

UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

FELIPE DA COSTA NOGUEZ

ONISCIENTE: UM MODELO DE GERAÇÃO DE CONTEXTO BASEADO EM RFID E
SENSORES

SÃO LEOPOLDO
2013

Felipe da Costa Noguez

ONISCIENTE: UM MODELO DE GERAÇÃO DE CONTEXTO BASEADO EM RFID E
SENSORES

Dissertação apresentada como requisito parcial
para a obtenção do título de Mestre pelo
Programa de Pós-Graduação em Computação
Aplicada da Universidade do Vale do Rio dos
Sinos — UNISINOS

Orientador:
Prof. Dr. Cristiano André da Costa

Coorientador:
Prof. Dr. Jorge Luis Victoria Barbosa

São Leopoldo
2013

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)

Noguez, Felipe da Costa

Onisciente: Um modelo de geração de contexto baseado em RFID e sensores / Felipe da Costa Noguez — 2013.

121 f.: il.; 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, 2013.

“Orientador: Prof. Dr. Cristiano André da Costa, Unidade Acadêmica de Pesquisa e Pós-Graduação”.

1. Ontologia. 2. Sensibilidade ao Contexto. 3. Sensores. 4. RFID. 5. Monitoramento. 6. Computação Móvel. 7. Computação Ubíqua. I. Título.

CDU 004.75.057.5

Bibliotecária Vanessa Borges Nunes — CRB 10/1556

(Esta folha serve somente para guardar o lugar da verdadeira folha de aprovação, que é obtida após a defesa do trabalho. Este item é obrigatório, exceto no caso de TCCs.)

AGRADECIMENTOS

Primeiramente, agradeço a minha família pelo apoio demonstrado durante o tempo necessário para a conclusão do curso de mestrado. Em especial, ao meu pai, que sempre esteve disposto a auxiliar nas correções de texto e a trocar ideias construtivas que ajudassem a melhorar este trabalho; e a minha mãe, que não está mais presente entre nós e, que mesmo doente sempre demonstrou seu apoio e incentivo para que eu pudesse terminar este trabalho.

Também agradeço aos meus amigos, que entenderam os motivos do meu desaparecimento nos finais de semana, em especial ao Iuri, que em momentos obscuros de desenvolvimento trouxe a luz para algumas questões técnicas que travavam a evolução deste trabalho.

À minha namorada Tamiris que soube aturar os momentos de estresse que surgiram durante este percurso, além do entendimento, a paciência e os aconselhamentos demonstrados .

Agradeço aos meus orientadores Prof. Dr. Cristiano e Prof. Dr. Jorge pelo suporte e incontáveis puxões de orelha com o intuito de fazer com que este trabalho atingisse o seu maior potencial.

Por fim, agradeço a Sawluz pelo financiamento deste projeto, sem o qual nada disso seria possível.

Obrigado a todos!

RESUMO

Em determinadas situações é interessante poder ter a capacidade de saber qual a situação pela qual algo ou alguém está passando. Possuir tais informações pode trazer vantagens a quem deseja zelar ou controlar alguma situação. Neste âmbito, o presente trabalho tem como finalidade apresentar o Onisciente, um modelo que utiliza a informação de um conjunto de sensores que estejam monitorando um mesmo elemento para determinar o seu contexto e obter conhecimento sobre as condições do mesmo. Para isso, o Onisciente realiza a identificação destas entidades, com o auxílio de uma leitora RFID acrescida de alguns sensores. Estes sensores capturam dados relacionados ao ambiente em que as entidades estão, ou dados da própria entidade em questão, realizando um tratamento e processamento, dando origem a um contexto. No Onisciente, um contexto representa uma imagem de momento preciso no tempo, na qual, ao analisarmos, podemos verificar fatos que estavam ocorrendo e com isso tomar decisões. As decisões tomadas pelo Onisciente envolvem classificar os contextos gerados, e conforme a classificação obtida, informar eventuais situações desfavoráveis às entidades monitoradas. Para determinar um contexto e realizar a classificação do significado que ele representa, foram empregados, além das tecnologias RFID e de sensores, ontologias, computação sensível ao contexto e modelagem de contexto. A coleta de informações é realizada por meio de um Arduino, dotado de variados tipos de sensores, e transmitida com o auxílio de um dispositivo móvel Android para o servidor do Onisciente. O servidor aplica os dados recebidos em uma ontologia que processa e gera os Contextos. Estes contextos gerados podem ser visualizados por meio de uma aplicação para dispositivos móveis ou por meio de uma página web. Uma série de experimentos foram realizados para avaliar a funcionalidade do Onisciente. Primeiramente testes específicos foram aplicados à ontologia objetivando analisar a qualidade de sua constituição, funcionalidade e qualidade dos contextos gerados. Em seguida, estudos de caso foram desenvolvidos para testar a funcionalidade e desempenho do Onisciente. Os resultados obtidos determinaram que o modelo apresentado neste trabalho conseguiu atingir os objetivos propostos ao gerar contextos precisos ao final dos testes realizados, tendo como destaques, além da capacidade de realizar as mesmas tarefas apresentadas pelos trabalhos relacionados estudados, apresentar recursos para integração com diversas tecnologias por meio de *web services*, dispor de variados recursos para o acompanhamento dos resultados obtidos e ter a capacidade de utilizar diversos tipos de RFID e sensores.

Palavras-chave: Ontologia. Sensibilidade ao Contexto. Sensores. RFID. Monitoramento. Computação Móvel. Computação Ubíqua.

