	++	-	-	-	++	--
Ontológico	++	++	+	+	++	+

Fonte: STRANG; LINNHOFF-POPIEN, 2004.

Como pode ser observado na tabela 2, cada técnica possui vantagens e desvantagens em relação a sua utilização, sendo que ainda não foi encontrada uma técnica que seja considerada a ideal, com isso, têm surgido as abordagens híbridas. Os modelos híbridos de modelagem de contexto são considerados os mais promissores, pois combinam diferentes técnicas de modelagem, com diferentes níveis de interpretação, para diferentes aspectos. Devido a importância dos modelos híbridos para este trabalho, no capítulo 4 são apresentadas algumas propostas para modelagem híbrida de contexto, juntamente com as suas principais características.

A utilização de modelagens híbridas traz um novo desafio, que é como realizar o gerenciamento do contexto, já que são combinados diferentes modelos e/ou várias formas de armazenar as informações contextuais. Com isso, são necessários mecanismos que garantam o acesso às informações contextuais já capturadas para que sejam disponibilizadas de forma atualizada às aplicações interessadas, tudo isto preservando a consistência destas informações.

### 3.8 Considerações Finais

Neste capítulo foram apresentadas as principais abordagens utilizadas durante a etapa de modelagem de contexto, identificando as vantagens e desvantagens em sua utilização. Ao final são apresentadas as modelagens baseadas na integração de modelos, modelagens híbridas, as quais vêm ganhando destaque na literatura e se apresentando como a forma mais adequada para modelar contexto de diferentes naturezas, por aproveitar os benefícios da utilização de cada modelo, e assim, melhorando a modelagem dos dados contextuais.

Com a utilização de modelos híbridos de contexto destaca-se ser importante o estudo de estratégias que possibilitem processar as informações contextuais de diferentes naturezas, as quais podem estar armazenadas em diferentes modelos de armazenamento. Apesar da utilização de modelos híbridos de contexto já ser aplicada nos últimos anos, ainda nota-se a ausência de estratégias que possibilitem a combinação das informações presentes nos diferentes modelos. Geralmente as aplicações cientes de contexto utilizam estratégias que processam o contexto de forma independente, por exemplo, utilizando o raciocínio dos modelos baseado em ontologia para uma aplicação específica e um processamento com base em regras, utilizando dados armazenados em modelo relacional, para outra finalidade. Com a utilização de uma estratégia que possibilite a combinação das informações presentes nos diferentes modelos, uma mesma aplicação pode se aproveitar de diferentes informações podendo, por exemplo, utilizar informações semânticas das ontologias juntamente com as informações que estão no modelo relacional e, desta forma, melhorar a identificação das situações de interesse.

## 4 TRABALHOS RELACIONADOS A MODELAGEM HÍBRIDA DE CONTEXTO

Os trabalhos de modelagem híbrida de contexto não apresentam a forma com que processam as informações contextuais internamente, desta forma, não foram encontradas abordagens específicas sobre a forma de acesso e iteração dos modelos. Com isso, na sequência são apresentados alguns dos principais trabalhos referentes a modelos híbridos de contexto, explorando aspectos referentes à sua arquitetura, sua modelagem, bem como suas principais funcionalidades.

### 4.1 Modelo Gráfico e de Ontologias

O artigo *Towards a hybrid approach to context modelling, reasoning and interoperation* (HENRICKSEN; LIVINGSTONE; INDULSKA, 2004) apresenta uma solução híbrida de modelagem de contexto, combinando as vantagens do modelo baseado em ontologias com a modelagem gráfica. É explorada a integração dos conceitos ontológicos com a abordagem baseada em fatos CML (*Context Modelling Language*), a qual é uma extensão do modelo ORM, com o intuito de produzir uma solução híbrida para modelagem e processamento de contexto.

O processamento de contexto através de ontologia é utilizado para derivar contextos adicionais, sendo baseado em regras e propriedades, tais como, transitividade e comutatividade, e ainda é responsável por realizar a detecção e a correção de inconsistências nas informações contextuais.

De forma a realizar o mapeamento do modelo CML para o modelo ontológico, que foi desenvolvido utilizando a linguagem OWL-DL (*Web Ontology Language-Description Logics*), foram criadas duas ontologias. A primeira ontologia define a modelagem CML em termos de classes e propriedades, esta ontologia possui dois componentes: o primeiro formado pelos conceitos para criação dos modelos CML; e o segundo formado pelos conceitos para criação das instâncias. A segunda ontologia foi desenvolvida para definir um modelo de exemplo, a qual também foi dividida em duas partes, a primeira descrevendo as propriedades do modelo e a segunda definindo os objetos e as

classes que suportam a instanciação do modelo.

Os autores destacam que não foi possível mapear completamente os conceitos presentes em CML para OWL-DL, particularmente devido à dificuldade para encontrar a representação adequada de chave única e os relacionamentos entre fatos alternativos.

## 4.2 Modelo Orientado a Objetos e de Ontologias

Nesta seção são destacados os seguintes trabalhos: ConDoR; MoCA; iTransIT.

### 4.2.1 ConDoR

Em (PAGANELLI; BIANCHI; GIULI, 2007) é apresentado o modelo de contexto ConDoR (*Context Distribution and Reasoning*), o qual possui como base uma abordagem de modelagem híbrida de contexto, com o objetivo de combinar as vantagens dos modelos orientado a objetos e baseado em ontologia. O modelo ConDoR é composto por dois submodelos: o contexto orientado a objetos CADDM (*Context Acquisition and Distribution Data Model*), que define o formato para troca de dados entre componentes distribuídos (por exemplo, sensores e aplicações), e o contexto baseado em ontologia CRDM (*Context Reasoning Data Model*), que é um modelo com propriedades semânticas, o qual permite raciocínio em dados de contexto.

Os submodelos podem ser divididos em dois níveis de representação, sendo estes:

- nível conceitual: fornece uma representação abstrata dos principais conceitos e relações, independentemente dos detalhes de implementação. Este nível fornece um formato de representação uniforme de ambos os modelos CADDM e CRDM.
- nível físico: adota paradigmas específicos de modelagem e tecnologias. O CADDM é representado na notação XML, pois fornece instrumentos para a definição de gramáticas para a troca de dados entre componentes distribuídos e suporta os conceitos de orientação a objetos, tais como polimorfismo e herança. O CRDM baseia-se na linguagem OWL, devido as suas características para representação da ontologia.

Como mecanismo de processamento e raciocínio de contexto são utilizadas as técnicas baseada em regras, as quais fornecem um mecanismo flexível para inferir contexto de alto nível por meio de regras definidas pelo usuário e técnicas de raciocínio próprias das ontologias, as quais podem ser usadas para determinar conceitos de satisfabilidade, subordinação de classe, consistência e verificação de instâncias.

#### 4.2.2 Modelo Híbrido do Middleware MoCA

No artigo *Promoting efficiency and separation of concerns through a hybrid model based on ontologies for context-aware computing* (ROCHA; CASANOVA; ENDLER, 2007) é apresentado um modelo de contexto híbrido para o *middleware* MoCA (*Mobile Collaboration Architecture*), que é um *middleware* baseado em serviços para o desenvolvimento de aplicações colaborativas ciente de contexto. A abordagem híbrida consiste da adição de um modelo baseado em ontologia (ONTO-Moca) ao modelo orientado a objetos, presente no *middleware*.

A abordagem foi proposta de forma que tanto os modelos, os mecanismos de verificação dos modelos, quanto os respectivos métodos de inferência sejam complementares, ao invés de serem equivalentes, já que o modelo baseado em ontologia implementa apenas um subconjunto do modelo orientado a objetos. A fim de garantir a consistência entre os dois modelos, foram desenvolvidos dois mecanismos, um o estático e o outro dinâmico, sendo eles responsáveis por realizar as atualizações em um modelo enquanto o outro é alterado.

O mecanismo estático é utilizado para realizar o mapeamento dos conceitos criados no modelo orientado a objetos para o modelo ontológico em tempo de implantação do contexto, utilizando um documento XML que descreve o modelo de contexto e o CT (*Context Tool*), responsável por validar a semântica do documento e o corrente estado do sistema. O mecanismo dinâmico é utilizado quando a informação de contexto deve ser atualizada no modelo baseado em ontologia, devido às mudanças no estado do ambiente. A manutenção dinâmica da ontologia é considerada o ponto crítico para o desempenho do sistema ciente de contexto.

A figura 9 mostra as operações que mantêm a coerência entre os dois modelos, mostrando os dois tipos de usuários que podem interagir com cada modelo: o desenvolvedor de aplicações cientes de contexto, e um desenvolvedor de provedor de contexto, que é responsável pela introdução de novos sensores ou mecanismos de inferência.

O desenvolvedor de aplicativos utiliza o modelo baseado em ontologia para entender como o estado do ambiente é descrito nos modelos em termos de informações de contexto, utilizando contextos de alto nível. Enquanto o desenvolvedor de provedor de contexto atua no modelo orientado a objetos, que descreve o comportamento em tempo de execução das informações de contexto que estão sendo publicadas.

Para implementação do mapeamento estático, foram utilizados os módulos incluídos no CT de forma a realizar o acesso ao modelo de contexto orientado a objetos (*MoCA's Context Model*). Para realizar a atualização das classes do modelo ontológico (ONTO-MoCA), como também para verificar as possíveis inconsistências que um novo tipo de contexto podem introduzir, foram utilizados módulos da ferramenta Jena (JENA, 2014).

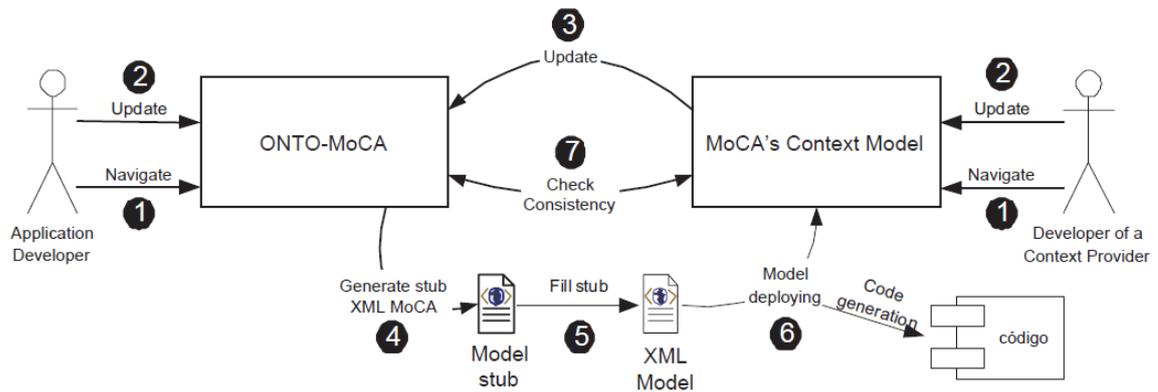


Figura 9: Processo para manter a consistência no *middleware* MoCA  
 Fonte: ROCHA; CASANOVA; ENDLER, 2007.

No mapeamento dinâmico, o modelo baseado em ontologia é atualizado com os novos conceitos e eventos adicionados ao modelo orientado a objetos em tempo de execução. A fim de não comprometer o desempenho do sistema e escalabilidade, as instâncias do contexto no modelo orientado a objetos não estão representados no modelo ontológico, mas apenas suas transformações.

#### 4.2.3 Modelo Híbrido do Framework iTRANSIT

O artigo *A Hybrid Approach to Context Modelling in Large-Scale Pervasive Computing Environments* (LEE; MEIER, 2009) apresenta uma abordagem híbrida para modelagem de contexto, combinando as modelagens orientada a objetos e baseada em ontologias, as quais são intituladas respectivamente de PCM (*Primary-Context Model*) e PCont (*Primary-Context Ontology*). O modelo PCM define as abstrações para a modelagem de informações e os contextos associados como objetos espaciais. Os Objetos espaciais foram concebidos como uma série de objetos abstratos para permitir a gestão, comunicação e acesso à informação e, para apoiar a correlação de informações. Os Objetos espaciais representam as informações e contextos como uma série de parâmetros estruturados como triplas nome-tipo-valor.

O modelo PCont é dividido em *Upper PCont* e *Extended PCont*. O *Upper PCont* define formalmente os objetos espaciais do PCM e o contexto que deve ser associado. *Extended PCont*s são sub-ontologias do *Upper PCont* e fornecem uma representação formal dos subdomínios específicos dentro de um ambiente ubíquo, por exemplo, em um ambiente de transporte, *Extended PCont*s poderiam descrever o clima ou o transporte público.

O PCM e PCont combinam os benefícios das técnicas de modelagem orientadas a objetos, tais como os benefícios de gerenciamento e comunicação, com a representação semântica e os mecanismos de inferência dos modelos baseados em ontologias. Os modelos foram projetados especificamente para o área de transporte,

fazendo parte do *framework* iTransIT, uma arquitetura de *middleware* para integrar os sistemas de transporte inteligentes em um ambiente urbano. Os modelos promovem uma abordagem para correlacionar diversos conjuntos de dados independentes e possivelmente distribuídos. O PCM fornece as tarefas de armazenamento, gerenciamento e compartilhamento de contexto de uma forma escalável. O PCOnt especifica formalmente os conceitos que podem ser referidos pelo PCM, bem como as relações entre os conceitos.

A figura 10 mostra a arquitetura do *framework* iTransIT, mostrando o *Upper PCOnt* que é a representação ontológica de PCM, também mostra que os sistemas de iTransIT podem modelar os contextos em termos dos conceitos e relações definidas em uma ou mais *Extended PCOnt*. Onde os módulos *Extended PCOnt* podem ser específicos para um único sistema iTransIT, ou podem abranger vários sistemas, destacando que os módulos *Extended PCOnt* permitem consultas mais expressivas em relação aos módulos *Upper PCOnt*.

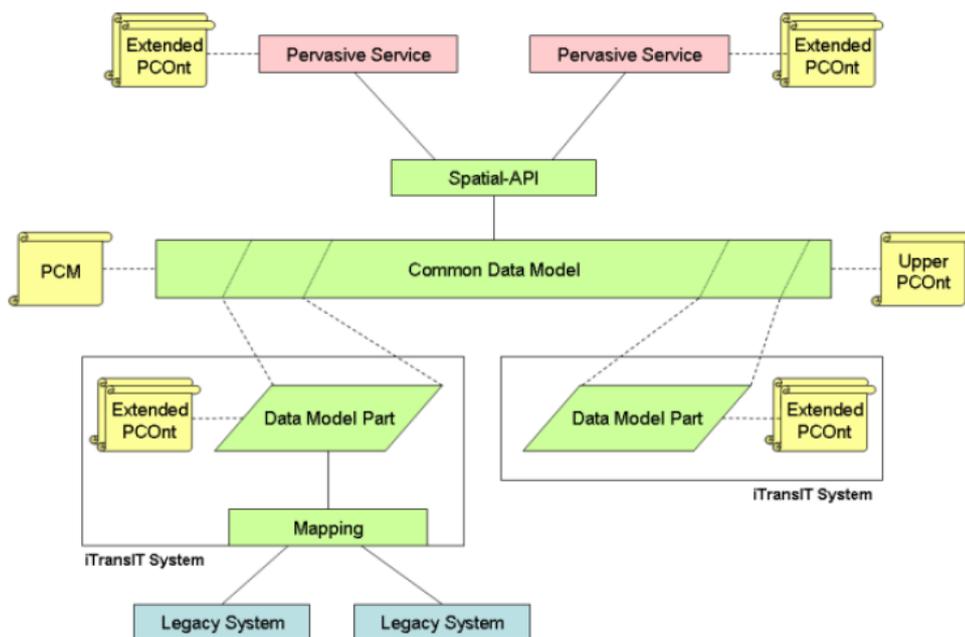


Figura 10: Modelagem híbrida para o *framework* iTransIT  
Fonte: LEE; MEIER, 2009.

### 4.3 Modelo Relacional e de Ontologias

CoCA (*Collaborative Context-Aware Service Platform*) (EJIGU; SCUTURICI; BRUNIE, 2007) é uma plataforma que realiza as tarefas de representação, raciocínio, agregação e interpretação de dados de contexto. As ações e decisões são realizadas com base nos dados de contexto adquiridos, e a plataforma suporta a colaboração e o compartilhamento de recursos computacionais entre os dispositivos. A figura 11

apresenta a arquitetura da plataforma CoCA dividida em camadas, mostrando suas principais funcionalidades.

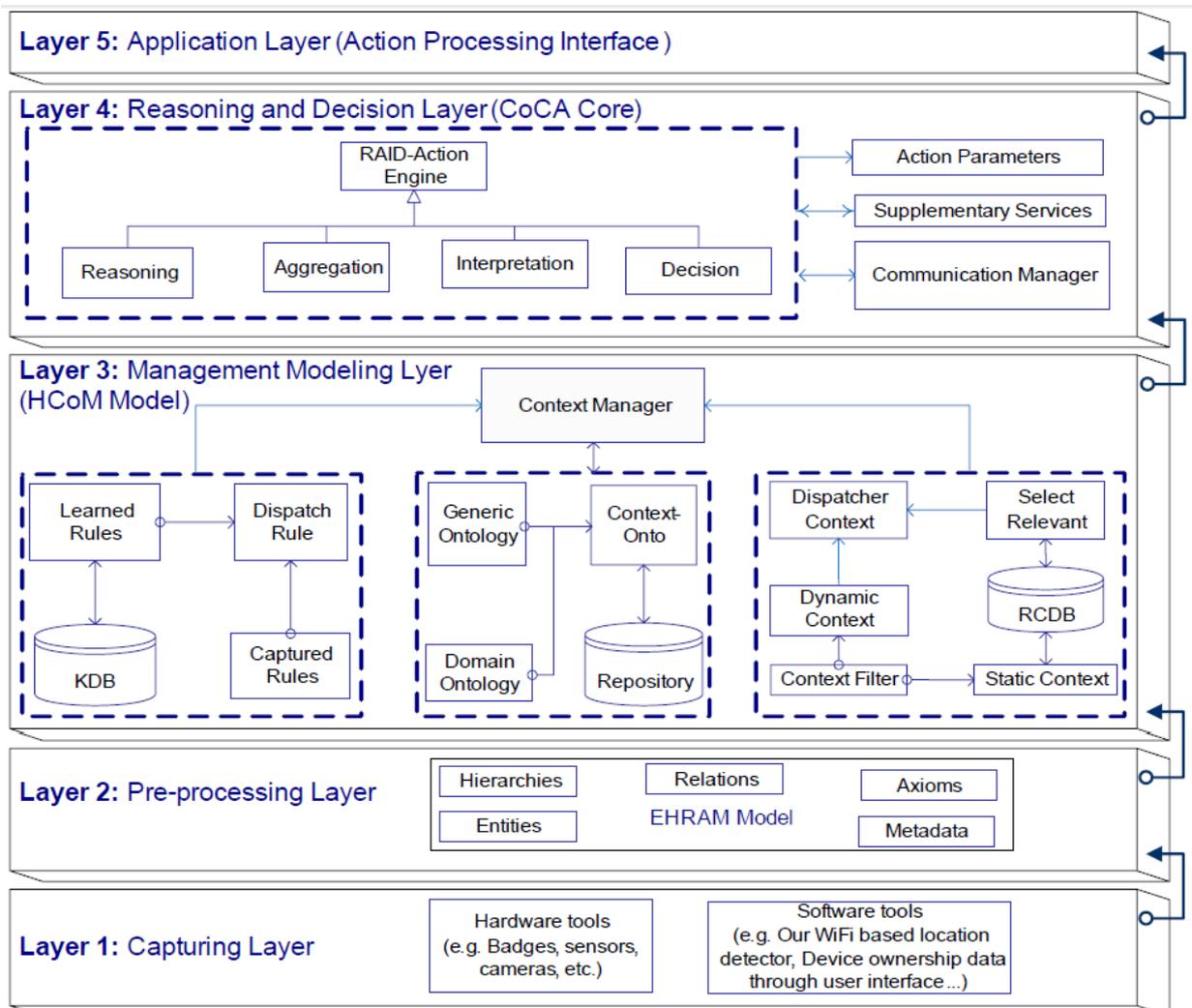


Figura 11: Arquitetura da plataforma CoCA  
Fonte: EJIGU, 2007.