ABSTRACT

In determined situations is interesting to have the capacity of know in which situation something or someone is. Have this information can bring advantages for whom desires to care or control some situation. In this scope the present work have as main finality to present the Onisciente, a model that uses the information from a group of sensors that are monitoring a same element to determine its context and obtain knowledge about its condition. To accomplish that, the Onisciente makes the identification of those entities using a RFID reader with some sensors. Those sensors acquire data related with the environment where the entities are placed in, or data directly from the entities, making some treatment and processing it, generating the context. In Onisciente, a context represents an image of a moment in the time, in which, when analyzed, allow us to verify facts that were occurring and make some decisions. Decisions made by the Onisciente involves classify the generated contexts and, according with the obtained classification, inform unfavorable situations of the monitored entities. To determine a context and realize the classification of what it means, where used beyond the RFID technology and sensor, ontology, context-aware computing and context modeling. The data collection is made using an Arduino hardware with several sensors, and transmitted, with the support of a mobile Android device, to the Oniscient server. The server applies the received data in an ontology that process then and generate the contexts. Those generated contexts can be visualized using and app for mobile devices or through a web page. A series of experiments were realized to evaluate the functionality of the Onisciente. Firstly, specific tests were applied over the ontology to analyze the quality of its constitution, functionality and quality of the generated contexts. Then, use cases were elaborated to test the Onisciente's functionality and performance. The acquired results determined that the presented model in this work managed to achieve the proposed objectives by generating accurate contexts in the end of the tests, having as highlights, beyond the capacity of realize the same tasks presented in the studied related works, to present resources to integrates with several technologies through web services, dispose of varied resources for the follow the obtained results and to have the capacity of use several types of RFID and sensors.

Keywords: Ontology. Context-Aware. Sensors. RFID. Monitoring. Mobile Computing. Ubiquitous Computing.

LISTA DE FIGURAS

Figura 1:	Exemplo de situação de contexto	18
Figura 2:	Técnica de triangulação	27
Figura 3:	Técnica de fingerprint	28
Figura 4:	Técnica de aproximação	29
Figura 5:	Técnica de AOA	29
Figura 6:	Arquitetura em três camadas do modelo	32
Figura 7:	Arquitetura em três camadas utilizado pelo modelo	34
Figura 8:	Arquitetura em três camadas utilizado pelo modelo	35
Figura 9:	Arquitetura em três camadas utilizado pelo modelo	38
Figura 10:	Arquitetura em três camadas	39
Figura 11:	Arquitetura em três camadas utilizado no modelo	40
Figura 12:	Visão Geral do modelo	44
Figura 13:	Casos de Uso do Servidor	47
Figura 14:	Casos de Uso do Cliente	48
Figura 15:	Arquitetura Geral do modelo	49
Figura 16:	Arquitetura do Servidor	50
Figura 17:	Diagrama de Atividades - Obter dados de Hardware	52
Figura 18:	Diagrama de Atividades - Filtrar Dados	53
Figura 19:	Diagrama de Atividades - Gerar Contexto	53
Figura 20:	Diagrama de Atividades - Analisar Condição de Entidade	54
Figura 21:	Diagrama de Atividades - Gerar Alarme	55
Figura 22:	Diagrama de Atividades - Realizar Operação de Entidade	56
Figura 23:	Diagrama de Atividades - Atualizar Ontologia	57
Figura 24:	Arquitetura do Cliente Web	58
Figura 25:	Arquitetura do Cliente Móvel	59
Figura 26:	Diagrama de Atividades - Cadastrar Entidade	60
Figura 27:	Diagrama de Atividades - Editar Entidade	60
Figura 28:	Diagrama de Atividades - Monitorar Entidade	61
Figura 29:	Diagrama de Atividades - Obter Informação de Contexto	62
Figura 30:	Diagrama de Atividades - Verificar Alarme	62
Figura 31:	Diagrama de Atividades - Analisar Passado de Entidade - Website	63
Figura 32:	Diagrama de Atividades - Analisar Passado de Entidade - Dispositivo Móvel	64
Figura 33:	Estrutura Geral de Classes da Ontologia	66
Figura 34:	Estrutura das Classes de Contexto	66
Figura 35:	Estrutura da informação de contexto após ser aplicada à ontologia	67
Figura 36:	Equipamentos utilizados para obter dados para o Onisciente	68
Figura 37:	Hardware utilizado no Onisciente	69
Figura 38:	Diagrama de classes do servidor	70
Figura 39:	Classes de entidade da ontologia e suas relações	74
Figura 40:	Eixos	76
Figura 41:	Contextos do Onisciente	77
Figura 42:	Consulta SPARQL que cria o contexto de pessoa febril.	81
Figura 43:	Quadro com valores de sensação térmica para valores de temperatura em graus °C	83

Figura 44:	Modelo entidade-relacionamento do banco de dados do onisciente	84
Figura 45:	Diagrama de classes do cliente Android	86
Figura 46:	Tela de login, monitoramento, seleção de entidade a monitorar e seleção de nova entidade.	88
Figura 47:	Tela de seleção de entidades e confirmação de escolha.	89
Figura 48:	Telas de cadastro do cliente Android.	89
Figura 49:	Telas de escolha de entidade a ser monitorada, definição de relação de entidades e adição de atributos.	90
Figura 50:	Diagrama de classes do cliente web	91
Figura 51:	Telas de login e monitoramento.	92
Figura 52:	Menu de opções e lista de objetos cadastrados.	93
Figura 53:	Páginas de cadastro de entidades e sensores.	94
Figura 54:	Inclusão de novo atributo	95
Figura 55:	Ferramenta Web OOPS!	97
Figura 56:	Lista de problemas de estrutura de ontologias detectáveis pelo OOPS!	98
Figura 57:	Resultado obtido ao utilizar o OOPS!	99
Figura 58:	Consulta SPARQL que procurará por objetos danificados.	101
Figura 59:	Consulta SPARQL que verificará objetos que estejam virados.	101
Figura 60:	Consulta SPARQL que procurará por pessoas que não estão confortáveis.	103
Figura 61:	Consulta SPARQL que procurará por pessoas com sensação de calor.	103
Figura 62:	Consulta SPARQL que procurará por pessoas com sensação de frio.	103
Figura 63:	Consulta SPARQL que procurará por objetos em temperatura alta.	103
Figura 64:	Consulta SPARQL que procurará por objetos em umidade baixa.	104
Figura 65:	Consulta SPARQL que procurará por objetos em temperatura baixa.	104
Figura 66:	Consulta SPARQL que procurará por objetos em temperatura alta.	104
Figura 67:	Consulta SPARQL que calcula a sensação térmica de uma pessoa.	106
Figura 68:	Consulta SPARQL que procurará por pessoas que não estão confortáveis baseado na sua sensação térmica.	106
Figura 69:	Consulta SPARQL que procurará por pessoas com sensação de calor baseado na sua sensação térmica.	107
Figura 70:	Consulta SPARQL que procurará por pessoas com sensação de frio baseado na sua sensação térmica.	107
Figura 71:	Rotas do primeiro caso de uso.	108
Figura 72:	Imagens do trecho 1 do primeiro caso de uso.	109
Figura 73:	Imagens do trecho 2 do primeiro caso de uso.	109
Figura 74:	Gráfico de temperatura do segundo caso de uso	112
Figura 75:	Gráfico de umidade do segundo caso de uso	113

LISTA DE TABELAS

Tabela 1:	Resumo da evolução do RFID	24
Tabela 2:	Características de cada frequência	26
Tabela 3:	Trabalhos que geram contexto a partir de informações obtidas por RFID . .	36
Tabela 4:	Trabalhos que utilizam RFID com sensores	42
Tabela 5:	Classificação de desconforto humano	82
Tabela 6:	Contextos de sensação de desconforto das pessoas em cada situação do cenário 2.	105
Tabela 7:	Contextos de objetos registrados em cada situação do cenário 2.	105
Tabela 8:	Contextos gerados ao final do cenário 2.	105
Tabela 9:	Registro de contextos do trecho 1	110
Tabela 10:	Registro de contextos do trecho 2	111
Tabela 11:	Comparativo entre o Onisciente e os trabalhos relacionados	116

LISTA DE SIGLAS

INMET	Instituto Nacional de Meteorologia
RFID	Radio Frequency Identification
GPS	Global Position System
XML	Extensible Markup Language
SLI	Sistema de Localização Indoor
OWL	Web Ontology Language
SOA	Arquitetura Orientada a Serviços
TAM	Technical Architecture Modeling Standard
DAO	Data Access Object
REST	Representational State Transfer
CRUD	Criar, Remover, Atualizar e Excluir
LINQ	Language Integrated Query
JSON	JavaScript Object Notation
GPRS	General Packet Radio Service
HTTPS	HyperText Transfer Protocol Secure
SPARQL	SPARQL Protocol and RDF Query Language
AJAX	Asynchronous Javascript and XML
EPC	Eletronic Product Code

SUMÁRIO

1 INTRODUÇÃO	13
1.1 Motivação	13
1.2 Questão de pesquisa	14
1.3 Objetivos	15
1.4 Estrutura do texto	16
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 Contexto	17
2.1.1 Computação sensível ao contexto	18
2.1.2 Modelagem de contexto	20
2.1.3 Ontologias	21
2.2 RFID	22
2.2.1 História do RFID	23
2.2.2 Tecnologia RFID	24
2.2.3 RFID e Sistemas de Localização Indoor	27
2.2.4 RFID e Sensibilidade ao Contexto	30
3 TRABALHOS RELACIONADOS	31
3.1 Grupo 1: Modelagem de contexto baseado em RFID	31
3.1.1 <i>A RFID-based Context-Aware Service Model</i>	31
3.1.2 <i>A Context-Aware Smart Infrastructure based on RFID Sensor-tags and its Application to the Health-Care Domain</i>	33
3.1.3 <i>An Extended Context Model in a RFID-based Context-Aware Service System</i>	34
3.1.4 Discussão sobre os trabalhos do grupo 1	35
3.2 Grupo 2: Exploração de dados obtidos por meio de RFID com sensores	36
3.2.1 <i>The Promise of RFID-Based Sensors in the Perishables Supply Chain</i>	37
3.2.2 <i>A simulation approach for optimal design of RFID sensor tag-based cold chain system</i>	38
3.2.3 <i>Improvement in the tracking of special loads by using a three-level RFID system</i>	40
3.2.4 Discussão sobre os trabalhos do grupo 2	41
3.3 Considerações sobre os trabalhos relacionados	43
4 ONISCIENTE	44
4.1 Visão geral do modelo	44
4.2 Requisitos	45
4.3 Casos de Uso	46
4.3.1 Casos de uso do servidor	46
4.3.2 Casos de uso do cliente	47
4.4 Arquitetura	49
4.4.1 Servidor	49
4.4.2 Diagramas de Atividade do Servidor	51
4.4.3 Cliente Site Web	55
4.4.4 Cliente Móvel	57
4.4.5 Diagramas de atividade do cliente	58
4.5 Modelo de representação da entidade	63