Camada 1: camada de aquisição, responsável por lidar com as ferramentas de aquisição de dados, tanto as ferramentas de hardware, sensores e câmeras, como as de software, onde destaca-se o rastreador local, que converte um sinal Wi-Fi (*Wireless Fidelity*) em um nome de localização significativa. A interface da plataforma é construída com base em APIs (*Application Programming Interface*), devido ao fato de ter sido concebida para ser utilizada por aplicações de diferentes domínios.

Camada 2: a camada de pré-processamento, é usada para formalizar e preparar os dados capturados para posterior processamento. Trata-se de uma modelagem conceitual dos dados capturados de acordo com o formalismo de representação do contexto EHRAM (*Entities Hierarchies Relations Axioms Metadata*), que realiza a separação de dados em entidades, hierarquias, relações, axiomas e metadados. Isto permite

organizar e processar os dados de contexto e dados semânticos de contexto separadamente.

Camada 3: a camada de modelagem de gerenciamento de contexto, trata de como são organizados os contextos úteis para o raciocínio. A representação formal destes dados é realizada por meio do modelo HCoM (*Hybrid Context Management*) (EJIGU; SCUTURICI; BRUNIE, 2008), que é uma estratégia híbrida que combina a característica semântica das ontologias e os esquemas relacionais. Onde é destacado que sistemas de gerenciamento de banco de dados por si só não podem ser usados para gerenciar contexto, e que as ontologias podem não ter um bom desempenho em termos de eficiência e de processamento de consultas com grandes volumes de dados, tornando-se assim, necessária a estratégia híbrida. Os principais componentes desta camada são:

- *Context Manager*: agrega os resultados e envia os dados para o motor de raciocínio;
- *Collaboration Manager*: responsável por tentar reunir mais dados de outras fontes de contexto possíveis, caso seja necessário;
- *Context Filter*: responsável por realizar a validação e decidir se o contexto precisa ser armazenado em RCDB (*Relational Context Database*);
- *Context Selector*: com base na solicitação do usuário decide o contexto que deve ser usado no processamento baseado na precisão, tempo e recursos computacionais exigidos;
- *Context-onto*: gerencia as ontologias e funciona como um repositório;
- *Rules and Policy*: permite aos usuários adicionar regras para o sistema;
- RCDB: armazena o contexto capturado em um sistema de gerenciamento de banco de dados;
- *Rule-Mining*: uma base de dados que consiste em regras que dizem quando as ações devem ser executadas;
- Interfaces: fornece interfaces para os consumidores de contexto.

Camada 4: camada de raciocínio e decisão, o núcleo da plataforma CoCA, é o lugar onde o raciocínio e as decisões cientes de contexto são executadas. Fornece o núcleo de serviço de ciência a contexto após o raciocínio sobre os componentes. Trata-se do motor de ação RAID (*Reasoning, Aggregation, Interpretation, Decision and Action*) que preenche a ontologia com os dados de contexto e, em seguida, aplica-se as regras e axiomas para o raciocínio e decisão sobre as ações a serem executadas.

Outra tarefa que é realizada nesta camada é a agregação, através da combinação de dois ou mais contextos de baixo nível para gerar um contexto de alto nível, o qual é mais significativo. Um exemplo de agregação realizada poder ser a combinação das informações referentes à temperatura corporal, frequência cardíaca e pressão arterial de um paciente para conseguir decidir o seu estado de saúde. Nesta camada também se encontram alguns serviços suplementares, tais como, serviços de descoberta de conhecimento, o processo de adicionar recursos de forma a aumentar à capacidade da plataforma, as tarefas relacionadas à privacidade e gerenciamento de segurança de serviço da plataforma.

Camada 5: camada de aplicação, é nesta camada onde as ações são desencadeadas.

#### **4.4 Modelo Relacional, Linguagem de Marcação e de Ontologias**

Em (KOTENKO; POLUBELOVA; SAENKO, 2012) é apresentada uma abordagem de implementação de um repositório híbrido para sistemas SIEM (*Security Information and Events Management*) que combina as técnicas de modelagem de bancos de dados relacionais, baseadas em XML e em ontologias. Os autores justificam que a representação relacional não permite especificar uma consulta parametrizada, por exemplo, com os nomes dos produtos e versões, para a análise de vulnerabilidades e processá-las no programa, o que torna este processo demorado. Com isso, uma solução alternativa na representação de dados em sistemas de processamento de informações com estruturas complexas de dados (tais como sistemas SIEM) é uma abordagem ontológica, o que facilita as expressões de relações complexas entre entidades e possibilita a formulação de conceitos e de relacionamentos envolvidos.

Segundo os autores, as mudanças no modelo de dados ontológico exigem menor esforço que no modelo relacional. O que é particularmente relevante em áreas onde é necessário armazenar diferentes tipos de informações que podem ser rapidamente alterados. O acoplamento fraco de ontologias de domínio facilita as tarefas de adição, exclusão. Além disso, os componentes das ontologias podem ser combinados de forma dinâmica, para satisfazer as necessidades específicas da aplicação. Com o uso de ontologias é necessário uma abordagem que permita criar um modelo que possa ser aplicado de forma rápida e flexível para todos os conceitos necessários em sistemas SIEM, com isso foi escolhido pelos autores o modelo em formato de triplas, o qual fornece uma representação ontológica para modelagem dos dados.

Na figura 12 é apresentada a arquitetura do repositório para sistemas SIEM. A arquitetura foi concebida com base nos princípios de SOA (*Service-Oriented Architecture*), sendo a arquitetura dividida nas camadas de *Repository storage* e *Repository service*, na qual destaca-se na camada de armazenamento as três abordagens uti-

lizadas para realizar a modelagem dos dados, sendo eles: o modelo relacional; o armazenamento em triplas; e o baseado em XML.

Na camada de serviços destacam-se os módulos de acesso aos dados, de apresentação e de controle de acesso. O módulo de acesso aos dados interpreta as consultas recebidas dos componentes SIEM, transformando no formato da notação correto para realizar o acesso aos dados. O módulo de controle é responsável por verificar os direitos de acesso das consultas realizadas no repositório, de acordo com as permissões definidas. O módulo de apresentação trata a interação com o usuário e o sistema SIEM, as principais características são de mapeamento de informações e interpretação de comandos do usuário com a sua conversão em operações correspondentes no contexto do domínio e fonte de dados.

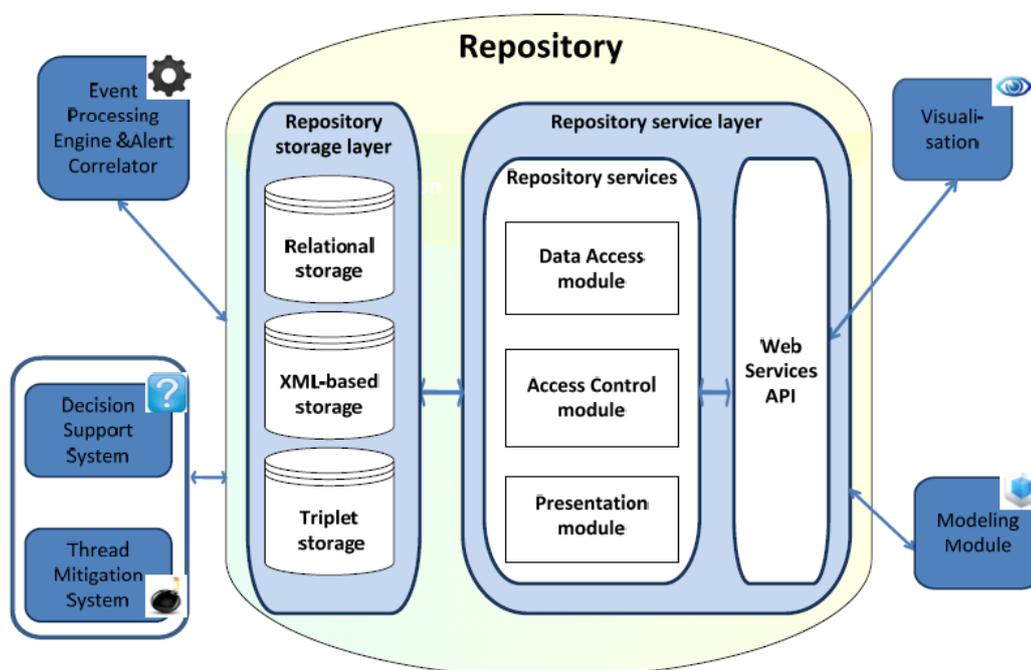


Figura 12: Arquitetura do repositório para sistemas SIEM  
Fonte: KOTENKO et al., 2013.

## 4.5 Modelo Objeto Relacional e de Ontologias

O artigo *Context Management Using Ontology and Object Relational Database Management System* (VANATHI, 2013) apresenta um sistema ciente de contexto que baseia-se em uma modelagem híbrida de contexto, a qual combina os modelos objeto relacional e ontológico. O modelo objeto relacional é criado a partir dos dados de contexto estáticos, e são atualizados em tempo de execução com os novos dados de contexto capturados. O modelo baseado em ontologia *Context-Onto* é composto de três componentes básicos: o esquema da ontologia, dados de contexto estático e dados de contexto dinâmicos. O esquema ontologia representa as hierarquias de con-

ceito, conceito de relações, axiomas e informações de metadados, que representam o conhecimento.

Na figura 13 é apresentada a arquitetura do sistema ciente de contexto, sendo esta descrita como uma arquitetura de três camadas: *Context Acquisition*; *Context Middleware* o qual é dividido em três partes, *Context Representation*, *Context Management* e *Context Reasoning*; e *Application*.

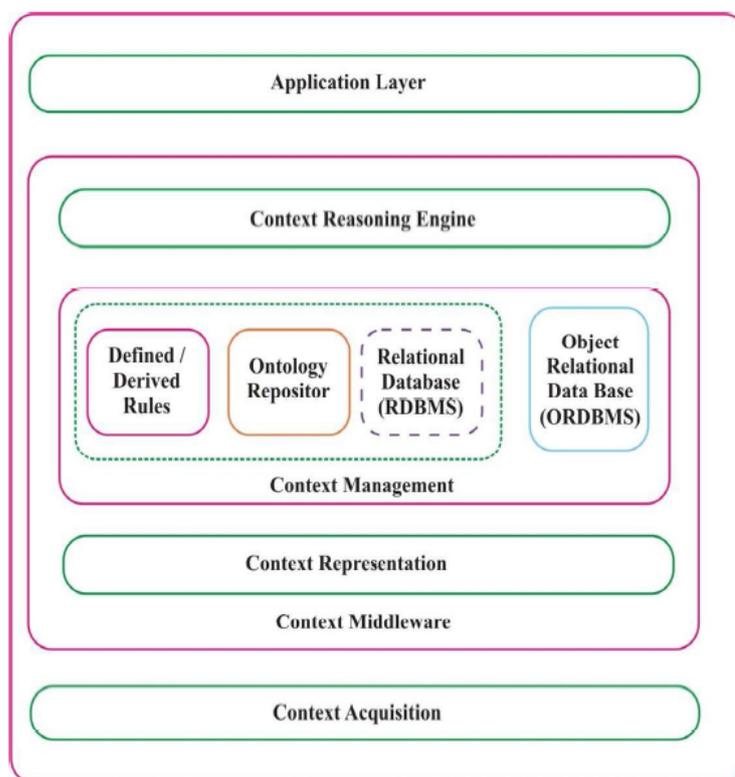


Figura 13: Diagrama de Bloco Funcional do sistema ciente de contexto  
Fonte: VANATHI, 2013.

- *Context Acquisition*: camada responsável pela detecção de dados contextuais e a captura dos mesmos. Um requisito importante desta camada é a consideração dos diferentes contextos presentes no ambiente, onde cada tipo de contexto pode ser modelado com características diferentes, tais como, pessoa, tempo, recursos e localização;
- *Context Representation*: componente responsável pelo fornecimento do esquema de representação, que fornece estruturação e recuperação eficiente. Os dados contextuais são representados num formato padrão, de modo a auxiliar o processo de compartilhamento de contexto. Foi dada ênfase ao modelo de representação de contexto com base na hierarquia de entidades de contexto;
- *Context Management* : componente responsável por determinar como armazenar e utilizar as informações contextuais. Os dados de contexto são armazena-

dos com base no seu estado (estático ou dinâmico). Dados de contexto estáticos são armazenados no banco de dados e os dinâmicos são armazenados na ontologia.

- *Context Reasoning*: componente que determina como o contexto desejado é obtido. Carregam-se apenas os dados relevantes para o raciocínio, minimizando o tamanho do espaço de raciocínio, e assim, reduzindo a sobrecarga dos contextos desnecessários, o que melhora o tempo de resposta do raciocínio de contexto.

## 4.6 Considerações Finais

Este capítulo apresentou trabalhos que utilizam abordagens de modelagem híbrida de contexto, onde foram apresentados as suas principais características, e modelos utilizados em cada trabalho. A tabela 3 apresenta uma análise comparativa entre os trabalhos discutidos neste capítulo, mostrando os tipos de sensores empregados na etapa de aquisição, os modelos utilizados para realizar a modelagem de contexto, as técnicas presentes para realizar o processamento, a forma de armazenamento dos dados contextuais e ainda, algumas ferramentas utilizadas para o desenvolvimento do trabalho.

Com base na tabela 3 é constatado que na etapa de aquisição todos os trabalhos utilizam como fonte de dados sensores físicos denotando a importância dos dados oriundos do ambiente físico. Além disso, observa-se que os trabalhos com suporte apenas para sensores físicos são mais focados em aplicações específicas, enquanto os que utilizam diferentes tipos de sensores podem atender aplicações de diferentes naturezas.

Destaca-se que todos os trabalhos analisados utilizam o modelo baseado em ontologias, principalmente devido as suas características de representação semântica. A utilização de modelos ontológicos segue a tendência da análise realizada em (STRANG; LINNHOFF-POPIEN, 2004), o qual destacou este tipo de modelagem, dentre os modelos analisados, como sendo o mais propício a ser utilizado para representação de contexto.

Em relação ao armazenamento das informações contextuais, nota-se que somente um trabalho possui armazenamento diferenciado para tratar às informações presentes no modelo baseado em ontologia, utilizando para esta finalidade um modelo no formato de triplas. Acredita-se serem necessários mecanismos diferenciados para armazenamento de contexto, já que utilizando diferentes abordagens de modelagens de dados, passa-se a trabalhar com diferentes tipos de dados, níveis semânticos entre outros detalhes, e a representação destes dados de forma eficiente é uma tarefa importante, pois pode facilitar o acesso, modificação e atualização das informações contextuais.

Tabela 3: Comparação entre os trabalhos de modelagem híbrida de contexto

Trabalhos	Aquisição	Modelos	Estratégia Processamento	Armazenamento	Ferramentas
HENRICKSEN; LIVINGSTONE; INDULSKA, 2004	Sensores Físicos Lógicos	Gráfico Ontologia	Regras Ontologia	Base Relacional	Informação Não Disponível
PAGANELLI; BIANCHI; GIULI, 2007	Sensores Físicos	Orientado a Objetos Ontologia	Regras Ontologia	Base Relacional	Jena MySQL
ROCHA; CASANOVA; ENDLER, 2007	Sensores Físicos Lógicos Virtuais	Orientado a Objetos Ontologia	Ontologia	Informação Não Disponível	Jena
EJIGU; SCUTURICI; BRUNIE, 2007	Sensores Físicos Lógicos Virtuais	Ontologia Relacional	Regras Aprendizagem de Máquina Ontologia	Base Relacional	Jena Protégé MySQL
LEE; MEIER, 2009	Sensores Físicos	Orientado a Objetos Ontologia	Ontologia	Informação Não Disponível	Protégé
KOTENKO; POLUBELOVA; SAENKO, 2012	Sensores Físicos Lógicos Virtuais	Linguagem de Marcação Ontologia Relacional	Regras Ontologia	Base Relacional Triplas	Virtuoso Derby
VANATHI, 2013	Sensores Físicos Lógicos Virtuais	Ontologia Relacional Objeto Relacional	Regras Ontologia	Objeto Relacional	PostgreSQL

Os trabalhos apresentados apesar de utilizarem modelos híbridos de contexto, não apresentam uma estratégia de processamento que permita uma aplicação combinar dados presentes nos diferentes modelos, podendo somente utilizar as informações de forma separada, normalmente possuindo finalidade específica de uso. Desta forma, nota-se a ausência de estratégias de permitam a combinação das informações contextuais dos diferentes modelos, e a criação destas estratégias aumenta a flexibilidade para as aplicações utilizarem os contextos de interesse, como também podem facilitar a busca pelas informações contextuais.

## 5 ESCOPO DO TRABALHO

Neste capítulo é caracterizado o *middleware* EXEHDA, por este ser parte importante na concepção deste trabalho, na sequência é apresentado um estudo sobre armazenamento de triplas, e são apresentadas as tecnologias utilizadas para a implementação da EXEHDA-HM.

### 5.1 Middleware EXEHDA

O EXEHDA é um *middleware* que visa criar e gerenciar um ambiente ubíquo, oferecendo para isso diversos serviços, dentre os quais aquele que é direcionado ao provimento de ciência de contexto às aplicações. Sua arquitetura é distribuída e oferece suporte à aquisição, processamento e armazenamento de informações contextuais (LOPES et al., 2014).

A arquitetura de software do *middleware* EXEHDA é apresentada na figura 14. Os principais requisitos que o EXEHDA deve atender são: (i) gerenciar de modo independente tanto aspectos não funcionais como funcionais da aplicação; (ii) dar suporte à adaptação dinâmica de aplicações; (iii) disponibilizar mecanismos para obter e tratar informações contextuais; (iv) empregar informações contextuais na tomada de decisão; (iv) decidir as ações adaptativas de forma colaborativa com a aplicação; e (v) disponibilizar a semântica siga-me, permitindo ao usuário iniciar as aplicações, acessar dados a partir de qualquer lugar e executar as aplicações continuamente mesmo em deslocamento (GUSMÃO, 2013).