4.6 Modelo de Contexto	64
5 IMPLEMENTAÇÃO	68
5.1 Tecnologias Utilizadas	68
5.2 Servidor	69
5.2.1 Classes do Servidor	70
5.2.2 Aquisição de Dados	72
5.2.3 Interação com a camada física	73
5.2.4 Controle de entidades monitoradas	73
5.2.5 Ontologia	73
5.2.6 Gestão da Ontologia	79
5.2.7 Geração de Contexto	80
5.2.8 Descrição de Contextos	82
5.2.9 Banco de Dados	83
5.3 Clientes	85
5.3.1 Cliente Android	85
5.3.2 Cliente Web	90
6 AVALIAÇÃO	96
6.1 Metodologia de avaliação	96
6.2 Avaliação da Ontologia	99
6.2.1 Avaliação por métricas	99
6.2.2 Avaliação por cenários	100
6.3 Avaliação do Onisciente	107
6.3.1 Primeiro caso de uso	107
6.3.2 Resultados do primeiro caso de uso	110
6.3.3 Segundo caso de uso	111
6.3.4 Resultados do segundo caso de uso	112
6.4 Discussão geral sobre as avaliações	113
7 CONSIDERAÇÕES FINAIS	114
7.1 Conclusões	114
7.2 Principais contribuições	115
7.3 Trabalhos Futuros	116
REFERÊNCIAS	118

1 INTRODUÇÃO

Ao final da segunda guerra, uma tecnologia que permitia detectar se um amigo ou inimigo estava se aproximando foi idealizada (STOCKMAN, 1948), mas esta tecnologia teve de esperar cerca de 30 anos para poder ser empregada (LANDT, 2005). Hoje, o RFID é utilizado em diversos tipos de aplicações e em diversas áreas, que vão, desde controle de acesso até monitoramento como, por exemplo, na cadeia de suprimentos e na área da saúde (ROUSSOS; KOSTAKOS, 2009).

Ao utilizar o RFID é possível determinar a posição de uma entidade dentro de um ambiente, isto é possível graças a algumas técnicas de localização existentes (BOUET; SANTOS, 2008). Uma entidade é qualquer ser vivo, objeto ou local que possa ser monitorado. Ao definirmos a localização de pelo menos duas ou mais entidades, ao mesmo tempo, e em um mesmo ambiente, teremos condições de descrever a ocorrência de uma situação em uma determinada fração de tempo (NOGUEZ F; COSTA, 2012). Esta situação representa um contexto (DEY, 2001) (MEHRA, 2012), que pode ser definido como uma fotografia de um momento. Alguns modelos de RFID permitem que sejam anexados sensores (ROUSSOS; KOSTAKOS, 2009), elevando a quantidade e tipos de dados que podem ser adquiríveis, fornecendo uma variedade de informação que, se adicionada a um contexto existente, permite que este seja complementado, aumentando o grau de precisão e o nível de detalhamento das condições que as entidades, nele descritas, se encontram.

Com base nisto, este trabalho procura apresentar um modelo que faça a identificação de entidades por meio do uso de RFID e também uma síntese das condições em que ela se encontra por meio de contextos, mediante a coleta de informações adquiridas por diversos tipos de sensores. Este modelo foi batizado de Onisciente, pois o principal objetivo dele é permitir ter conhecimento sobre todas as condições de uma entidade.

1.1 Motivação

Poder definir contextos a partir da combinação de informações brutas de diversas fontes individuais e variadas podem trazer benefícios a diversas áreas. Por exemplo, na saúde, contextos resultantes do monitoramento constante de pacientes por meio de sensores pode elevar o nível de agilidade e precisão na análise de médicos sobre as suas condições; na cadeia de suprimentos e logística, permite que sejam obtidas informações sobre as condições de uma mercadoria em qualquer momento da fase de produção, transporte ou armazenamento, além de permitir adquirir conhecimento sobre as condições das estradas e outros dados, como tempo de viagem e tráfego.

Ter conhecimento sobre a situação de uma entidade em qualquer momento permite que as empresas evitem desperdícios, aumentem seus lucros e alterem planejamentos em busca de melhorias pontuais dentro de sua metodologia de trabalho (SHAMSUZZOHA A.; HELO,

2011). Por esse motivo, possuir uma ferramenta que possa adquirir e interpretar informações diversas, de fontes variadas, e apresente um resultado que descreva o que está acontecendo com uma entidade, quase que em tempo real, teria ampla utilidade no meio empresarial. Por outro lado, esse tipo de ferramenta permitiria que pesquisas na área da computação sensível ao contexto possam evoluir mais rapidamente.

A tecnologia RFID vem sendo estudada mundialmente com o objetivo de explorar a sua aplicabilidade e os desafios encontrados para sua implementação, visando desenvolver soluções que possam agregar eficiência.

Durante o processo de pesquisa não foram encontrados trabalhos que apresentassem um modelo que abordasse todas as etapas na elaboração de contexto a partir de dados oriundos de sensores. Os trabalhos que foram encontrados tinham como o seu maior foco ou o processo de coleta de informações, ou o processo de tratamento de dados para a geração de contextos. Entre os trabalhos que tratam da coleta de informações, graças a Delen et. al (2011), Kang et. al (2012) e Escribano et. al(2012), foi possível determinar quais os tipos de sensores mais utilizados na área de sensibilidade ao contexto, além do tipo de RFID que atende as necessidades impostas para o funcionamento do Onisciente e também padrões de tecnologia mais utilizados como arquitetura e método de coleta de dados. Quanto aos trabalhos que abordavam a coleta de informações, Sheng et. al(2011), Catarinucci et. al(2009) e Zhang, Zheng e Liu (2008) forneceram informações relevantes para a definição do método de modelagem de contexto mais indicado, além do formato de representação de ontologia que deveria ser utilizado e a maneira como a representação de contexto deveria ser feita. De posse destas informações, foi possível traçar um objetivo para o Onisciente que é apresentado na próxima seção.

1.2 Questão de pesquisa

Atualmente o RFID é uma tecnologia que tem sido cada vez mais empregada, se tornando bastante presente no dia a dia das pessoas mesmo que elas não saibam. A capacidade de expansão dos limites dessa tecnologia, com incorporação de sensores, a torna apta a ser utilizada em inúmeras áreas para uma diversidade de finalidades diferentes. Com o RFID podemos monitorar a situação da segurança de trabalhadores em uma construção, ao obter suas localizações e atividades, e relacioná-las com o funcionamento de equipamentos da obra. Também podemos estender para o meio veicular, na qual poderíamos obter a condição de veículos ao passarem por pontos pré-determinados. Ou, ainda, monitorar o que crianças realizam dentro das escolas, por onde mais andam, com quem passam mais tempo, etc. Com base nestas situações mais a constante expansão no uso do RFID e as informações que podem ser adquiridas a partir de sensores, pensou-se na seguinte questão de pesquisa:

"Como seria um modelo que permitisse acompanhar diversos contextos de entidades utilizando RFID e sensores?"

Imagine que inúmeros objetos estejam armazenados em um local amplo, onde cada um desses objetos possui um identificador RFID e pode ser facilmente localizado. Agora, qual a situação desse objeto dentro do ambiente? Ele está armazenado corretamente? O local onde ele está atende aos requisitos necessários para um maior tempo de vida útil? Ele está em perfeitas condições? Para uma pessoa obter os dados que façam com que essas perguntas possam ser respondidas pode consumir uma grande quantidade de tempo, isso porque deve ser levado em consideração que esses dados devem ser comparados com as características individuais de cada objeto do ambiente. O Onisciente surge como um recurso que realiza uma compilação entre os dados do ambiente com as características das entidades, no exemplo acima, *Objetos*, fornecendo respostas que informem a real situação das entidades por meio de contextos.

Ao final deste trabalho, pretende-se apresentar, como maior contribuição, um modelo que possibilite monitorar tipos variados de entidades, sob variadas condições ambientais, sendo que, ao final do processo, seja possível obter uma leitura da sua situação ou determinar a influência que o ambiente exerce sobre ela por meio da análise dos contextos gerados. O diferencial deste modelo, comparado com os trabalhos estudados em sua concepção, está no fato de que ele é modelado de forma genérica, tornando-o capaz de ser aplicado em diversas áreas.

1.3 Objetivos

O principal objetivo do Onisciente é ser um modelo para a geração de contexto baseado em informações adquiridas por sensores e RFID. Como principal resultado, é esperado que o Onisciente forneça uma descrição que permita a interpretação das condições em que uma determinada entidade monitorada esteve ou se encontra no momento atual.

Para alcançar o objetivo proposto alguns objetivos específicos foram traçados:

- O modelo dever ser capaz de efetuar detecção de temperatura e umidade;
- Deve trabalhar apenas sobre entidades identificadas;
- Deve ter a capacidade de inferir a condição de entidades a partir da informação de um acelerômetro;
- Deve ser genérico e capaz de ser aplicado a diversos tipos de entidades;
- Deve fornecer meios de integração para diversos tipos de tecnologias;
- Deve apresentar resultados legíveis para o usuário final.

E para conseguir executar todas estas etapas, os seguintes passos deverão ser efetuados:

- Estudar RFID e o seu uso com sensores em aplicações de monitoramento;
- Estudar Contexto e Modelagem de Contexto e suas áreas de aplicação;

- Pesquisar ontologias e sua aplicação em sistemas sensíveis ao contexto;
- Elaborar um modelo para criação de contextos a partir de informações, ou de nenhum, ou um ou mais sensores, independente do tipo utilizado;
- Implementar um protótipo de modelo a ser proposto;
- Avaliar a eficácia da ontologia na geração de contextos de entidades monitoradas;
- Avaliar o modelo desenvolvido por meio de estudos de caso;
- Analisar o desempenho do Onisciente sob situação de *stress*;

1.4 Estrutura do texto

No capítulo 2 são apresentadas as definições e conceitos sobre Contexto, RFID, Sensores e demais assuntos envolvidos nesse trabalho. O capítulo 3 traz alguns trabalhos que apresentam alguma relevância com o assunto proposto para este trabalho. No capítulo 4 é apresentada uma descrição do modelo do Onisciente para criação de contexto a partir de dados obtidos por sensores e RFID. O capítulo 5 faz o detalhamento do processo de desenvolvimento do modelo. No capítulo 6 são descritos os estudos de casos utilizados para realizar a avaliação do Onisciente e também uma análise sobre os resultados obtidos. Por fim, o capítulo 7 apresenta nossas considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são abordados os assuntos relacionados ao objetivo proposto, procurando conceituar e descrever os conteúdos de forma que se tenha um entendimento básico dos temas envolvidos, melhorando a compreensão e funcionalidades do modelo a ser apresentado. O capítulo inicia apresentando contextos, que é também o assunto relacionado ao objetivo do Onisciente. A seção sobre contextos procura definir o que é contexto, qual o papel que desempenha na computação sensível ao contexto e de que maneiras ele pode ser modelado para que possa ser aproveitado com máxima eficácia. Dentro do assunto modelagem de contexto, é dado um maior destaque sobre ontologias, que, conforme poderá ser observado nos trabalhos relacionados, é amplamente utilizada com contextos. Na evolução do capítulo, também é realizada uma abordagem sobre RFID, um dos principais elementos que constituem o Onisciente. Na discussão sobre o tópico, é abordada a origem da tecnologia, desde a sua idealização até os dias de hoje. Posteriormente, é realizada a apresentação da tecnologia em si, descrevendo o seu papel na sensibilidade ao contexto e em sistemas de localização indoor, sua principal aplicabilidade.