Considerando o escopo deste trabalho, destaca-se que o Subsistema de Adaptação e Reconhecimento de Contexto apresentado na figura 14, está organizado em duas abstrações sendo representadas respectivamente por: (i) Servidor de Borda, o qual é responsável pela interação com o meio através de sensores e atuadores; e (ii) Servidor de Contexto, que atua no armazenamento e processamento das informações contextuais.

O ambiente ubíquo gerenciado pelo *middleware* EXEHDA é formado por equipamentos multi-institucionais, o que gera a necessidade de adotar procedimentos de

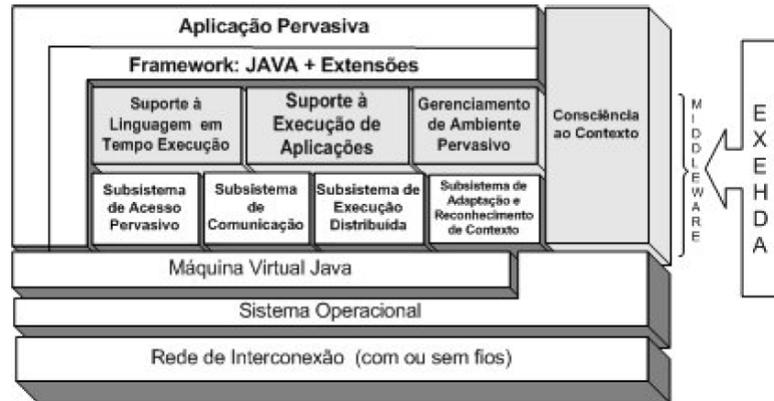


Figura 14: Arquitetura de software do middleware EXEHDA  
Fonte: GUSMÃO, 2013.

gerência iguais aos utilizados em ambientes de Grade Computacional (LOPES et al., 2014). O gerenciamento da organização celular do ambiente ubíquo resguarda a autonomia das instituições envolvidas. Dessa forma, a composição do ambiente ubíquo envolve tanto os dispositivos dos usuários, como os equipamentos de suporte necessários. A sua organização é composta por um conjunto de células de execução, conforme pode ser observado na figura 15.

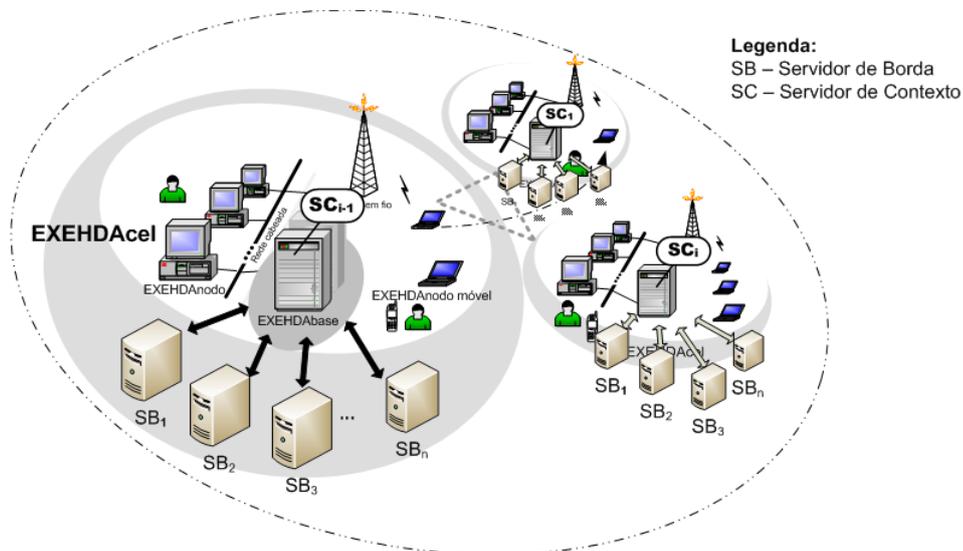


Figura 15: Ambiente ubíquo provido pelo EXEHDA  
Fonte: GUSMÃO, 2013.

Os recursos da infraestrutura física que formam o ambiente ubíquo são mapeados para três abstrações básicas (LOPES et al., 2014):

- EXEHDAcels: indica a área de atuação de uma EXEHDAbase, sendo composta por esta e por EXEHDAodos;
- EXEHDAbase: é o ponto de convergência para os EXEHDAodos, sendo res-

ponsável por todos os serviços básicos do ambiente ubíquo, dentre eles, armazenar as informações coletadas, bem como permitir a manipulação (processamento, visualização, etc.) destas informações. Um componente importante desta abstração é o RIC (Repositório de Informações Contextuais), o qual foi concebido para atender as demandas de armazenamento de dados de contexto;

- EXEHDA nodo: são os dispositivos de processamento disponíveis no ambiente ubíquo, sendo responsáveis pela execução das aplicações. Um subcaso deste tipo de recurso é o EXEHDA nodo móvel, o qual possui elevada portabilidade, tipicamente dotados de interface de rede para operação sem fio.

## 5.2 Armazenamento de triplas: conceitos e ferramentas

As abordagens de modelagens híbridas podem utilizar diversas formas para armazenar os seus dados, tais como: bancos de dados relacionais; baseados em XML; e armazenamento em triplas. Os bancos de dados com armazenamento em triplas foram construídos para o armazenamento e recuperação de informações no formato RDF (*Resource Description Framework*). O formato RDF é indicado para situações em que a informação precisa ser processada por máquinas e não somente apresentada, pois fornece uma interface que permite que informações sejam trocadas entre aplicações sem perda de consistência (OTONI, 2013).

A seguir serão apresentadas algumas das ferramentas que podem ser utilizadas para o armazenamento de dados no formato de triplas, ao final do capítulo é apresentada uma análise entre as ferramentas.

### 4store

4store (4STORE, 2014) é uma base de dados e um motor de consultas que trata dados no formato de triplas. Não providencia muitas funções, referindo-se apenas ao armazenamento de dados em triplas, consultas SPARQL (*SPARQL Protocol and RDF Query Language*), funções de importação de dados e de cópias de segurança/restauração. Os autores destacam como pontos fortes da ferramenta a escalabilidade, o desempenho e a segurança.

### AllegroGraph

AllegroGraph (ALLEGROGRAPH, 2014) é uma estrutura de banco de dados e aplicação de alto desempenho para o armazenamento e consulta de dados no formato de triplas. Ele pode ser implantado como um servidor de banco de dados independente e oferece interfaces para acesso remoto, onde a comunicação entre os processos de servidor e cliente é realizada através da web. Possui a possibilidade de processamento utilizando SPARQL, RDFS (*Resource Description Framework Schema*) e

Prolog a partir de aplicações clientes.

### **Bigdata**

Bigdata (BIGDATA, 2014) é um banco de dados gráfico de alto desempenho que suporta o modelo de triplas, fornece uma maneira de descrever, trocar, e consultar dados utilizando consultas SPARQL. Bigdata pode ser utilizado no formato de triplas ou quádruplas, e possui suporte a inferências.

### **Jena**

Jena (JENA, 2014) é um *framework* que fornece uma abstração e interface para manipulação de grafos RDF representados na memória principal e apoiados pelo mecanismo de banco de dados. A persistência de grafos RDF é alcançada utilizando um banco de dados relacional. Jena suporta um número significativo de sistemas de banco de dados como MySQL, Postgres, Oracle e BerkeleyDB. Os dados armazenados podem ser recuperados através de consultas SPARQL.

### **Mulgara**

Mulgara (MULGARA, 2014) é uma ferramenta para armazenamento de dados no formato de triplas, sendo escrita em Java. Fornece uma API de conexão que pode ser usada para se conectar aos dados armazenados. Possui suporte para consultas SPARQL através de conexões, sendo que estas podem ser realizadas através de uma interface de linha de comando.

### **Ontotext GraphDB**

GraphDB (antigamente conhecido como OWLIM) (GRAPHDB, 2014) é uma ferramenta para armazenamento de dados no formato de triplas. Possui a possibilidade de realização de inferência semântica, permitindo aos usuários criar novos fatos semânticos de dados existentes. Tanto as consultas como as inferências realizadas sobre os dados armazenados podem ser realizadas em tempo de execução.

### **Stardog**

Stardog (STARDOG, 2014) é um banco de dados gráfico semântico, implementado em Java. Ele fornece suporte para RDF e OWL, fornecendo capacidades de raciocínio, utilizando SPARQL como linguagem de consulta. Possui disponibilidade de acesso aos dados utilizando a web, e *plugins* que possibilitam a utilização de outras ferramentas como Jena e Sesame

## Sesame

Sesame (SESAME, 2014) é um *framework* Java de código aberto para processamento de dados no formato de triplas. Isso inclui a análise, armazenamento, inferência e consulta, sobre esses dados. Ele oferece uma API com facilidade de utilização, que pode ser conectada a outras soluções de armazenamento de triplas e permite a conexão com terminais SPARQL.

## Virtuoso

Virtuoso (VIRTUOSO, 2014) é um *middleware* e sistema gerenciador de banco de dados híbrido que combina a funcionalidade de um banco de dados relacional, objeto relacional, XML, RDF. O software é projetado para tirar vantagem do suporte de segmentação do sistema operacional e múltiplos processadores. O armazenamento de dados é realizado utilizando uma quadrupla onde além de armazenar a tripla básica (sujeito, predicado e objeto), também é armazenado o grafo relacionado. Com isso o Virtuoso consegue trabalhar com múltiplos grafos de forma simultânea. Pode ser utilizado como um servidor de aplicação para serviços Web e oferece uma interface Web para consultas SPARQL. É distribuído sobre duas licenças, sendo uma comercial, e a outra código de aberto com o nome OpenLink Virtuoso.

## Análise das Ferramentas

Na tabela 4 é realizada uma comparação entre as ferramentas apresentadas, onde são destacadas algumas características sendo elas: modelo da base de dados; linguagem em que foi desenvolvida a ferramenta; APIs disponíveis; método de consulta; mecanismos para manter a integridade do modelo; licença de utilização; e a última versão e data de lançamento.

A partir da comparação realizada na tabela 4 e das funcionalidades apresentadas por cada uma delas, destaca-se a ferramenta Virtuoso por ser um sistema híbrido trabalhando tanto com formatos relacionais, quanto com triplas, e por ter alcançado o melhor desempenho no estudo realizado em (MAHARAJAN, 2012), destacando-se pela escalabilidade demonstrada. Além disso, por ter sido considerada como sendo uma das três ferramentas notáveis a partir de 2014 (SEMANTICMEDIAWIKI, 2014).

## 5.3 Tecnologias Utilizadas

Esta seção é dedicada para apresentar com maiores detalhes as ferramentas escolhidas para o desenvolvimento do trabalho.

Tabela 4: Comparação entre as ferramentas para armazenamento de triplas

Ferramenta	Modelo da Base de Dados	Linguagem de desenvolvimento	API	Método de Consulta	Integridade do Modelo	Licença	Última Versão
4store	RDF	C	PHP, Ruby, Python, Java, SPARQL	SPARQL	Informação Não Disponível	GPL v3	Versão 1.1.5 Julho de 2012
AllegroGraph	RDF	Common Lisp	Java, Sesame, Jena, Python, Ruby, Perl, C#, Lisp, Scala, REST, SPARQL	SPARQL REST JavaScript RDFS++ Prolog	ACID	Proprietária Eclipse Public License v1	Versão 4.14.1 Setembro de 2014
Bigdata	RDF	Java	Java, Sesame, Blueprints, Gremlin, SPARQL, REST	SPARQL	MVCC ACID	GPLv2 Licença Comercial	Versão 1.4 Novembro de 2014
Jena	Relacional RDF	Java	Java, SPARQL, SPARQL/Update	SPARQL	Informação Não Disponível	Apache License 2.0	Versão 2.12.1 Outubro de 2014
Mulgara	RDF	Java	JTA, SPARQL Jena, Sesame	SPARQL	ACID	Open Software License	Versão 2.1.13 Janeiro de 2012
Ontotext GraphDB	RDF	Java	Java, Jena, Sesame, SPARQL	SPARQL	Informação Não Disponível	GNU LGPL Licença Comercial	Versão 6.1 Dezembro de 2014
Stardog	RDF	Java	Java, Sesame, Jena, SNARL, HTTP/REST, Python, Ruby, Node.js, C#, Spring, SPARQL	SPARQL	ACID	Proprietária	Versão 2.2.4 Dezembro de 2014
Sesame	RDF	Java	Java, HTTP/REST, SPARQL	SPARQL SeRQL	ACID	BSD	Versão 2.7.14 Novembro de 2014
Virtuoso	RDF Relacional Orientado à coluna Objeto relacional Não relacional Escrita Orientada a Documentos	C C++ C#	SPARQL, XMLA, ODBC, JDBC, ADO.NET, OLE DB, Jena, Sesame, Virtuoso PL/SQL, Java, Python, Perl, PHP, HTTP/REST	SQL SPARQL XQuery JSON XPath RDFS++ SOAP	MVCC ACID	GPLv2 Proprietária	Versão 7.1.0 Fevereiro de 2014

### 5.3.1 PostgreSQL

O PostgreSQL (POSTGRESQL, 2015) é um sistema gerenciador de banco de dados objeto-relacional de código aberto coordenado pelo PostgreSQL Global Development Group. Dentre as suas principais características pode se destacar:

- compatibilidade com a maioria dos sistemas operacionais, incluindo Mac OS X, Windows, GNU/Linux, entre outras variantes de Unix;
- compatível com as propriedades de atomicidade, consistência, isolamento e durabilidade;
- possui suporte para a utilização de chaves estrangeiras, joins, views, triggers e procedimentos armazenados (em várias linguagens). Também inclui a maioria

dos tipos de dados SQL: 20081, incluindo INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, e TIMESTAMP;

- Compatibilidade com várias linguagens de programação, entre elas, Java, PHP, Python, Ruby, e C/C++;
- Suporte à utilização de SSL para criptografia das suas conexões.

Além das funcionalidades citadas, destaca-se pela segurança do sistema e por ser considerada uma ferramenta robusta e confiável, a capacidade de gerenciar bases de dados de grande porte e permitir um elevado número de usuários concorrentes (SCHERER; JACOBSEN; SANTOS, 2008).

### 5.3.2 MongoDB

O MongoDB (MONGODB, 2015) é um sistema gerenciador de banco de dados não relacional orientado a documentos, criado em 2007, o qual foi desenvolvido na linguagem C++. Os documentos podem ser considerados análogos aos registros em um banco de dados relacional, e as operações de inserir, atualizar e excluir podem ser executadas em uma coleção. Uma coleção pode armazenar vários documentos, funcionando de forma similar à uma tabela do modelo relacional. Uma coleção pode armazenar documentos de diferentes estruturas devido ao fato do MongoDB ser um banco de dados sem esquema (FILHO, 2015).

Na concepção desse modelo, não foram inseridas funções presentes no modelo relacional, como por exemplo: SELECT \* FROM, JOINS, dentre outras. Isso se deve pela simples justificativa de que cada função destas requer para sua execução certo processamento, e um dos princípios do MongoDB é o desempenho, o que reforça a escalabilidade da solução.

### 5.3.3 Virtuoso

A ferramenta Virtuoso foi escolhida com base na análise apresentada na seção 5.2, para ser utilizada como sistema gerenciador de banco de dados no formato de triplas, o qual é empregado na EXEHDA-HM. O Virtuoso disponibiliza uma interface web intitulada de *CONDUCTOR*, a qual permite a visualização gráfica de todas as funcionalidades da plataforma, e disponibiliza entre outras funcionalidades, uma interface para realização de consultas SPARQL e a possibilidade de criação de novos grafos a partir de arquivos RDF ou OWL.

### 5.3.4 SPARQL

SPARQL é uma linguagem e um protocolo de consulta para RDF, sendo escolhida para ser empregada pela EXEHDA-HM para acesso as informações presentes

no modelo de triplas. SPARQL é um padrão recomendado pelo W3C<sup>1</sup> (*World Wide Web Consortium*), desde janeiro de 2008, que permite consultar documentos RDF remotamente através da sua linguagem de consulta que é sintaticamente semelhante ao SQL (*Structured Query Language*). SPARQL permite quatro formas diferentes de consultas:

- ASK: retorna um valor booleano (TRUE ou FALSE), de acordo com a existência ou não do padrão definido na base consultada;
- CONSTRUCT: forma de consulta que tem como retorno um novo grafo, de acordo com uma condição definida;
- DESCRIBE: retorna um grafo RDF que descreve o recurso(s) procurado(s). Pode ser utilizada nos casos onde não se tem conhecimento prévio da estrutura dos dados ou vocabulários utilizados na descrição destes. O retorno é composto por todas as triplas, onde o objeto de consulta esteja envolvido, seja este um sujeito, predicado ou objeto;
- SELECT: Retorna todas ou um subconjunto das variáveis que atendem ao padrão definido na consulta, semelhante com o SELECT do SQL.

Além das quatro formas de consulta, utilizando a linguagem SPARQL é possível inserir (INSERT) novos dados na ontologia manipulada, aumentando a sua representação, como também é possível excluir (DELETE) elementos presentes na ontologia caso seja necessário.

### 5.3.5 Jena

Para manipulação das informações contextuais presentes no modelo de triplas foi escolhido o *framework* Jena o qual foi desenvolvido na linguagem Java, e fornece uma abstração e interface para manipulação de ontologias. Foi desenvolvido por pesquisadores do laboratório de pesquisa da HP, o HP Labs, e incubado pela Apache em 2010. O *framework* fornece classes e interfaces para criação e manipulação de ontologias baseadas em RDF e OWL, e possui um motor de inferência para OWL e RDFS. Algumas das capacidades do Jena são:

- API para manipulação de arquivos RDF e OWL;
- ler e escrever em RDF/XML, N3, entre outros;
- armazenamento em memória, arquivo ou banco de dados;

---

<sup>1</sup>É um consórcio internacional formado por organizações, uma equipe em tempo integral e o público, que trabalha para desenvolver padrões para a Web. <http://www.w3.org/>

- motor de consulta SPARQL;
- interface de comunicação através do protocolo HTTP (*Hypertext Transfer Protocol*).

## 5.4 Considerações Finais

Este capítulo apresentou uma revisão do escopo do trabalho, onde primeiramente foi apresentado o *middleware* EXEHDA, descrevendo suas principais características. Na sequência, foi introduzido um estudo sobre o armazenamento de triplas, apresentando algumas das principais ferramentas utilizadas para o armazenamento neste formato, sendo realizada uma análise destas ferramentas. Posteriormente, foram apresentadas as tecnologias utilizadas para o desenvolvimento deste trabalho: PostgreSQL, sistema gerenciador escolhido para ser utilizado no modelo relacional de armazenamento; MongoDB, utilizado para gerenciar o modelo não relacional orientado a documentos; Virtuoso, sistema gerenciador do banco de dados em formato de triplas utilizado para o armazenamento das ontologias; SPARQL, linguagem de consulta para o formato de triplas; e Jena, *framework* utilizado para manipulação de ontologias, sendo utilizado para realizar a comunicação com o repositório de triplas.