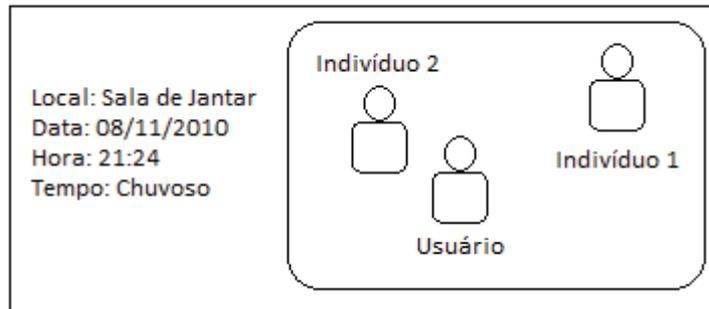
2.1 Contexto

Contexto é uma palavra com significado vago, é muito utilizada em diferentes domínios e, portanto, tem diversas definições. Geralmente se refere ao que está à volta do centro de interesse, oferecendo fontes de informações adicionais como quem, onde, o quê, aumentando a sua compreensão (MOSTEFAOUI; PASQUIER-ROCHA; BREZILLON, 2004). Na computação, a literatura cita diversas definições para contexto. Sua primeira introdução foi dada por Schilit e Theimer (1994), onde eles o definiam como sendo "a localização de uso, a coleção de pessoas e objetos próximos, assim como as mudanças sobre esses objetos ao passar do tempo". A mais aceita, porém, é a de Dey (2001) onde ele diz que,

contexto é qualquer informação que possa ser utilizada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, lugar ou objeto que seja considerado relevante para a interação entre um usuário e uma aplicação, incluindo o usuário e a aplicação.

Contexto pode ser composto por informações perceptivas referentes ao ambiente, localização, social e tempo (HOAREAU; SATOH, 2009) (Figura 1), informando, além das condições climáticas, onde uma entidade está precisamente localizada, quem está ao redor, e o que existe em sua proximidade (SCHILIT; ADAMS; WANT, 1994).

Diferentes campos da ciência da computação, como processamento de linguagem natural, aprendizagem de máquina, visão computacional, suporte a decisão, recuperação de informação, computação pervasiva e segurança, utilizam contexto para adicionar adaptação e efetividade na

Figura 1: Exemplo de situação de contexto

Fonte: Elaborada pelo autor.

tomada de decisões (MOSTEFAOUI; PASQUIER-ROCHA; BREZILLON, 2004).

A computação pervasiva está intimamente relacionada com contexto devido à grande heterogeneidade e ubiquidade de comunicação desses tipos de entidades, onde ambas necessitam adaptar-se em tempo real às situações de uso ou contexto (MOSTEFAOUI; PASQUIER-ROCHA; BREZILLON, 2004).

Mostéfaoui (2004 apud CHALMERS, 2002) cita cinco usos de informações contextuais em ambientes pervasivos:

- Sensibilidade contextual, onde o contexto é adquirido e a descrição da informação descrevendo o contexto atual é apresentada ao usuário;
- Associar contexto com dados, conhecido como aumento contextual;
- Habilitar descoberta de recursos contextuais: utiliza recursos mais próximos;
- Ações disparadas por contextos: contexto é utilizado para realizar ações;
- Mediação contextual: contexto é utilizado para modificar serviços;

2.1.1 Computação sensível ao contexto

A computação sensível ao contexto é aquela em que as informações pertencentes ao mundo físico são adquiridas, processadas e utilizadas para fornecer uma variedade de serviços para o usuário, de acordo com a situação (HOAREAU; SATOH, 2009). O termo foi introduzido por Schilit, Adams e Want (1994) como "um software que adapta-se de acordo com sua localização, a coleção de pessoas e objetos à sua volta, assim como as mudanças sobre esses objetos ao passar do tempo". Outra definição foi dada por Dey (2001), dizendo que: "Um sistema é sensível ao contexto, se este utiliza contexto para fornecer informações relevantes e/ou serviços ao usuário, onde a relevância depende da atividade do usuário."

Segundo Chen e Kotz (2000)

computação sensível ao contexto é um paradigma da computação móvel na qual as aplicações podem descobrir e tirar vantagem da informação contextual (como localização do usuário, hora do dia, pessoas e dispositivos próximos, e atividade do usuário).

As aplicações sensíveis ao contexto necessitam constantemente adaptar-se às mudanças sobre as informações do contexto, como: localização de uso, pessoas próximas, usuários e dispositivos acessíveis (SCHILIT; ADAMS; WANT, 1994). Para isso, elas necessitam preocupar-se com a aquisição, à abstração e compreensão correspondente a estímulos sensoriais, e ao comportamento da aplicação baseado no reconhecimento (HOAREAU; SATOH, 2009).

Existem três categorias de recursos que uma aplicação sensível ao contexto pode suportar (DEY, 2001):

- Apresentação de informação e serviços para o usuário;
- Execução automática de um serviço para o usuário; e
- Marcação de informação de contexto para auxiliar em uma recuperação posterior.

Chen e Kotz (2000) explicam que existem basicamente duas maneiras de utilizar contexto, ativa e passivamente, respectivamente adaptando-se de acordo com o contexto descoberto ou apresentando-o ao usuário.

Definem ainda, que a computação sensível ao contexto possui duas definições:

- **Sensibilidade ao contexto ativa:** Uma aplicação automaticamente se adapta aos contextos descobertos, alterando o comportamento da aplicação.
- **Sensibilidade ao contexto passiva:** Uma aplicação apresenta o novo ou atualizado contexto para um usuário interessado ou torna o contexto persistente para o usuário recuperar mais tarde.

A aquisição do contexto pode ser obtida de diversas maneiras. Com base em observações, Mostefaoui, Pasquier-Rocha e Brezillon (2004) assinalam três maneiras:

- **Sensoriamento de contexto:** a informação é adquirida por meio de sensores físicos ou de software, tais como temperatura, pressão, iluminação e nível de ruído.
- **Contexto derivado:** este tipo de informação contextual pode ser computado on the fly.
- **Contexto explicitamente fornecido:** preferências do usuário quando este responde um requisito da aplicação.

A aquisição de contexto por meio de sensoriamento não é uma tarefa simples. Informação pode ser adquirida por meio de diversos sensores e acaba necessitando que uma etapa de interpretação seja executada para que seja útil para uma aplicação. Informação contextual é

dinâmica por natureza e essas alterações necessitam ser capturadas pela infraestrutura sensível. Um dos maiores objetivos da computação sensível ao contexto é desenvolver aplicações que façam a coisa certa no momento certo (DEY, 2001).

Para uma melhor exploração do potencial de um contexto, é necessário definir um formato para o armazenamento de sua informação, este assunto será abordado a seguir.

2.1.2 Modelagem de contexto

De nada adianta conseguir captar o contexto se a maneira como ele é armazenado não favorece a aplicação em que ele será utilizado, causando perda de informação ou incompatibilidade. Por isso, modelar a informação do contexto se torna uma forma crucial de garantir a boa funcionalidade do sistema ao qual ele será aplicado.

De acordo com as propriedades da informação captada e seu tipo, uma modelagem pode se tornar mais indicada do que outra. Chen e Kotz (2000) e Hoareau e Satoh (2009) apresentam em seus trabalhos um levantamento de modelos de localização e estruturas de dados utilizadas dentro da modelagem de contexto.

Um modelo de localização utiliza o posicionamento de uma entidade como seu recurso principal, com ele pode-se dizer onde um objeto se encontra ou o que existe em um determinado local. Neste grupo encontram-se três modelos:

- **Modelo geométrico:** usado quando se utiliza as coordenadas capturadas por sensores de posicionamento para representar a posição de entidades: por exemplo, um sistema que utiliza GPS. Apesar da precisão no posicionamento, essas informações não apresentam nenhum significado semântico com relação aos lugares demarcados. Os sistemas baseados em navegação se enquadram nessa categoria;
- **Modelo simbólico:** utilizado quando a topologia e o relacionamento dos dados são mais importantes do que as coordenadas. Os dados e as relações entre eles possuem a mesma importância para o sistema. Nessa categoria se enquadram os sistemas que trabalham divisões simbólicas;
- **Modelo híbrido:** utiliza as vantagens e desvantagens dos modelos anteriores para aplicações que necessitem das informações de localização de ambos, a precisão das coordenadas do modelo geométrico com o significado semântico do modelo simbólico.

A modelagem de contexto é utilizada para informar e trocar a informação de contexto dentro do sistema. Ela utiliza estruturas de dados que servem para dar suporte aos sistemas sensível ao contexto. Os autores citam seis dessas estruturas de dados que tendem a ser as mais usadas:

- **Modelo chave de valor:** ou Key-value, um modelo simples de pares que consiste em utilizar um atributo e um valor, normalmente adquirido por um sensor do ambiente. Exemplo, <Temperatura do quarto, 25> onde "Temperatura do quarto" é a chave e "25" o valor;

- **Modelo de marcação:** este modelo utiliza tags e estruturas de dados em forma de hierarquia, compostos por atributos e conteúdos. Permite o uso recursivo de sua estrutura, permitindo conter outras marcações de dados dentro de seu conteúdo, similar a um arquivo XML;
- **Modelo orientado a objetos:** é baseado no conceito de objetos e relacionamentos da orientação a objetos, foi especificamente desenvolvido para contexto de localização. Fornece uma estrutura flexível e capacidade de uso de encapsulamentos, reutilização e herança;
- **Modelo baseado em lógica:** neste modelo informação contextual inclui dados expressos como fatos de um sistema baseado em regras. É uma maneira intuitiva e bem adaptável de se prototipar aplicações sensíveis ao contexto. O usuário pode inserir novas regras assim como entrar com novas requisições na base de dados;
- **Ontologia:** são muito utilizadas na Web Semântica para representar e raciocinar com contexto. Técnicas baseadas em ontologias suportam um vocabulário para situações predicadas, podendo fornecer semânticas adicionais para capturas de sensores. Ontologias possuem características que as tornam indicadas para a área de sensibilidade ao contexto, merecendo um estudo mais aprofundado que será apresentado no evoluir do capítulo.