## 6 EXEHDA-HM: ARQUITETURA E PRINCIPAIS FUNCIONALIDADES

Conforme apresentado na seção 3.6 os modelos baseados em ontologias podem fornecer uma representação completa do conhecimento e a possibilidade de raciocínio sobre informações de contexto. No entanto, ontologias não são concebidas para apoiar a captura e processamento de informações em ambientes dinâmicos, mas estaticamente para representar o conhecimento de um domínio (BETTINI et al., 2010).

Com isso, torna-se necessária a utilização de outras abordagens, como é o caso dos os modelos relacionais, que proporcionam um estilo de gestão de banco de dados de contexto e oferecem interfaces para consultar as informações de contexto. Facilitando algumas questões críticas no projeto de sistemas de gerenciamento de contexto, como o armazenamento de contexto, divulgação e atualização de forma instantânea e replicação das informações contextuais em diferentes domínios (ROUSSAKI et al., 2006).

Com a utilização de modelos híbridos de contexto novos desafios se mostram presentes, dentre os quais destacam-se: (i) o gerenciamento das informações contextuais; (ii) o armazenamento em diferentes modelos proporcionando a persistência das informações contextuais; e (iii) o processamento destas informações, já que com o emprego de modelos híbridos as aplicações podem ter os contextos armazenados em diferentes modelos e muitas vezes é importante poder combinar estas informações para melhorar a identificação das situações de interesse.

Pensando nos desafios de utilização das abordagens híbridas, o presente trabalho propõe a abordagem EXEHDA-HM, a qual possibilita o emprego de modelos híbridos de contexto, onde é concebido um repositório híbrido para armazenamento das informações contextuais, provendo as facilidades particulares de cada um dos diferentes modelos e uma estratégia para processamento dos contextos adquiridos com base em regras composicionais. As regras composicionais possuem como diferencial a capacidade de combinar em uma mesma regra as informações presentes nos diferentes modelos de armazenamento, onde para isso são utilizadas *tags* de marcação na regra criada, sendo estas *tags* substituídas pelas informações que serão buscadas

no modelo desejado.

A arquitetura da abordagem EXEHDA-HM é apresentada na figura 16, onde pode ser visualizada, na parte superior da figura, a camada de processamento que será responsável pelo tratamento das regras composicionais e o repositório desenvolvido intitulado de RHIC (Repositório Híbrido de Informações Contextuais), o qual é dividido em duas camadas, a primeira sendo a camada de serviço, que é responsável por disponibilizar métodos que facilitem o acesso às informações presentes no repositório, e a segunda a camada de armazenamento, que possui como finalidade prover formas diferenciadas de armazenamento, provendo o suporte para armazenamento de informações sintáticas e semânticas. Conforme pode ser notado o RHIC possui três modelos para dados distintos sendo eles: relacional; de triplas; e não relacional. Destaca-se que cada modelo possui vantagens em sua utilização bem como características distintas que justificam a sua escolha (KOTENKO et al., 2013), (ROSA et al., 2015).

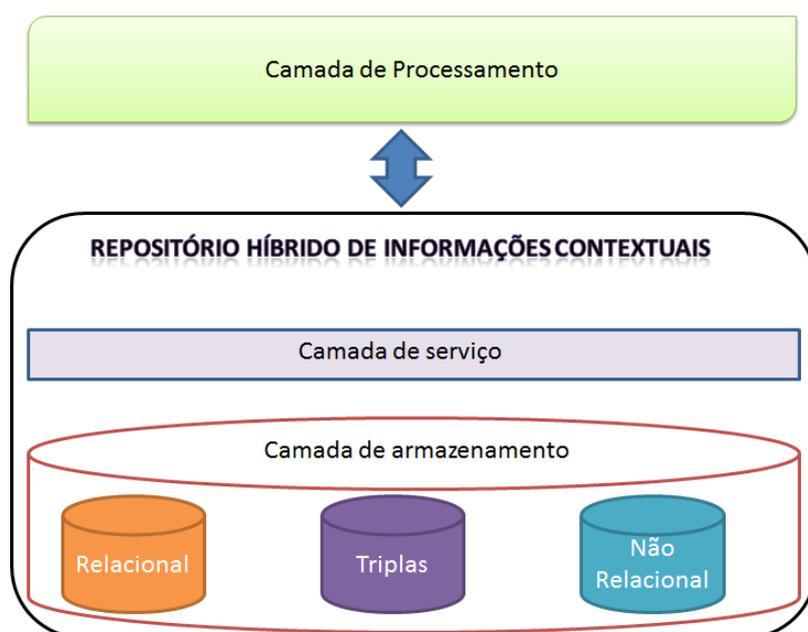


Figura 16: Arquitetura da EXEHDA-HM

A abordagem proposta foi integrada ao *middleware* EXEHDA, de forma a se beneficiar do suporte à aquisição, processamento e armazenamento de informações contextuais, assim como da arquitetura distribuída do *middleware*, características oportunas às funcionalidades da EXEHDA-HM. Para realizar esta integração modificações se fizeram necessárias na arquitetura de software do EXEHDA, as principais foram realizadas no Subsistema de Adaptação e Reconhecimento de Contexto, sendo garantida a interface desse subsistema com os outros existentes no *middleware*.

Com a integração da EXEHDA-HM ao *middleware* EXEHDA, o mesmo passa a estar apto a trabalhar com modelos híbridos de contexto, e com a inclusão do RHIC

no lugar do antigo repositório utilizado, o *middleware* passa a possuir a capacidade de tratar com diferentes formatos de informações contextuais, podendo manipular tanto informações semânticas que agregam significado à representação do contexto quanto sintáticas que representa o contexto sem associar semântica.

A EXEHDA-HM foi incorporada ao Servidor de Contexto do EXEHDA, o qual atua no armazenamento e processamento das informações contextuais, e sua arquitetura é organizada em seis módulos autônomos, onde cada módulo é responsável por uma etapa para obtenção da ciência de contexto. A arquitetura do Servidor de Contexto é apresentada na figura 17, onde pode ser observada a presença do RHIC incorporado a arquitetura do servidor. Os módulos presentes no Servidor de Contexto são (GUSMÃO, 2013):

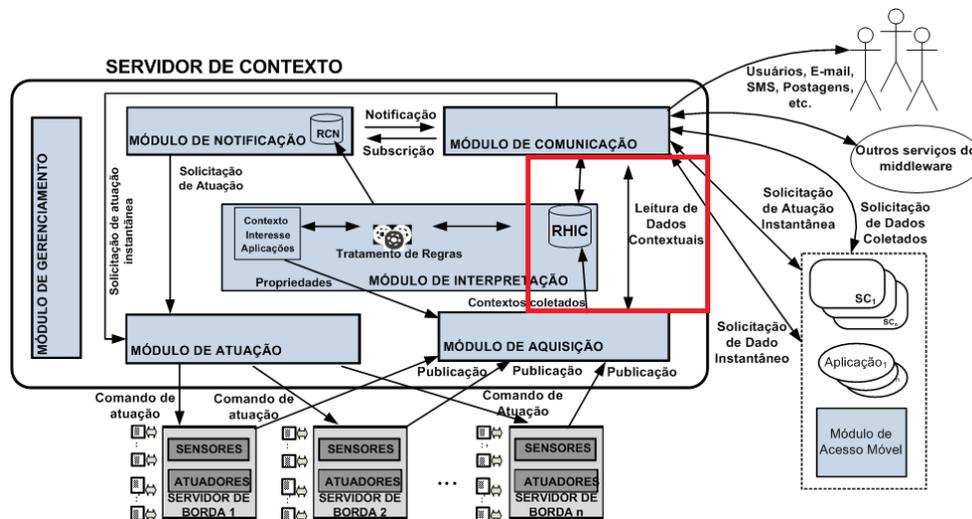


Figura 17: Arquitetura do servidor de contexto  
Fonte: GUSMÃO, 2013.

- módulo de aquisição: responsável por prover suporte à captura das informações contextuais, coletadas pelos Servidores de Borda considerando sensores lógicos (interfaces de software) e/ou hardware;
- módulo de interpretação: tem como principal função realizar tarefas de manipulação e dedução das informações contextuais, utilizando para isto informações dos contextos de interesse das aplicações. O RHIC foi inserido neste módulo para ser responsável pelo gerenciamento das informações contextuais;
- módulo de notificação: responsável por notificar o resultado do processamento contextual realizado pelo Módulo de Interpretação;
- módulo de comunicação: empregado por Servidores de Contexto remotos e/ou aplicações quando da solicitação de dados contextuais e/ou o disparo de atua-

dores. O mesmo provê a disseminação de informações para outros serviços do *middleware*, bem como o envio de mensagens aos usuários;

- módulo de atuação: responsável pelo controle (ativação, desativação e configuração) dos atuadores, após ser notificado pelos outros módulos do Servidor de Contexto;
- módulo de gerenciamento: tem como objetivo permitir ao usuário um gerenciamento confortável das configurações do Servidor de Contexto, provendo facilidades para que sejam especificados os diferentes aspectos dos sensores e atuadores.

Dentre os benefícios decorrentes do desenvolvimento da abordagem proposta, pretende-se facilitar o emprego dos contextos adquiridos, onde as aplicações não precisam se envolver com a interoperação entre as formas de modelagem que estão sendo utilizadas no processamento dos vários contextos de seu interesse. Ainda, com o emprego da estratégia composicional de processamento as aplicações que utilizem o *middleware* EXEHDA tornam-se aptas a utilizarem de forma combinada as informações presentes nos diferentes modelos de armazenamento, aumentando a flexibilidade na utilização dos contextos adquiridos, melhorando assim a identificação das situações de interesse, conseqüentemente, resultando em uma melhora no serviço oferecido.

A seguir com maiores detalhes são apresentados os componentes da abordagem EXEHDA-HM onde, primeiramente, é apresentada a camada de serviço proposta para o RHIC, após, a camada de armazenamento e por último, a camada de processamento, a qual utiliza uma estratégia de processamento composicional.

## 6.1 EXEHDA-HM: Camada de Serviço

A camada de serviço desenvolvida para o RHIC é responsável pela disponibilização de métodos que permitem o acesso às informações contextuais armazenadas na camada de armazenamento, como também pela realização da tarefa de consistência dos dados considerando os três modelos utilizados para realizar a persistência das informações presentes na camada de armazenamento.

As consultas no repositório de triplas são realizadas com base na linguagem de consulta SPARQL, podendo ser executadas diretamente pela interface web, onde podem ser visualizadas as informações presentes no repositório, ou utilizando o mecanismo de gerenciamento das informações contextuais, o qual disponibiliza o acesso às informações de forma que as aplicações possam realizar as consultas em nível de programação utilizando o método de consultas disponibilizado pelo mecanismo na linguagem Java. Para manipulação das informações contextuais presentes no modelo

de triplas foi utilizado o *framework* Jena, o qual permite a manipulação em memória de ontologias, permitindo realizar consultas SPARQL, realização de inferências e as tarefas de edição e inserção de novas informações ao modelo armazenado no repositório.

O processo de inferência das ontologias é realizado utilizando o *reasoner* Pellet<sup>1</sup> (SIRIN et al., 2007), o qual é integrado juntamente com o *framework* Jena facilitando este processo. As ontologias são processadas antes de serem armazenadas no repositório de triplas, desta forma, no momento em que as aplicações desejarem consultar as informações presentes no modelo ontológico, o processo de inferência já estará realizado, podendo resultar em uma melhora do tempo de resposta do sistema, já que o processo de inferência pode se tornar lento dependendo do tamanho da ontologia a ser processada.

O mecanismo de gerenciamento disponibiliza ainda métodos para acesso aos modelos relacional e não relacional, sendo disponibilizados métodos que facilitam a manipulação das informações presentes nos modelos, como também a inserção e remoção de informações desejadas.

Os métodos disponibilizados para tratamento das informações presentes no RHIC são apresentados na figura 18, onde pode ser visualizado que o mecanismo de gerenciamento possui três subclasses, sendo uma para cada modelo de armazenamento presente no RHIC.

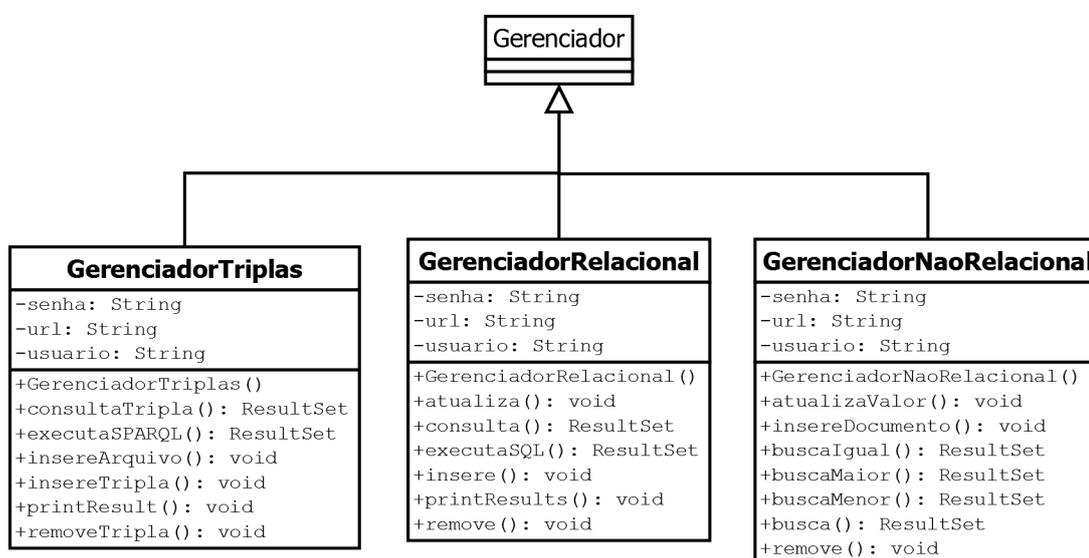


Figura 18: Diagrama de classes do mecanismo de gerenciamento

O mecanismo de gerenciamento para o modelo de triplas disponibiliza os seguintes métodos:

<sup>1</sup> é um *reasoner* desenvolvido em Java e baseado em OWL 2, podendo ser utilizado em conjunto com as bibliotecas da API Jena e OWL, como também pode ser baixado e ser incluído em outras aplicações.

- “consultaTripla”, que realiza consultas SPARQL na ontologia desejada, possuindo como facilidades a composição da consulta SPARQL internamente ao método;
- “executaSPARQ”, responsável por executar algum comando SPARQL na base desejada, sendo válido todos os comandos da linguagem SPARQL;
- “insereArquivo”, que dá a opção de carregar uma nova ontologia para o modelo de triplas do RHIC, diretamente do arquivo OWL ou RDF;
- “insereTripla”, o qual possui como finalidade a inserção de novas informações no formato de triplas na ontologia desejada;
- “printResult”, método disponibilizado para impressão na tela dos resultados de triplas retornados de uma consulta realizada;
- “removeTripla”, realiza a remoção de triplas com base no critério definido da ontologia desejada.

Destaca-se que o método “executaSPARQL” recebe como parâmetro uma string que contém o comando SPARQL que deve ser executado, enquanto os demais métodos utilizam parâmetros que separam a consulta de forma a facilitar a composição da mesma, sendo passadas somente as informações necessárias como variáveis de retorno e possíveis filtros que serão aplicados.

O gerenciador relacional desenvolvido, disponibiliza os seguintes métodos:

- “atualiza”, oferece a possibilidade de atualizar um valor de um determinado campo de uma tabela;
- “consulta”, realiza uma consulta SQL na base desejada;
- “executaSQL”, executa um comando SQL na base desejada, sendo que este pode ser qualquer um dos comandos de SQL;
- “insere”, responsável por inserir novos elementos na tabela desejada;
- “printResults”, imprime na tela o retorno da consulta desejada;
- “remove”, disponibiliza método para remoção de elementos inseridos na tabela com base nas condições desejadas.

Destaca-se que o método “executaSQL” recebe como parâmetro uma string que contém o comando SQL que deve ser executado, enquanto os demais métodos utilizam parâmetros que separam a consulta de forma a facilitar a composição da mesma,

sendo passada somente as informações necessárias como variáveis de retorno e possíveis filtros que serão aplicados.

O mecanismo disponibiliza para o modelo não relacional os seguintes métodos:

- “atualizaValor”, método para atualizar o valor dos campos desejados de um determinado documento presente no RHIC;
- “insereDocumento”, responsável pela inserção de um novo documento no repositório não relacional;
- “buscalgual”, método que realiza a busca por elementos que possuem o valor de um campo igual ao desejado;
- “buscaMaior”, realiza a busca por elementos que possuem o valor de um campo maior que o desejado;
- “buscaMenor”, método que realiza a busca por elementos que possuem o valor de um campo menor que desejado;
- “busca”, realiza a busca desejada no modelo não relacional presente no RHIC, sendo que a busca deve ser composta anteriormente ao método;
- “remove”, realiza a remoção dos elementos desejados do documento armazenado.

## 6.2 EXEHDA-HM: Camada de Armazenamento

Como pode ser notado na figura 16, o RHIC possui três repositórios de armazenamento, são combinados os modelos: relacional, modelo que era utilizado pelo EXEHDA; de triplas, utilizado para armazenar as ontologias; e não relacional, cujo uso tem como motivação central o suporte, quando necessário, a elevados volumes de dados contextuais.

O modelo relacional, como citado anteriormente, é o modelo que é utilizado pelo RIC (GUSMÃO, 2013) no EXEHDA para armazenar as informações contextuais. A sua utilização é mantida, pois se comporta de forma satisfatória para diversas situações, por exemplo, armazenar os sensores que estão sendo monitorados por cada servidor de borda presente no ambiente ubíquo, onde no repositório é mantida a relação do servidor de borda com o sensor, o que facilita o acesso à esta informação como também a sua modificação caso seja necessário. A ferramenta utilizada para ser o sistema gerenciador deste modelo foi o PostgreSQL, a qual era a ferramenta empregada, sendo assim mantida.

O modelo de triplas (sujeito, predicado, objeto) foi empregado para ser utilizado para o armazenamento de ontologias. Com a inserção do modelo de armazenamento de dados no formato de triplas, espera-se prover facilidades similares às experimentadas nos sistemas gerenciadores dos modelos relacionais, onde o acesso às informações presentes nas ontologias é feito de forma automatizada, bem como a manipulação das mesmas, por exemplo, inserção de novas informações, como também de instâncias de classes que foram adicionadas. A ferramenta escolhida para ser o sistema gerenciador deste modelo foi o Virtuoso, o qual se destacou nos estudos realizados, por trabalhos presentes na literatura (MAHARAJAN, 2012) caracterizarem o mesmo como sendo uma ferramenta robusta e com desempenho satisfatório para manipulação dos dados presentes no repositório.

O modelo não relacional foi utilizado tendo como premissa o suporte para o tratamento de grandes volumes de dados. Atualmente, tem surgido um aumento na quantidade e variedade de dados de contexto que devem ser tratados pelo *middleware* EXEHDA, como por exemplo, arquivos de log de aplicações que são monitorados possuindo dentre outras finalidades, melhorar a segurança do ambiente computacional. Para implementação deste modelo foi escolhida a categoria orientada a documentos, a qual armazena estruturas de dados independentes na forma de árvores hierárquicas e autodescritivas, constituídas de mapas, coleções e valores escalares (SHARP et al., 2013).

Uma característica interessante desta categoria de banco de dados não relacional consiste no fato de que os campos vazios são ignorados, o que otimiza o espaço em disco utilizado, sendo a sua utilização recomendada para o tratamento de dados semiestruturados (PRAMOD J SADALAGE, 2013). A ferramenta escolhida para ser o sistema gerenciador do modelo não relacional foi o MongoDB, esta escolha deve-se as características demonstradas pela mesma em diversos trabalhos presentes na literatura (POLITOWSKI; MARAN, 2014) (FILHO, 2015), onde vem se destacando em questões relacionadas ao desempenho no acesso e manipulação de grandes volumes de dados.

Destaca-se que na abordagem proposta fica a cargo da aplicação determinar onde prefere armazenar as informações contextuais, sendo indicada a utilização do modelo relacional para dados que possuem relação entre eles, e assim, aproveitando-se das características do modelo. O modelo de triplas para armazenar as ontologias, de forma a propiciar facilidades para o tratamento das informações presentes nas mesmas. E recomenda-se o armazenamento no modelo não relacional, quando é necessário o tratamento de grande volume de dados e/ou ainda, quando os dados armazenados possuem variedade de formatos, para os quais caso fosse utilizado o modelo relacional, resultaria no mal aproveitamento de espaço.

### 6.3 EXEHDA-HM: Camada de Processamento

A EXEHDA-HM oferece o suporte ao processamento das informações contextuais utilizando uma estratégia baseada regras, consistindo esta da estratégia mais utilizada para esta funcionalidade (PERERA et al., 2014). De forma a prover a funcionalidade da aplicação compor as regras com dados dos diferentes modelos utilizados, as regras definidas devem utilizar *tags* de marcação que comecem com o símbolo “#”, o qual será substituído pela consulta responsável por buscar esta informação no RHIC.

Para dar suporte ao emprego das regras composicionais, foi necessária a inclusão de novas tabelas no banco de dados relacional presente no RHIC, a figura 19 apresenta o diagrama Entidade Relacionamento referente as relações presentes na modelagem das informações necessárias para utilização das regras composicionais.

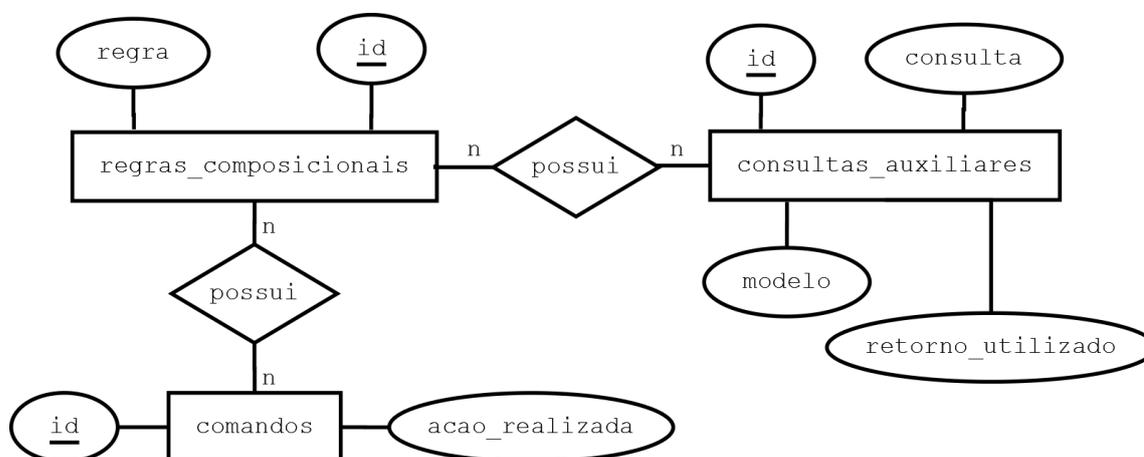


Figura 19: Diagrama entidade relacionamento das regras composicionais

Conforme pode ser visualizada na figura 19, foi criada uma entidade para modelar as regras composicionais, onde está presente o atributo “id”, responsável por identificar as regras, sendo este atributo a chave primária da tabela. As aplicações conseguem identificar qual regra será utilizada buscando pelo respectivo “id” e atributo “regra” referente a regra composicional que será aplicada. Destaca-se que cada regra composicional pode ser composta de uma ou mais consultas auxiliares que podem ser realizadas nos diferentes modelos de armazenamento do RHIC. Além disso, as regras podem possuir um ou mais comandos de ações a serem utilizadas após a identificação da situação de interesse.

As consultas auxiliares são modeladas em uma entidade específica a qual possui como atributo o “id”, o qual é utilizado para identificar as consultas auxiliares, sendo o mesmo utilizado juntamente com a *tag* de marcação na regra composicional, o atributo consulta, que é referente a consulta de fato que deve ser realizada, o modelo de armazenamento que deve ser buscado a informação desejada, e o retorno utilizado, o qual é o valor que será substituído na regra composicional no lugar da *tag* de marcação

referente a consulta realizada. Como retorno das consultas auxiliares e valores a serem substituídos nas regras composicionais, podem ser utilizados os elementos retornados nas consultas (atributos de retorno da consulta), como também o método que verifica se foi encontrado algum elemento na consulta realizada, o qual retorna “true” caso a consulta retorne algum elemento e “false” caso contrário.

Ainda, pode ser visualizada na figura 19 a entidade “comandos”, a qual possui o atributo “id” que é responsável pela identificação dos comandos, sendo o valor deste atributo utilizado na regra composicional para identificar qual o comando que deve ser realizado. Além deste, a entidade possui o atributo “ações”, o qual especifica as ações devem ser realizadas dependendo da situação que foi identificada na execução da regra composicional, sendo que estas ações podem ser de diferentes funcionalidades, por exemplo, o envio de e-mail ou a execução de *scripts*, os quais são determinados pela aplicação.

A figura 20 apresenta a comunicação entre a abordagem EXEHDA-HM e aplicação ciente de contexto, onde pode ser visualizada a requisição realizada pela aplicação e o retorno referente a situação que foi identificada e comando que foi executado. A camada de processamento realiza a execução da regra composicional seguindo um fluxo específico de execução, o qual é apresentado na figura 21.



Figura 20: Comunicação entre EXEHDA-HM e aplicação ciente de contexto

Conforme pode ser visualizado na figura 21, primeiramente é buscada a regra desejada no RHIC, após, são identificadas as consultas auxiliares que devem ser realizadas, tendo como intuito buscar os dados necessários no RHIC para substituírem as *tags* de marcação presentes na regra buscada. Posteriormente, com o retorno

das consultas auxiliares, a regra é composta pelos valores necessários e então é analisada, tendo como retorno o comando desejado a ser executado, onde na regra composicional se encontra apenas o identificador do comando. Com isso é necessário buscar o comando desejado no RHIC, e então, executar o comando retornado.

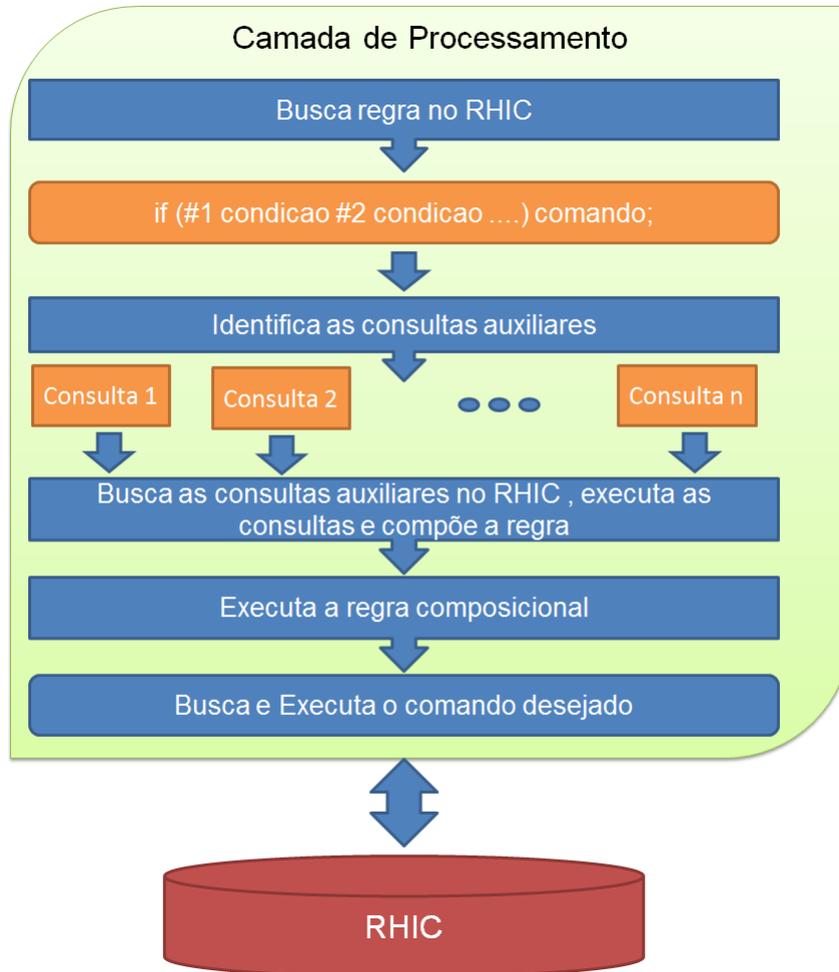


Figura 21: Fluxo de execução da regra composicional

Destaca-se que com a utilização da EXEHDA-HM as regras composicionais podem ser compostas de dados de diferentes modelos, podendo utilizar tanto dados semânticos quanto sintáticos, melhorando a capacidade de identificação de situações de interesse. Caracteriza-se ainda, a facilidade de atualização e manutenção das regras criadas, pois caso mais de uma aplicação utilize as mesmas regras só é necessário atualizar em um determinado campo e as duas aplicações estarão atualizadas. Outra característica de facilidade de utilização é que os comandos que devem ser executados após a identificação da situação não são implementados diretamente nas regras criadas e sim em uma tabela do RHIC, o que facilita a reutilização dos comandos criados como também a manutenção dos mesmos.

## 6.4 Considerações Finais

Neste capítulo foram apresentadas as principais características da abordagem desenvolvida, destacando-se a concepção da EXEHDA-HM, a qual é incorporado ao *middleware* EXEHDA e é concebida com o intuito de oferecer o suporte necessário para a utilização de modelos híbridos de contexto, disponibilizando o armazenamento de informações contextuais em diferentes modelos e uma estratégia de processamento composicional.

Foram apresentadas as duas camadas presentes no RHIC, com as funcionalidades presentes em cada camada, sendo a camada de serviço responsável por disponibilizar métodos que permitem o acesso às informações contextuais, e a camada de armazenamento, a qual realiza a persistência das informações contextuais em um dos três modelos de armazenamento: relacional; não relacional; de triplas.

Destaca-se ainda, a estratégia desenvolvida para ser utilizada para o processamento dos contextos adquiridos, onde foi concebida a utilização de regras composicionais, as quais podem ser compostas de acordo com a necessidade da aplicação, possuindo como diferencial a possibilidade de combinar as informações inseridas nos diferentes modelos de armazenamento presentes no RHIC, e assim, aumentando a flexibilidade na utilização dos contextos adquiridos, podendo melhorar a identificação das situações de interesse com a combinação de informações sintáticas e semânticas.

A tabela 5 apresenta um resumo das principais características da EXEHDA-HM em comparação com os trabalhos relacionados apresentados no capítulo 4. Conforme pode ser visualizado na tabela 5 em relação a etapa de aquisição de contexto, a EXEHDA-HM aproveita-se da capacidade do *middleware* EXEHDA de trabalhar com diferentes tipos de sensores. Na etapa de modelagem de contexto, propõe-se a utilização do modelo relacional juntamente com o modelo baseado em ontologia, seguindo a tendência observada por outros trabalhos com a utilização deste modelo.

Analisando a estratégia de processamento utilizada pelos trabalhos relacionados, destaca-se que apesar de utilizarem diferentes estratégias as mesmas são utilizadas de forma individual, não possibilitando a combinação das informações presentes nos diferentes modelos utilizados. Enquanto que a EXEHDA-HM apresenta uma estratégia de regras composicionais que permite à aplicação combinar os contextos de diferentes modelos, possibilitando assim, a utilização de forma combinada de informações sintáticas e semânticas.

Destaca-se ainda a utilização de diferentes formas de armazenamento da EXEHDA-HM, a qual possibilita o armazenamento no modelo relacional, de triplas, que possui como finalidade armazenar as informações presentes nas ontologias, e não relacional, o qual permite um tratamento eficiente para grandes volumes de dados, e possui como um diferencial a otimização do espaço em disco utilizado para o

Tabela 5: Comparação dos trabalhos de modelagem híbrida de contexto com a EXEHDA-HM

Trabalhos	Aquisição	Modelos	Estratégia Processamento	Armazenamento	Ferramentas
HENRICKSEN; LIVINGSTONE; INDULSKA, 2004	Sensores Físicos Lógicos	Gráfico Ontologia	Regras Ontologia	Base Relacional	Informação Não Disponível
PAGANELLI; BIANCHI; GIULI, 2007	Sensores Físicos	Orientado a Objetos Ontologia	Regras Ontologia	Base Relacional	Jena MySQL
ROCHA; CASANOVA; ENDLER, 2007	Sensores Físicos Lógicos Virtuais	Orientado a Objetos Ontologia	Ontologia	Informação Não Disponível	Jena
EJIGU; SCUTURICI; BRUNIE, 2007	Sensores Físicos Lógicos Virtuais	Ontologia Relacional	Regras Aprendizagem de Máquina Ontologia	Base Relacional	Jena Protégé MySQL
LEE; MEIER, 2009	Sensores Físicos	Orientado a Objetos Ontologia	Ontologia	Informação Não Disponível	Protégé
KOTENKO; POLUBELOVA; SAENKO, 2012	Sensores Físicos Lógicos Virtuais	Linguagem de Marcação Ontologia Relacional	Regras Ontologia	Base Relacional Triplas	Virtuoso Derby
VANATHI, 2013	Sensores Físicos Lógicos Virtuais	Ontologia Relacional Objeto Relacional	Regras Ontologia	Objeto Relacional	PostgreSQL
EXEHDA-HM	Sensores Físicos Lógicos Virtuais	Ontologia Relacional Não Relacional	Regras Composicionais Ontologia	Base Relacional Triplas Não Relacional	PostgreSQL Virtuoso MongoDB Jena

armazenamento de dados semiestruturados. Ainda pode ser visualizadas na tabela 5 as ferramentas utilizadas para a concepção da EXEHDA-HM, onde se optou pelo uso de ferramentas que vêm se destacando na literatura.

## **7 CENÁRIOS DE USO**

Visando avaliar a abordagem EXEHDA-HM, proposta nesta dissertação, a seguir são apresentados dois cenários que se beneficiam da sua utilização, aproveitando-se da possibilidade de modelagem híbrida de contexto e da estratégia de processamento composicional, aumentando a flexibilidade da aplicação e permitindo que seja utilizado em uma mesma regra dados sintáticos e semânticos, desta forma, melhorando a identificação das situações de interesse.

### **7.1 Cenário 1 - Situação de Ataque a Rede**

O presente cenário tem como foco a identificação de situações de interesse na área de segurança da informação, mais especificamente, objetiva identificar tentativas de ataque à rede. Uma das técnicas que pode ser explorada para identificar ataques a rede é a utilização de ontologias, as quais apresentam resultados promissores para este problema (ABDOLI; KAHANI, 2009) (ISAZA et al., 2010) (KHAIRKAR, 2013).

Dentre as diversas ontologias encontradas na literatura optou-se por empregar neste estudo de caso a modelagem utilizada na ontologia definida em (UNDERCOFFER; JOSHI; PINKSTON, 2003), a qual é responsável por mapear as conexões do tráfego da rede e identificar possíveis ataques a mesma. Uma versão resumida da estrutura da ontologia utilizada é apresentada na figura 22, onde podem ser visualizadas as diversas informações que estão presentes nas conexões mapeadas, com destaque para as categorias de ataque que são modeladas como subclasses de consequências.

A identificação de tentativas de ataque a partir do tráfego da rede possui como uma das suas dificuldades o número significativo de falsos positivos e falsos negativos, os quais acontecem quando uma conexão “normal” é identificada como uma categoria de “ataque” e quando um “ataque” é classificado como uma “conexão normal”. Devido a isso, outras informações podem ser utilizadas para melhorar o processo de identificação de ataques, dentre elas, informações de status do dispositivo que está sendo monitorado, tais como, quantidade de memória utilizada, utilização do processador, entre outras. Este tipo de informação sintática possui uma relação que

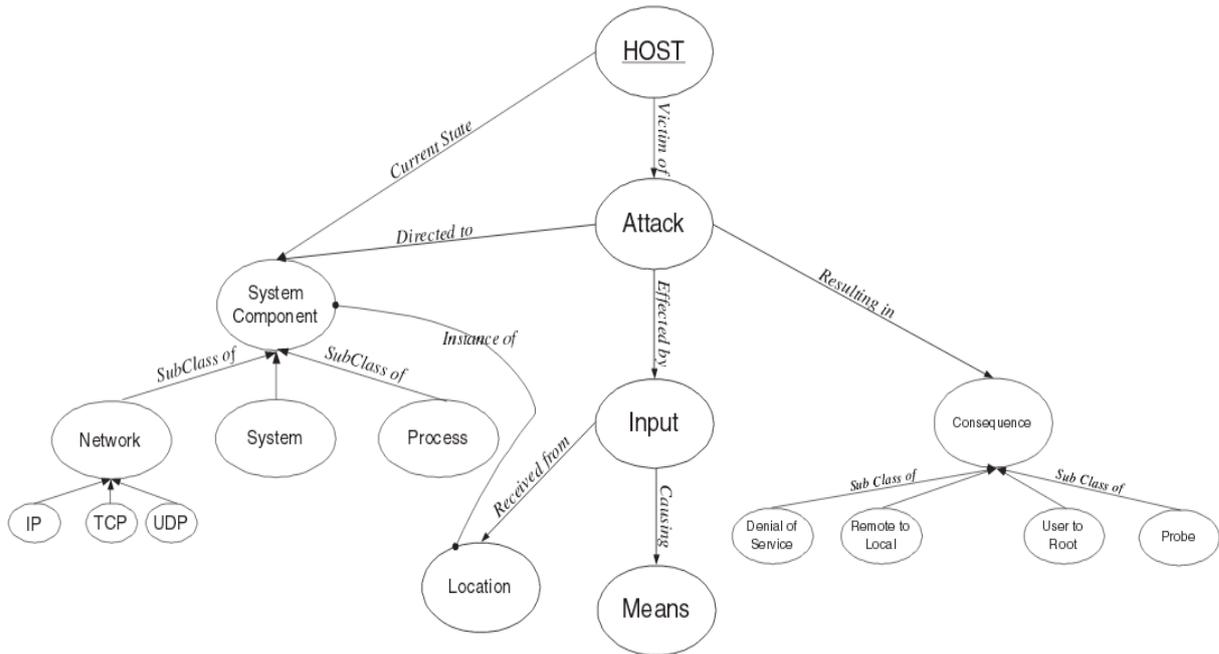


Figura 22: Estrutura resumida da ontologia que modela o tráfego da rede  
 Fonte: UNDERCOFFER; JOSHI; PINKSTON, 2003.

se beneficia do armazenamento no modelo relacional, onde é mantida a relação do dispositivo monitorado e os seus respectivos status.

O emprego deste cenário se beneficia da utilização da EXEHDA-HM, a qual permite o armazenamento da ontologia que detecta ataques ao tráfego da rede no modelo de triplas do RHIC, e das informações referentes aos status do dispositivo no modelo relacional. Além disso, com a estratégia de processamento composicional torna-se possível a combinação destas duas informações, tendo como intuito melhorar a identificação do ataque a rede de um dispositivo computacional.

Para execução deste cenário, foi utilizada como fonte de contexto relacionado ao tráfego da rede a base de dados “KDD Cup 99 Data”, a qual é considerada umas das principais bases utilizadas na avaliação de mecanismos para detecção de tentativas de ataques a servidores de rede (ELEKAR; WAGHMARE; PRIYADARSHI, 2015). O conjunto de treinamento utilizado contém um número significativo de conexões já classificadas em cinco categorias de conexões, sendo uma delas a categoria de conexão “Normal” e quatro categorias de ataques (LEE; STOLFO; MOK, 1999).

Além disso, foi simulada a percentagem de utilização do processador, sendo a mesma categorizada em: Alta; Media; e Baixa. Com esta simulação foi possível testar as diversas combinações com os atributos de identificações de ataques e a categoria de utilização do de processador utilizado em um determinado momento.

Com o intuito de processar os contextos adquiridos foi adicionada a regra composicional a ser executada no RHIC. Onde primeiramente, é realizado um *parser* na regra com o intuito de identificar as informações que devem ser buscadas nos respectivos

repositórios, as quais são representadas pelas *tags* # seguido de um número, sendo o # para identificar as consultas auxiliares que devem ser realizada antes da execução da regra composicional e o número de identificação da consulta.

Com os números das consultas auxiliares as mesmas são buscadas no RHIC, após é realizada a aquisição das informações necessárias para execução da regra com o intuito de identificar a situação desejada. O fluxo de execução da regra híbrida aplicada para a identificação do ataque a rede é apresentado na figura 23, pode-se notar que a regra composicional possui duas *tags* de marcação, onde a *tag* #1 é referente a uma busca no modelo relacional e a #2 a uma busca no modelo de triplas.

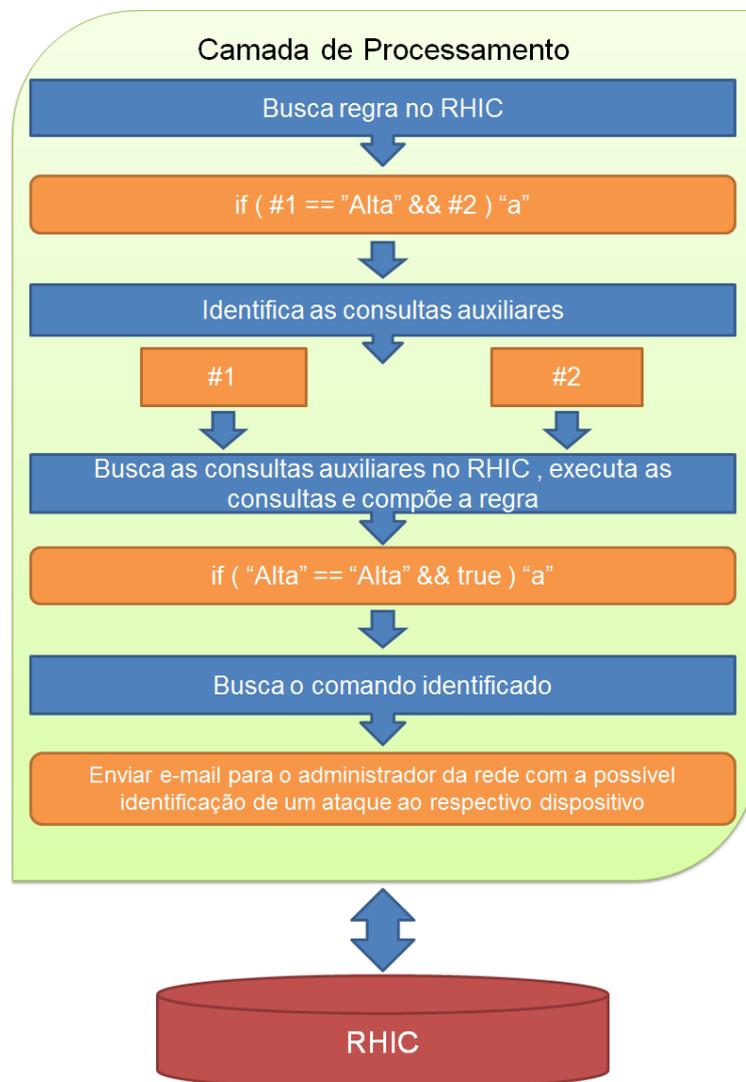


Figura 23: Funcionamento da regra para detecção de ataques a rede

Como pode ser visualizada na figura 23 a regra composicional executada é *if ( #1 == "Alta" && #2 ) "a"*, onde a *tag* de marcação #1 é referente a consulta SQL apresentada na figura 24, a qual busca a categoria que foi classificada a utilização do processador do dispositivo desejado no modelo relacional do RHIC.

```
Select lastvalue, date from items_hosts
where name = 'CPU [Utilizada]' and hostid = 'x'
```

Figura 24: Consulta SQL referente à busca pela a categoria de CPU utilizado

E a *tag #2* é referente a consulta apresentada na figura 25, onde pode ser visualizada a consulta que será realizada no RHIC, mais precisamente no modelo de triplas. A consulta apresentada na figura 25 é responsável por buscar no modelo baseado em triplas, a identificação de um ataque classificado como DoS (*Denial of Service*) para o dispositivo monitorado, este tipo de categoria de ataque costuma gerar uma quantidade significativa de tentativas de aberturas de conexão com um servidor com o intuito de sobrecarregá-lo, gerando assim um aumento na utilização do processador do servidor.

```
SELECT ?individuo ?host ?data
WHERE {
  ?individuo rdf:type trafego:DoS.
  ?individuo trafego:host ?host.
  ?host trafego:id "x"^^xsd:string.
  ?individuo trafego:data ?data.
  FILTER (?data >= data_inic^^xsd:dateTime || ?data <= data_final^^xsd:dateTime).
}
```

Figura 25: Consulta SPARQL referente à busca pelo ataque DoS a um dispositivo

Com o intuito de melhorar a identificação da situação a consulta SPARQL realizada leva em consideração uma janela de tempo, já que a ontologia pode vir a identificar a situação de ataque antes de aumentar a utilização do processador como também o inverso é possível. Como condição na regra composicional é utilizado o método que verifica se foi retornado algum resultado para consulta realizada.

Após as *tags* de marcação serem substituídas pelos valores resultantes de cada consulta, a regra composicional é executada, a identificação da situação de interesse pode ser alcançada, resultando na execução do comando “a”, o qual é referente ao envio de um e-mail para o administrador da rede informando que foi identificado um ataque ao respectivo host.

Com este estudo de caso foi possível demonstrar a utilização da abordagem desenvolvida, onde a aplicação pode se beneficiar da utilização do modelo híbrido de armazenamento e da estratégia de regras composicionais, desta forma, permitindo a combinação da identificação de um ataque à rede com o emprego da ontologia e a quantidade de utilização do processador, e assim, melhorar a identificação da situação

de interesse, mais especificamente, a identificação de um ataque ao dispositivo monitorado.

## 7.2 Cenário 2 - Reforço de Políticas de Senhas

Durante os últimos anos, o Departamento de Segurança da Informação e Comunicações (DSIC) da Presidência da República do Brasil estabeleceu diversos decretos relacionados à Segurança da Informação e Comunicações no âmbito da Administração Pública Federal (APF). Além disso, o DSIC desenvolveu algumas normas complementares com diretrizes para ações a serem desenvolvidas pela APF, sendo que algumas destas normas referenciam a ABNT NBR ISO/IEC (Associação Brasileira de Normas Técnicas Norma Brasileira *International Organization for Standardization/International Electrotechnical Commission*) 27001:2013 e a ABNT NBR ISO/IEC 27002:2013.

Tendo isto em vista, um dos requisitos a ser cumprido pelas instituições Federais, é a elaboração da Política de Segurança da Informação e Comunicações (PO-SIC) (BRASIL, 2000). Sabe-se que um dos desafios relacionados à POSIC é a sua implementação e, mais especificamente, a verificação da sua conformidade. Por exemplo, definir como verificar se os usuários estão em conformidade com a política de senhas (documento geralmente derivado da POSIC).

Desta forma, este estudo de caso explora a ISO 27002 que apresenta um código de prática para a gestão da segurança da informação. Mais especificamente, este estudo de caso demonstra a utilização da abordagem EXEHDA-HM, utilizando o modelo não relacional e o modelo de triplas, onde são combinadas as informações presentes nos dois modelos com o emprego da estratégia de processamento composicional.

Para implementação deste estudo de caso foi utilizada uma ontologia baseada em (ALCÁZAR; FENZ, 2012), a qual é responsável por mapear a estrutura da ISO 27002, incluindo seus controles, ativos, ameaças e vulnerabilidades. Onde, foram adicionadas novas classes de forma a facilitar o acesso às informações de interesse, sendo que optou-se por adicionar como subclasses as diretrizes relacionadas a “password policy”, tais como, “complexity requirements” e “clear text” como pode ser visualizada na figura 26. A ontologia foi armazenada no modelo de triplas presente no RHIC, tendo como intuito, facilitar o acesso às informações presentes na mesma.

Como fonte para a aquisição de contexto, foi explorado o monitoramento dos logs do *Web Application Firewall* modsecurity que foi devidamente configurado para identificar quebras na política de senhas. Um exemplo de log gerado pelo modsecurity é apresentada na figura 27, onde pode ser visualizada as diversas informações que estão presentes no mesmo, como mensagem, a regra utilizada que gerou a detecção pelo modsecurity, URI (*Uniform Resource Identifier*), endereço IP (*Internet Protocol*),

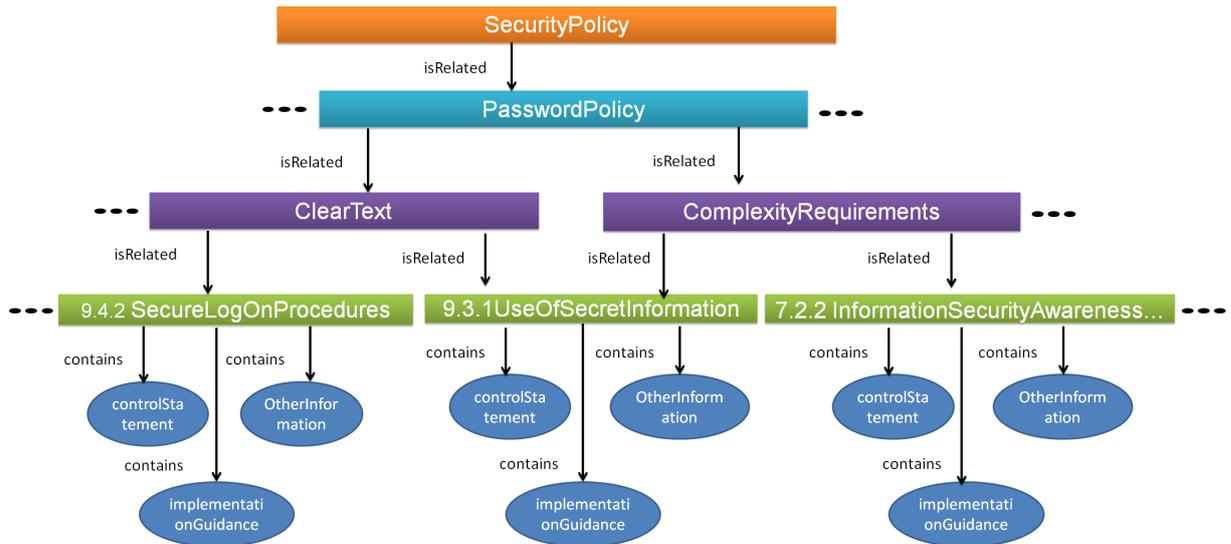


Figura 26: Hierarquia adicionada a ontologia da ISO 27002

entre outras.

```

[Wed Oct 21 19:31:43 2015] [error] [client y.y.y] ModSecurity: Warning. Pattern
match "^(?=[a-zA-Z0-9]?[A-Z])(?=[a-zA-Z0-9]?[a-z])(?=[a-zA-Z0-9]*?[0-9])[a-zA-Z0-
9]{8,}$" at ARGS:pwd. [file "/usr/share/modsecurity-owasp-
crs/optional_rules/modsecurity_crs_16_username_tracking.conf"] [line "30"] [id
"981076"] [msg "Password does meet complexity requirements."] [hostname "x"]
[uri "/site/wp-login.php"] [unique_id "VigEP38AAQEAAHmKC-oAAAAX"]
  
```

Figura 27: Exemplo de log para identificação da situação de política de senha

Após o tratamento do log, é realizada a identificação da situação de interesse, sendo caracterizada pela mensagem “password does meet complexity requirements” e por último a situação identificada é armazenada no modelo não relacional do RHIC, já que as situações possuem um formato semiestruturado, desta forma mostrando-se propícia para ser armazenada no modelo não relacional orientado a documentos utilizado no RHIC, o qual possui como diferenciais o tempo de resposta e uma melhor utilização do espaço em disco que o modelo relacional.

Com a situação já identificada e armazenada no RHIC, pode ser utilizada a estratégia de processamento composicional conforme apresentado na figura 28, com a execução da regra *if(#3 && #4) “b”*. Sendo a *tag* de marcação #3 referente a consulta que busca a situação de política de senhas, mais precisamente a situação de “ComplexityRequirements”, a qual é uma subcategoria de política de senhas, para realizar a busca pela situação no modelo não relacional do RHIC é utilizado o método apresentado na figura 29 que busca as situações que possuem o atributo *subcategory* com o valor ComplexityRequirements.

A *tag* #4 representa a consulta SPARQL que é apresentada na figura 30, a qual

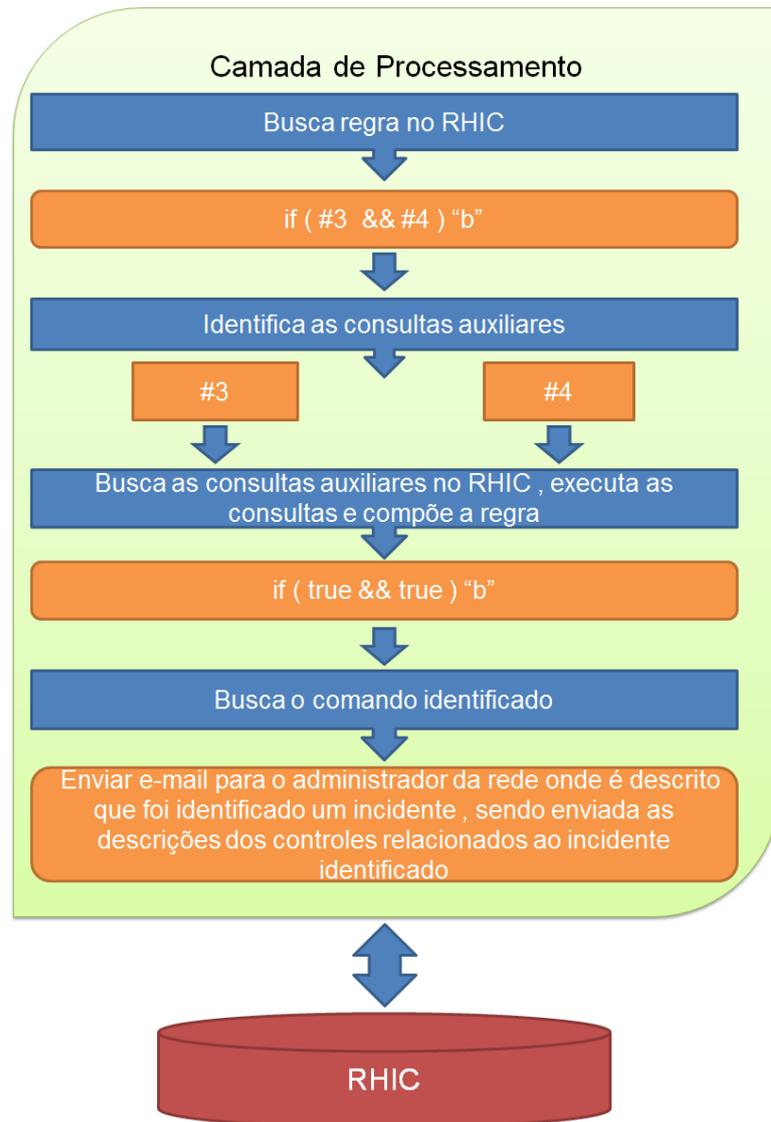


Figura 28: Funcionamento da regra para o reforço de políticas de senha

busca as diretrizes que estão relacionadas a ComplexityRequirements de acordo com a ISO 27002 e as propriedades controlStatement, implementationGuidance, otherInformation onde estão inseridas as informações a respeito de cada diretriz retornada.

Como condição a ser executada na regra composicional, foram utilizados os métodos que verificam se foi retornado algum resultado nas consultas, o qual retorna “true”, caso positivo, e “false”, caso não tenha nenhum retorno. O comando previamente configurado para ser executado, caso seja detectada a situação, é representado por “b”, sendo este referente ao envio de e-mail para os analistas de segurança da informação, onde é descrito que foi identificado um incidente com as descrições dos controles que servirão como sugestões de melhorias a serem empregadas, sendo os analistas responsáveis por avaliar a necessidade de tomada de ações para reforçar a política de senhas.

Com este estudo de caso é possível explorar os benefícios da abordagem

```
find(eq("subcategory", "ComplexityRequirements"))
```

Figura 29: Consulta no modelo não relacional pela situação de interesse

```
SELECT ?guideline ?controlStatement ?implementationGuidance ?explanatoryNotes
WHERE {
    ?guideline rdfs:subClassOf myonto:ComplexityRequirements.
    ?guideline myonto:controlStatement ?controlStatement.
    ?guideline myonto:implementationGuidance ?implementationGuidance.
    ?guideline myonto:otherInformation ?otherInformation.
}
```

Figura 30: Consulta SPARQL referente à busca pela diretriz complexity requirements

EXEHDA-HM, com a utilização do modelo baseado em ontologias, a qual é responsável por mapear a ISO 27002, facilitando a busca pelas informações presentes nesta norma, e ainda, o modelo de armazenamento não relacional, que armazena as situações referentes às políticas de senhas. Além do processamento de informações sintáticas e semânticas de forma combinada, destaca-se a possibilidade de execução do comando desejado, o qual irá enviar informações significativas para os analistas de segurança da instituição, tendo como intuito melhorar o emprego das políticas relacionadas à segurança da informação, mais precisamente referentes às políticas de senhas.

### 7.3 Considerações Finais

Este capítulo apresentou dois cenários que se beneficiam do emprego da abordagem EXEHDA-HM, onde foi demonstrada a utilização do RHIC para o armazenamento das informações contextuais, juntamente com a estratégia de processamento com base em regras composicionais, as quais possuem como diferencial a possibilidade de utilizar em conjunto às informações presentes nos diferentes modelos de armazenamento presentes no RHIC, sendo utilizadas de acordo com a necessidade da aplicação. Com o emprego da abordagem EXEHDA-HM as aplicações possuem uma maior flexibilidade para a criação das regras, e podem melhorar a identificação de situações de interesse, já que agora podem utilizar de forma combinada as informações sintáticas e semânticas armazenadas no RHIC.

## 8 CONCLUSÃO

As principais contribuições alcançadas com o desenvolvimento desta dissertação são: (i) a concepção de uma abordagem com suporte a modelagem híbrida de contexto; (ii) uma camada de armazenamento que provê o armazenamento em três modelos distintos, facilitando o armazenamento de dados sintáticos, semânticos e semiestruturados; (iii) uma camada de serviço que permite o acesso as informações contextuais por parte das aplicações cientes de contexto; e (iv) a concepção de uma estratégia composicional para processamento de contexto, a qual possui como diferencial a possibilidade de combinar dados presentes nos diferentes modelos de armazenamento utilizados.

Esta dissertação apresentou uma revisão sobre sistemas cientes de contexto, mostrando as principais etapas da ciência de contexto, sendo elas aquisição, modelagem, e processamento de contexto. Na etapa de aquisição foram apresentadas as principais fontes utilizadas de forma a adquirir os dados de contexto, caracterizando os pontos positivos e negativos de cada fonte.

A etapa de processamento de contexto possui grande importância para aplicações cientes de contexto, já que é responsável por realizar raciocínio sobre os dados de contexto adquiridos, e assim deduzir informações significativas, que auxiliem as aplicações cientes de contexto. Foram apresentadas as principais técnicas utilizadas, mostrando suas principais características e vantagens em sua utilização, onde destaca-se a utilização da técnica baseada em regras, devido a sua simplicidade, e por ser a técnica mais utilizada nesta etapa.

Na etapa de modelagem de contexto, foram apresentadas as principais abordagens utilizadas, mostrando suas principais características como também as vantagens e desvantagens em sua utilização. Das abordagens apresentadas destacam-se os modelos híbridos, que vêm ganhando espaço combinando dois ou mais modelos de forma a aproveitar os pontos positivos de cada abordagem, tendo como objetivo o aprimoramento da etapa de modelagem de contexto e, com isso, qualificar a utilização das informações contextuais.

Foram apresentados trabalhos que utilizam a modelagem baseada na integração

de modelos, modelos híbridos, combinando os pontos positivos de cada modelo, onde foram destacadas as arquiteturas e características de cada trabalho. Uma característica interessante dos trabalhos analisados foi que todos utilizam o modelo baseado em ontologia combinado com algum outro modelo.

Foi realizado um levantamento sobre ferramentas de armazenamento em formato de triplas, as quais podem ser utilizadas para armazenar as informações contextuais, de forma a manter a persistência dos dados e a facilitar o acesso às informações presentes nos modelos baseado em ontologias. Dentre as ferramentas analisadas ressalta-se o Virtuoso, por ser um gerenciador de dados híbrido, trabalhando tanto com dados no formato de triplas, como também com dados relacionais. Dada esta característica híbrida da ferramenta, ela se sobressai para ser utilizada para realizar o armazenamento dos dados, em cenários que empregam uma modelagem baseada na integração de modelos, já que esta ferramenta pode ser utilizada para manter a persistência e tratar diferentes tipos de dados.

Considerando isto, de forma a capacitar o *middleware* EXEHDA a trabalhar com diferentes abordagens de modelagem de contexto, o presente trabalho propôs uma abordagem composicional para processamento de informações contextuais em modelos híbridos de contexto, intitulada de EXEHDA-HM, a qual provê suporte ao tratamento de informações contextuais tanto sintáticas quanto semânticas. Na proposta, foi criado um novo repositório para armazenamento de informações contextuais, o RHIC, o qual é dividido em duas camadas: de serviço e de armazenamento:

- A camada de serviço é responsável por gerenciar as informações contextuais presentes na camada de armazenamento, o mecanismo de gerenciamento presente nesta camada disponibiliza métodos para consulta, inserção e remoção de dados na camada de armazenamento. São disponibilizados métodos para tratar as informações contextuais nos três modelos de armazenamento presentes no RHIC.
- A camada de armazenamento distingue-se pela utilização de três modelos para o armazenamento de informações contextuais tratadas pelo *middleware* EXEHDA, os modelos utilizados são: relacional, modelo que vinha sendo utilizado pelo *middleware*; não relacional, o qual possui como premissa oferecer o suporte ao tratamento eficiente de grandes volumes de dados; e formato de triplas, sendo este o modelo escolhido para realizar o armazenamento das informações contextuais presentes nas ontologias, oferecendo mecanismos que facilitam o acesso à estas informações.

Além da criação do RHIC, a abordagem EXEHDA-HM possui como diferencial a camada de processamento desenvolvida, onde se utiliza de uma estratégia baseada em

regras composicionais, permitindo combinar nas regras criadas dados de diferentes modelos de armazenamento e, assim, poder utilizar tantos dados sintáticos quanto semânticos. Destaca-se que com a utilização da abordagem composicional aumenta-se a flexibilidade da aplicação, permitindo que sejam utilizados em uma mesma regra dados sintáticos e semânticos, melhorando a identificação das situações de interesse das aplicações.

Foram desenvolvidos testes com o intuito de avaliar a abordagem proposta, mostrando a implementação das regras composicionais para identificação das situações de interesse. O primeiro estudo de caso tem a premissa de identificar situações de ataque à rede, mais precisamente, com a utilização de uma ontologia, a qual possui a capacidade de detectar um ataque, por exemplo, do tipo DoS, e a combinação desta categoria de ataque com a informação da quantidade de processador utilizada em um determinado momento. Este primeiro cenário se beneficiou da utilização do RHIC, o qual proveu o armazenamento da ontologia no modelo baseado em triplas e a quantidade de processador no modelo relacional, e assim utilizando a estratégia de processamento de regras composicionais foi possível a identificação da situação com a combinação das informações presentes nos dois modelos de armazenamento.

O segundo estudo de caso tem como premissa reforçar as políticas de senha, onde para isso, foi utilizada a detecção de situações relacionadas às políticas de senha, analisando os logs do *Web Application Firewall* modsecurity e, ainda, a utilização de uma ontologia que realiza o mapeamento das informações presentes na ISO 27002. Com a utilização desta estratégia de processamento foi possível combinar a identificação de situações de política de senha presente no modelo não relacional e as informações presentes na ISO 27002 referentes à situação identificada, sendo que a ontologia foi armazenada no modelo de triplas.

## 8.1 Trabalhos Futuros

Dentre os aspectos levantados para continuidade do trabalho destaca-se:

- criar interfaces que utilizem técnicas de visualização de dados, com o intuito de facilitar a interpretação das informações contextuais armazenadas como também das situações que foram identificadas;
- avaliar a utilização da ferramenta Virtuoso para gerenciamento do modelo relacional, bem como estudos que avaliem questões relacionadas a desempenho, como por exemplo, tempo de inserção e consulta dos dados, de forma a identificar a forma mais interessante de utilizar a presente ferramenta;
- realizar estudos de ferramentas para processamento de eventos complexos e suas respectivas sintaxes, tendo como intuito avaliar alternativas às regras utili-

zadas que seguem o padrão do “condicional if” da linguagem de programação Java. Com este estudo, espera-se avaliar a possibilidade de facilitar a criação das regras e a utilização das mesmas para detecção de situações de interesse;

- aplicar a abordagem EXEHDA-HM em diferentes cenários de utilização, por exemplo, em ambientes educacionais, os quais costumam utilizar tanto modelos baseado em ontologia, quanto relacional, tornando possível o processamento das informações presentes nos modelos de forma combinada.

## 8.2 Publicações

No desenvolvimento desta dissertação foram efetuadas algumas publicações pertinentes à abordagem EXEHDA-HM, bem como as que foram desenvolvidas no grupo de pesquisa Laboratory of Ubiquitous and Parallel Systems (LUPS) que tiveram participação efetiva deste autor. Primeiramente, destaca-se uma publicação em revista:

- MACHADO, R. S., ALMEIDA, R. B., YAMIN, A. C., PERNAS, A. M. LogA-DM: An Approach of Dynamic Log Analysis. In: Latin America Transactions, IEEE (Revista IEEE America Latina), vol. 13, no. 9, Set. 2015.

Além desta, foram realizadas publicações em anais de eventos, escolas regionais e simpósios:

- MACHADO, R. S., ALMEIDA, R. B., ROSA, D. Y. L., RIPPEL, H. V., YAMIN, A. C., PERNAS, A. M. DLNA-ML: Uma Abordagem de Análise Dinâmica de Log e Tráfego da Rede. In: 13<sup>a</sup> Escola Regional de Redes de Computadores, 2015.
- ALMEIDA, R. B., MACHADO, R. S., ROSA, D. Y. L., RIPPEL, H. V., DONATO, L. M., YAMIN, A. C., PERNAS, A. M. NS<sup>2</sup>A: consciência de situação aplicada a segurança de redes de computadores. In: 13<sup>a</sup> Escola Regional de Redes de Computadores, 2015.
- ROSA, D. Y. L., RAMBO, I. J., MACHADO, R. S., ALMEIDA, R. B., RIPPEL, H. V., Yamin, Adenauer C., PERNAS, A. M. Uma abordagem híbrida para armazenamento de dados de contexto no EXEHDA. In: 13<sup>a</sup> Escola Regional de Redes de Computadores, 2015.
- MACHADO, R. S., ALMEIDA, R. B., YAMIN, A. C., PERNAS, A. M. UM MECANISMO PARA O GERENCIAMENTO DE MODELOS HÍBRIDOS DE CONTEXTO. In: XVII Encontro de Pós-Graduação da UFPel, 2015.

- ALMEIDA, R. B., MACHADO, R. S., ROSA, D. Y. L., DONATO, L. M., YAMIN, A. C., PERNAS, A. M. UMA ARQUITETURA HIERÁRQUICA MULTINÍVEL PARA CONSCIÊNCIA DE SITUAÇÃO EM SEGURANÇA DE AMBIENTES COMPUTACIONAIS. In: XVII Encontro de Pós-Graduação da UFPel, 2015.
- ROSA, D. Y. L., RAMBO, I. J., MACHADO, R. S., ALMEIDA, R. B., RIPPEL, H. V., YAMIN, A. C., PERNAS, A. M. UMA PROPOSTA PARA GERENCIAMENTO HÍBRIDO DE DADOS DE CONTEXTO NO EXEHDA. In: XVII Encontro de Pós-Graduação da UFPel, 2015.
- ALMEIDA, R. B., MACHADO, R. S., ROSA, D. Y. L., DAVET, P. T., DONATO, L. M., YAMIN, A. C., PERNAS, A. M. Segurança na Internet das Coisas: uma abordagem de SIEM customizável e baseada em Consciência de Situação. In: 42º SEMISH - Seminário Integrado de Software e Hardware, 2015.
- MACHADO, R. S., ALMEIDA, R. B., DAVET, P. T., Yamin, Adenauer C., PERNAS, A. M. Uma Proposta de Gerenciamento de Modelo Híbrido de Contexto em Sistemas Distribuídos. In: Escola Regional de Alto Desempenho, 2015, Gramado/RS.
- ALMEIDA, R. B., MACHADO, R. S., ROSA, D. Y. L., DONATO, L. M., Yamin, Adenauer C., PERNAS, A. M. Uma Proposta Distribuída para Detecção de Intrusão, Hierárquica, Multiagente e Consciente de Situação. In: Escola Regional de Alto Desempenho, 2015, Gramado/RS.
- ALMEIDA, R. B., MACHADO, R. S., DONATO, L. M., PERNAS, A. M., YAMIN, A. C. Consciência de Situação Aplicada à Segurança de Ambientes Ubíquos. In: VI Simpósio Brasileiro de Computação Ubíqua e Pervasiva, 2014, Brasília.
- MACHADO, R. S., ALMEIDA, R. B., ROSA, D. Y. L., YAMIN, A. C., PERNAS, A. M. Análise de Log: Uma Abordagem Baseada em Mineração Dinâmica de Dados. In: Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul, 2014, Alegrete, RS.
- ALMEIDA, R. B., MACHADO, R. S., ROSA, D. Y. L., DONATO, L. M., YAMIN, A. C., PERNAS, A. M. Explorando a Consciência de Situação no Gerenciamento de Aspectos de Segurança em Sistemas Distribuídos. In: Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul, 2014, Alegrete, RS.

## REFERÊNCIAS

4STORE. Disponível em: <[4store.org](http://4store.org)>, acesso em de dezembro 2015.

ABDOLI, F.; KAHANI, M. **Ontology-based distributed intrusion detection system**. 14th International CSI Computer Conference.

AHN, S.; 0001, D. K. Proactive Context-Aware Sensor Networks. In: EWSN, 2006. **Anais...** Springer, 2006. p.38–53. (Lecture Notes in Computer Science, v.3868).

AKMAN, V.; SURAV, M. The Use of Situation Theory in Context Modeling. **Computational Intelligence**, [S.l.], v.13, n.3, p.427–438, 1997.

ALCÁZAR, F.; FENZ, S. Mapping ISO 27002 into Security Ontology. **Treball final de grau – Universitat Politècnica de Catalunya. Escola d’Enginyeria de Telecomunicació i Aeroespacial de Castelldefels. Departament de Teoria del Senyal i Comunicacions, 2012 (E. T. de Telecomunicació, especialitat en Telemàtica)**, [S.l.], 2012.

ALLEGROGRAPH. Disponível em: <[franz.com/agraph/allegrograph/](http://franz.com/agraph/allegrograph/)>, acesso em de dezembro 2015.

ALMEIDA, D. R. de; SOUZA BAPTISTA, C. de; ANDRADE, F. G. de. Using Ontologies in Context-Aware Applications. In: DEXA WORKSHOPS, 2006. **Anais...** IEEE Computer Society, 2006. p.349–353.

ASENSIO, E. S. **Context Modeling and Collaborative Context-Aware Services for Pervasive Computing**. 2011. PROGRAMA DE DOCTORADO EN INFORMÁTICA - Centro de Investigación en Métodos de Producción de Software — Universidad Politécnica de Valencia, Valencia, Spain.

BACON, J.; BATES, J.; HALLS, D. Location-oriented multimedia. **IEEE Personal Commun.**, [S.l.], v.4, n.5, p.48–57, 1997.

BARDRAM, J. E. The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications. In: PERVA-

SIVE, 2005. **Anais...** Springer, 2005. p.98–115. (Lecture Notes in Computer Science, v.3468).

BAUER, J. S.; NEWMAN, M. W.; KIENTZ, J. A. Thinking About Context: Design Practices for Information Architecture with Context-Aware Systems. In: CONFERENCE 2014 PROCEEDINGS, 2014. **Anais...** [S.l.: s.n.], 2014. p.398–411.

BELLAVISTA, P.; CORRADI, A.; FANELLI, M.; FOSCHINI, L. A survey of context data distribution for mobile ubiquitous systems. **ACM Comput. Surv.**, New York, NY, USA, v.44, n.4, p.24:1–24:45, Sept. 2012.

BETTINI, C.; BRDICZKA, O.; HENRICKSEN, K.; INDULSKA, J.; NICKLAS, D.; RANGANATHAN, A.; RIBONI, D. A Survey of Context Modelling and Reasoning Techniques. **Pervasive Mob. Comput.**, Amsterdam, The Netherlands, The Netherlands, v.6, n.2, p.161–180, Apr. 2010.

BIBRI, S. E. Context Modeling, Representation, and Reasoning: An Ontological and Hybrid Approach. In: **The Human Face of Ambient Intelligence**. [S.l.]: Atlantis Press, 2015. p.197–257. (Atlantis Ambient and Pervasive Intelligence, v.9).

BIGDATA. Disponível em: <[www.bigdata.com/](http://www.bigdata.com/)>, acesso em de dezembro 2015.

BOLCHINI, C.; CURINO, C. A.; QUINTARELLI, E.; SCHREIBER, F. A.; TANCA, L. A Data-oriented Survey of Context Models. **SIGMOD Rec.**, New York, NY, USA, v.36, n.4, p.19–26, Dec. 2007.

BRASIL. Decreto Lei nº 3505 – Institui a Política de Segurança da Informação nos órgãos e entidades da Administração Pública Federal. **Diário Oficial da União**, Brasília, Junho 2000.

BRÉZILLON, P. Context in problem solving: a survey. **Knowl. Eng. Rev.**, New York, NY, USA, v.14, n.1, p.47–80, May 1999.

BRÉZILLON, P. Task-Realization Models in Contextual Graphs. In: CONTEXT, 2005. **Anais...** Springer, 2005. p.55–68. (Lecture Notes in Computer Science, v.3554).

CARVALHO, A. G. **Interface NoSQL integrada a banco relacional para gerenciamento de dados em nuvem privada**. 2014. Monografia Bacharelado em Engenharia da Computação — Centro Universitário de Brasília Faculdade de Tecnologia e Ciências Sociais Aplicadas.

CHEVERST, K.; MITCHELL, K.; DAVIES, N. Design of an object model for a context sensitive tourist GUIDE. **Computers & Graphics**, [S.l.], v.23, n.6, p.883–891, 1999.

CONAN, D.; ROUVOY, R.; SEINTURIER, L. Scalable Processing of Context Information with COSMOS. In: DAIS, 2007. **Anais...** Springer, 2007. p.210–224. (Lecture Notes in Computer Science, v.4531).

CORRADI, A.; FANELLI, M.; FOSCHINI, L. Implementing a scalable context-aware middleware. In: ISCC, 2009. **Anais...** IEEE, 2009. p.868–874.

COUTAZ, J.; CROWLEY, J. L.; DOBSON, S.; GARLAN, D. Context is Key. **Commun. ACM**, New York, NY, USA, v.48, n.3, p.49–53, Mar. 2005.

DEY, A. K. Understanding and Using Context. **Personal and Ubiquitous Computing**, [S.l.], v.5, p.4–7, 2001.

EJIGU, D. **Context Modeling and Collaborative Context-Aware Services for Pervasive Computing**. 2007. Doctoral School of Computer and Information Sciences (EDIIS) Affiliated Area: Computer Science — National Institute of Applied Sciences, Insa de Lyon.

EJIGU, D.; SCUTURICI, M.; BRUNIE, L. CoCA: A Collaborative Context-Aware Service Platform for Pervasive Computing. In: INFORMATION TECHNOLOGY, 2007. ITNG '07. FOURTH INTERNATIONAL CONFERENCE ON, 2007. **Anais...** [S.l.: s.n.], 2007. p.297–302.

EJIGU, D.; SCUTURICI, M.; BRUNIE, L. Hybrid approach to collaborative context-aware service platform for pervasive computing. **Journal of computers**, [S.l.], p.40, 2008.

ELEKAR, K.; WAGHMARE, M.; PRIYADARSHI, A. Use of rule base data mining algorithm for intrusion detection. In: PERVASIVE COMPUTING (ICPC), 2015 INTERNATIONAL CONFERENCE ON, 2015. **Anais...** [S.l.: s.n.], 2015. p.1–5.

FERNANDES, P. C. C. **UbiFEX: Uma Abordagem para Modelagem de Características de Linha de Produtos de Software Sensíveis ao Contexto**. 2009. Dissertação (Mestrado em Engenharia de Sistemas e Computação) — UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação.

FILHO, M. A. P. M. **SQL X NOSQL: Análise de Desempenho do Uso do MongoDB em Relação ao Uso do PostgreSQL**. 2015. Graduação em Ciência da Computação — Universidade Federal de Pernambuco.

GRAPHDB, O. Disponível em: <<http://www.ontotext.com/products/ontotext-graphdb/>>, acesso em de dezembro 2015.

GRAY, P. D.; SALBER, D. Modelling and Using Sensed Context Information in the Design of Interactive Applications. In: EHCI, 2001. **Anais...** Springer, 2001. p.317–336. (Lecture Notes in Computer Science, v.2254).

GRUBER, T. R. A translation approach to portable ontology specifications. **Knowledge Acquisition**, London, UK, UK, v.5, n.2, p.199–220, June 1993.

GUSMÃO, M. **Uma Arquitetura de Software direcionada à Consciência de Contexto na UbiComp**. 2013. Dissertação de Mestrado em Ciência da Computação — PPGC/UFPel, Pelotas-RS.

HELD, A.; BUCHHOLZ, S.; SCHILL, A. Modeling of Context Information for Pervasive Computing Applications. In: WORLD MULTICONFERENCE ON SYSTEMICS, CYBERNETICS AND INFORMATICS, 6., 2002, Orlando, FL. **Proceedings...** [S.l.: s.n.], 2002.

HENRICKSEN, K.; INDULSKA, J. A Software Engineering Framework for Context-Aware Pervasive Computing. In: IN: PERCOM, IEEE COMPUTER SOCIETY, 2004. **Anais...** IEEE Computer Society, 2004. p.77–86.

HENRICKSEN, K.; INDULSKA, J. Developing Context-Aware Pervasive Computing Applications: Models and Approach. **Pervasive and Mobile Computing**, In, [S.l.], v.2, p.2005, 2005.

HENRICKSEN, K.; INDULSKA, J.; RAKOTONIRAINY, A. Modeling context information in pervasive computing systems. In: **Pervasive Computing**. [S.l.]: Springer Berlin Heidelberg, 2002. p.167–180.

HENRICKSEN, K.; LIVINGSTONE, S.; INDULSKA, J. Towards a hybrid approach to context modelling, reasoning and interoperation. In: FIRST INTERNATIONAL WORKSHOP ON ADVANCED CONTEXT MODELLING, REASONING AND MANAGEMENT, 2004. **Proceedings...** [S.l.: s.n.], 2004.

HERBERT, J.; O'DONOGHUE, J.; CHEN, X. A Context-Sensitive Rule-Based Architecture for a Smart Building Environment. In: FGCN (2), 2008. **Anais...** IEEE Computer Society, 2008. p.437–440.

HOU, C.-H.; HSIAO, H.-C.; KING, C.-T.; LU, C.-N. Context discovery in sensor networks. In: ITRE, 2005. **Anais...** IEEE, 2005. p.2–6.

ISAZA, G. A.; CASTILLO, A. G.; LÓPEZ, M.; CASTILLO, L. F. **Towards Ontology-Based Intelligent Model for Intrusion Detection and Prevention**. Journal of Information Assurance and Security 5.

JACOB, C.; LINNER, D.; STEGLICH, S.; RADUSCH, I. Bio-inspired Context Gathering in Loosely Coupled Computing Environments. In: INTERNATIONAL CONFERENCE ON BIO INSPIRED MODELS OF NETWORK, INFORMATION AND COMPUTING SYSTEMS, 1., 2006, New York, NY, USA. **Proceedings...** ACM, 2006. (BIONETICS '06).

JENA. **A free and open source Java framework for building Semantic Web and Linked Data applications.** Disponível em: <<http://jena.apache.org/>>, acesso em dezembro de 2015.

KAENAMPORNPAN, M.; O'NEILL, E. An integrated context model: Bringing activity to context. **Proc. Workshop on Advanced Context Modelling, Reasoning and Management**, [S.l.], 2004.

KATSIRI, E.; MYCROFT, A. Knowledge Representation and Scalable Abstract Reasoning for Sentient Computing Using First-Order Logic. In: Challenges and Novel Applications for Automated Reasoning Workshop, in CADE-19, Miami, 2003. **Anais...** [S.l.: s.n.], 2003. Conference Paper.

KHAIRKAR, A. D. **Intrusion Detection System based on Ontology for Web Applications.** 2013. Master of Technology, Computer Engineering — Department of Computer Engineering and Information Technology College of Engineering, Pune.

KNAPPMAYER, M.; KIANI, S.; REETZ, E.; BAKER, N.; TONJES, R. Survey of Context Provisioning Middleware. **Communications Surveys Tutorials, IEEE**, [S.l.], v.15, n.3, p.1492–1519, Third 2013.

KOTENKO, I.; POLUBELOVA, O.; CHECHULIN, A.; SAENKO, I. Design and Implementation of a Hybrid Ontological-Relational Data Repository for SIEM Systems. **Future Internet - Special Issue Security of Systems and Software Resiliency**, [S.l.], 2013.

KOTENKO, I.; POLUBELOVA, O.; SAENKO, I. The Ontological Approach for SIEM Data Repository Implementation. In: IEEE INTERNATIONAL CONFERENCE ON GREEN COMPUTING AND COMMUNICATIONS, 2012., 2012, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2012. p.761–766. (GREENCOM '12).

KRUMM, J. **Ubiquitous Computing Fundamentals.** 1st.ed. [S.l.]: Chapman & Hall/CRC, 2009.

LEE, D.; MEIER, R. A Hybrid Approach to Context Modelling in Large-scale Pervasive Computing Environments. In: FOURTH INTERNATIONAL ICST CONFERENCE ON COMMUNICATION SYSTEM SOFTWARE AND MIDDLEWARE, 2009, New York, NY, USA. **Proceedings...** ACM, 2009. p.14:1–14:12. (COMSWARE '09).

LEE, W.; STOLFO, S.; MOK, K. A data mining framework for building intrusion detection models. In: SECURITY AND PRIVACY, 1999. PROCEEDINGS OF THE 1999 IEEE SYMPOSIUM ON, 1999. **Anais...** [S.l.: s.n.], 1999. p.120–132.

LI, F.; SEHIC, S.; DUSTDAR, S. COPAL: An adaptive approach to context provisioning. In: WIMOB, 2010. **Anais...** IEEE, 2010. p.286–293.

LOPES, J.; SOUZA, R.; GEYER, C.; COSTA, C.; BARBOSA, J.; PERNAS, A.; YAMIN, A. A Middleware Architecture for Dynamic Adaptation in Ubiquitous Computing. **Journal of Universal Computer Science**, [S.l.], v.20, n.9, p.1327–1351, sep 2014. [http://www.jucs.org/jucs\\_20\\_9/a\\_middleware\\_architecture\\_for](http://www.jucs.org/jucs_20_9/a_middleware_architecture_for).

MACHADO, A. **Associação do Contexto de Interesse do Usuário às Atividades Clínicas na Arquitetura Clinicspace**. 2010. Dissertação de Mestrado em Ciência da Computação — Programa de Pós-Graduação em Informática/UFSM, Santa Maria-RS.

MAHARAJAN, S. Performance of native SPARQL query processors. , [S.l.], 2012.

MONGODB. Disponível em: <<https://www.mongodb.org/>>, acesso em de dezembro 2015.

MULGARA. Disponível em: <[www.mulgara.org/](http://www.mulgara.org/)>, acesso em de dezembro 2015.

OTONI, P. P. **Ambiente para automação via Web Semântica utilizando Linux embarcado em microcontroladores ARM**. 2013. Graduação em Engenharia Elétrica com ênfase em Eletrônica — Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos-SP.

PAGANELLI, F.; BIANCHI, G.; GIULI, D. A Context Model for Context-aware System Design Towards the Ambient Intelligence Vision: Experiences in the eTourism Domain. In: CONFERENCE ON USER INTERFACES FOR ALL, 9., 2007, Berlin, Heidelberg. **Proceedings...** Springer-Verlag, 2007. p.173–191. (ERCIM'06).

PATEL, P.; JARDOSH, S.; CHAUDHARY, S.; RANJAN, P. Context Aware Middleware Architecture for Wireless Sensor Network. In: IEEE SCC, 2009. **Anais...** IEEE Computer Society, 2009. p.532–535.

PAWAR, P.; BOROS, H.; LIU, F.; HEIJENK, G. J.; BEIJNUM, B.-J. van. Bridging Context Management Systems in the ad hoc and mobile environments. In: ISCC, 2009. **Anais...** IEEE, 2009. p.882–888.

PERERA, C.; ZASLAVSKY, A.; CHRISTEN, P.; GEORGAKOPOULOS, D. Context Aware Computing for The Internet of Things: A Survey. **Communications Surveys Tutorials, IEEE**, [S.l.], v.16, n.1, p.414–454, First 2014.

PERNAS, A. M. F. **Sensibilidade à Situação em Sistemas Educacionais na Web**. 2012. Tese de Doutorado em Ciência da Computação — Instituto de Informática-UFRGS, Porto Alegre-RS.

PETRELLI, D.; NOT, E.; STRAPPARAVA, C.; STOCK, O.; ZANCANARO, M. Modeling Context i s Like Taking Pictures. In: CONFERENCE ON HUMAN FACTORS IN COMPUTERS, WORKSHOP "THE WHAT, WHO, WHERE, WHEN, WHY AND HOW OF CONTEXT-AWARENESS", 2000. **Anais...** [S.l.: s.n.], 2000.

PIETSCHMANN, S.; MITSCHICK, A.; WINKLER, R.; MEISSNER, K. CroCo: Ontology-Based, Cross-Application Context Management. In: SEMANTIC MEDIA ADAPTATION AND PERSONALIZATION, 2008. SMAP '08. THIRD INTERNATIONAL WORKSHOP ON, 2008. **Anais...** [S.l.: s.n.], 2008. p.88–93.

POLITOWSKI, C.; MARAN, V. Comparação de Performance entre PostgreSQL e MongoDB. **Escola Regional de Banco de Dados, At São Francisco do Sul**, [S.l.], 2014.

POSTGRESQL. Disponível em: <<http://www.postgresql.org/>>, acesso em de dezembro 2015.

PRAMOD J SADALAGE, M. F. **NoSQL Essencial, Um Guia Conciso para o Mundo Emergente da Persistência Poliglota**. [S.l.]: Novatec, 2013.

RAMPARANY, F.; POORTINGA, R.; STIKIC, M.; SCHMALENSTROEER, J.; PRANTE, T. An open context information management infrastructure the IST-amigo project. In: IET INTERNATIONAL CONFERENCE ON INTELLIGENT ENVIRONMENTS (IE 2007), 3., 2007. **Anais...** [S.l.: s.n.], 2007. p.398–403.

RAZZAQUE, M.; MILOJEVIC-JEVRIC, M.; PALADE, A.; CLARKE, S. Middleware for Internet of Things: A Survey. **Internet of Things Journal, IEEE**, [S.l.], v.3, n.1, p.70–95, Feb 2016.

REETZ, E. S.; TÖNJES, R.; BAKER, N. Towards Global Smart Spaces: Merge Wireless Sensor Networks into Context-aware Systems. In: IEEE INTERNATIONAL CONFERENCE ON WIRELESS PERVASIVE COMPUTING, 5., 2010, Piscataway, NJ, USA. **Proceedings...** IEEE Press, 2010. p.337–342. (ISWPC'10).

RIBONI, D.; BETTINI, C. Context-Aware Activity Recognition through a Combination of Ontological and Statistical Reasoning. In: UIC, 2009. **Anais...** Springer, 2009. p.39–53. (Lecture Notes in Computer Science, v.5585).

ROCHA, R. da; CASANOVA, M.; ENDLER, M. Promoting Efficiency and Separation of Concerns through a Hybrid Model Based on Ontologies for Context-Aware Computing.

In: PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS, 2007. PERCOM WORKSHOPS '07. FIFTH ANNUAL IEEE INTERNATIONAL CONFERENCE ON, 2007. **Anais...** [S.l.: s.n.], 2007. p.9–13.

ROSA, D.; RAMBO, I.; MACHADO, R.; ALMEIDA, R.; RIPPEL, H.; YAMIN, A.; PERNAS, A. Uma abordagem híbrida para armazenamento de dados de contexto no EXEHDA. In: ESCOLA REGIONAL DE REDES DE COMPUTADORES, 2015. **Anais...** [S.l.: s.n.], 2015.

ROUSSAKI, I.; STRIMPAKOU, M.; KALATZIS, N.; ANAGNOSTOU, M.; PILS, C. Hybrid context modeling: a location-based scheme using ontologies. In: PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS, 2006. PERCOM WORKSHOPS 2006. FOURTH ANNUAL IEEE INTERNATIONAL CONFERENCE ON, 2006. **Anais...** [S.l.: s.n.], 2006. p.6 pp.–7.

SANCHEZ, L.; LANZA, J.; OLSEN, R.; BAUER, M.; GIROD-GENET, M. A Generic Context Management Framework for Personal Networking Environments. **Mobile and Ubiquitous Systems, Annual International Conference on**, Los Alamitos, CA, USA, v.0, p.1–8, 2006.

SCHERER, A. P. Z.; JACOBSEN, D. G.; SANTOS, M. L. dos. **PostgreSQL**: instalando e conhecendo seus recursos. Porto Alegre, RS, Brasil: Faculdade Dom Bosco de Porto Alegre, 2008.

SEMANTICMEDIAWIKI. Disponível em: <[semantic-mediawiki.org/wiki/Help:Using\\_SPARQL\\_and\\_RDF\\_stores](http://semantic-mediawiki.org/wiki/Help:Using_SPARQL_and_RDF_stores)>, acesso em de dezembro 2015.

SESAME. Disponível em: <[rdf4j.org/](http://rdf4j.org/)>, acesso em de dezembro 2015.

SHAEIB, A.; CAPPELLARI, P.; ROANTREE, M. A framework for real-time context provision in ubiquitous sensing environments. In: ISCC, 2010. **Anais...** IEEE, 2010. p.1083–1085.

SHARP, J.; MCMURTRY, D.; OAKLEY, A.; SUBRAMANIAN, M.; ZHANG, H. **Data Access for Highly-Scalable Solutions**: Using SQL, NoSQL, and Polyglot Persistence. 1st.ed. [S.l.]: Microsoft patterns &#38; practices, 2013.

SIMONS, C. CMP: A UML Context Modeling Profile for Mobile Distributed Systems. In: HICSS, 2007. **Anais...** IEEE Computer Society, 2007. p.289.

SIRIN, E.; PARSIA, B.; GRAU, B. C.; KALYANPUR, A.; KATZ, Y. Pellet: A Practical OWL-DL Reasoner. **Web Semantics: Science, Services and Agents on the World Wide Web**, Amsterdam, The Netherlands, v.5, n.2, p.51–53, June 2007.

STARDOG. Disponível em: <[stardog.com/](http://stardog.com/)>, acesso em de dezembro 2015.

STRANG, T.; LINNHOF-POPIEN, C. A Context Modeling Survey. In: IN: WORKSHOP ON ADVANCED CONTEXT MODELLING, REASONING AND MANAGEMENT, UBI-COMP 2004 - THE SIXTH INTERNATIONAL CONFERENCE ON UBIQUITOUS COMPUTING, NOTTINGHAM/ENGLAND, 2004. **Anais...** [S.l.: s.n.], 2004.

STRANG, T.; LINNHOF-POPIEN, C.; FRANK, K. CoOL: A Context Ontology Language to Enable Contextual Interoperability. In: STEFANI, J.-B.; DEMEURE, I.; HAGIMONT, D. (Ed.). **Distributed Applications and Interoperable Systems**. [S.l.]: Springer Berlin / Heidelberg, 2003. p.236–247. (Lecture Notes in Computer Science, v.2893).

UNDERCOFFER, J.; JOSHI, A.; PINKSTON, J. Modeling Computer Attacks: An Ontology for Intrusion Detection. In: VIGNA, G.; KRUEGEL, C.; JONSSON, E. (Ed.). **Recent Advances in Intrusion Detection**. [S.l.]: Springer Berlin Heidelberg, 2003. p.113–135. (Lecture Notes in Computer Science, v.2820).

VANATHI, B. **Context Management Using Ontology and Object Relational Database Management System**. 2013. Doctor of Philosophy — Faculty of Information and Communication Engineering Anna University, Chennai.

VIEIRA, V.; SOUZA, D.; SALGADO, A. C.; TEDESCO, P. **Uso e Representação de Contexto em Sistemas Computacionais**. [S.l.]: UFSCAR, 2006. p.127–166.

VIRGILIO, R. D.; TORLONE, R. A general methodology for context-aware data access. In: MOBIDE, 2005. **Anais...** ACM, 2005. p.9–15.

VIRTUOSO. Disponível em: <<http://virtuoso.openlinksw.com/>>, acesso em de dezembro 2015.

WEISER, M. The Computer for the 21st Century. **Scientific American**, [S.l.], v.265, n.3, p.66–75, January 1991.

WIBISONO, W.; ZASLAVSKY, A. B.; LING, S. CoMiHoC: A Middleware Framework for Context Management in MANET Environment. In: AINA, 2010. **Anais...** IEEE Computer Society, 2010. p.620–627.

WILSON, D. H.; LONG, A. C.; ATKESON, C. A context-aware recognition survey for data collection using ubiquitous sensors in the home. In: CHI EXTENDED ABSTRACTS, 2005. **Anais...** ACM, 2005. p.1865–1868.

YAMIN, A. C. **Arquitetura para um Ambiente de Grade Computacional Direcionada às Aplicações Distribuídas, Móveis e Conscientes de Contexto da**

**Computação Pervasiva.** 2004. Tese de Doutorado em Ciência da Computação — Instituto de Informática/UFRGS, Porto Alegre-RS.

ZHOU, X.; TANG, X.; YUAN, X.; CHEN, D. SPBCA: Semantic Pattern-Based Context-Aware Middleware. In: ICPADS, 2009. **Anais...** IEEE, 2009. p.891–895.