2.1.3 Ontologias

Uma ontologia é definida por Gruber (1995) como um meio de expressar conhecimento sobre um determinado domínio. Ela é constituída por um conjunto de objetos pertencentes a esse domínio e a descrição das relações entre esses objetos. Estas relações e objetos formam a base de conhecimento que é utilizada no desenvolvimento de sistemas.

Segundo Noy e McGuinness (2001), a utilização de ontologias é recomendada quando se deseja compartilhar entendimento comum da estrutura de informação entre pessoas ou agentes de software, permitir a reutilização do domínio de conhecimento, tornar suposições de domínio explícitas, separar domínio de conhecimento do conhecimento operacional e analisar domínio de conhecimento.

A reutilização de ontologias permite que o desenvolvimento de um sistema de conhecimento não tenha de ser realizado do zero. Esta característica permite que uma ontologia já bem desenvolvida sobre um assunto específico possa servir como base de uma nova ontologia, onde nesta nova ontologia serão agregadas informações que complementem a ontologia base.

Noy e McGuinness (2001) definem a estrutura de uma ontologia em quatro partes:

- **Classes:** que descrevem os conceitos que integram a ontologia;
- **Slots:** que descrevem as propriedades e atributos de cada classe;
- **Facets:** que representam as restrições das propriedades (Slots);

- **Instâncias:** que representa um objeto pertencente a uma classe.

Para exemplificar: Uma classe Vinhos representa todos os vinhos existentes, um vinho específico é uma Instância dessa classe. As características de um vinho como cor, tipo de uva, cidade de origem, etc, são atributos (Slots) dessa instância. De acordo com Noy e McGuinness (2001) o processo de desenvolvimento de uma ontologia envolve:

- Definir classes para o domínio determinado;
- Arrumar as classes em uma hierarquia taxonômica (Classe-subclasse);
- Definir os Slots para cada classe, definindo os valores permitidos para cada um;
- Preencher os valores para os Slots definidos para cada instância inserida.

Gruber (1995) apresenta uma lista de critérios para a elaboração de uma ontologia; o objetivo desses critérios é servir como um guia para que o processo de criação não saia do foco do objetivo desejado. Os critérios são:

- **Clareza:** As definições da ontologia devem ser objetivas e expressar seu significado;
- **Coerência:** A ontologia deve sancionar inferências que sejam consistentes com as definições;
- **Extensibilidade:** Uma ontologia deve permitir a inclusão de novos termos no vocabulário já existente;
- **Viés de codificação mínima:** A conceituação deve ser especificada no nível de conhecimento sem depender de codificações específicas, porque agentes de compartilhamento de conhecimento podem ser implementados em sistemas e estilos de representação diferentes;
- **Compromisso ontológico mínimo:** A ontologia deve exigir um mínimo de comprometimento ontológico que seja suficiente para as atividades de compartilhamento de conhecimento. Devem-se definir apenas os termos que são essenciais para a sua consistência.

No Onisciente, ontologias representam contextos de entidades que sofrem ação ou reação de influências externas medidas por sensores, entre eles o RFID.

2.2 RFID

Conforme Bouet e Santos (2008) e Liu et al. (2007), o RFID é um meio de armazenar e recuperar dados por meio de transmissão eletromagnética para um circuito de rádio frequência (RF) compatível, podendo ser utilizado para identificação e localização. De acordo com Want (2006), o RFID permite identificação a distância sem a necessidade de uma linha de visão e

pode distinguir variados tipos de identificadores em uma mesma região sem a necessidade de assistência humana.

No restante da seção será apresentada a história do RFID, desde a sua concepção até quando finalmente pôde ser empregada da maneira como é nos tempos atuais. Após isso, será realizada uma descrição de como funciona a tecnologia e suas características. Fechando o conteúdo, falando sobre as suas duas principais aplicações, sistemas de localização indoor e sensibilidade ao contexto.

2.2.1 História do RFID

Segundo Landt (2005) após o surgimento do radar no início do século 20 e o resultado de pesquisas realizadas para seu aperfeiçoamento durante a Segunda Guerra Mundial, em 1948 a ideia do RFID foi idealizada por Stockman (1948), mas devido a necessidade de desenvolvimento de outras tecnologias, como: transistores, circuitos integrados, microprocessadores, comunicação de redes e a maneira de como os negócios eram realizados, somente nos anos 60 que atividades comerciais começaram a ser desenvolvidas.

Foi nos anos 70 que o foco de desenvolvedores, instituições acadêmicas e laboratórios governamentais passaram a trabalhar ativamente no RFID e avanços notáveis começaram a surgir. Nesse período, o uso da tecnologia era focado no rastreamento de animais, veículos e automação comercial. Um marco para a tecnologia foi em 1975 quando o artigo de Koelle, Depp e Freyman (1975) sinalizou o início do uso de tags passivas com o alcance de dezenas de metros.

A década de 80 foi de implementação da tecnologia RFID, quando em várias partes do mundo passaram a ter interesse nessa tecnologia para diversas finalidades, como acessos e controle de animais, transporte e trânsito. Tags começaram a ser produzidas em grande escala, seu tamanho foi reduzido e suas funcionalidades melhoradas.

A partir dos anos 90 o RFID foi amplamente utilizado em pedágios eletrônicos e seu sucesso culminou no uso em acessos a estacionamentos, cobranças de tarifas, acessos em condomínios fechados e campus estudantis. A tecnologia continuou a evoluir, expandindo suas funcionalidades.

O século 21 passou a apresentar tags de proporções mínimas que podem ser inseridas em diversos objetos, o uso do RFID passou a ser cada vez mais ubíquo e surgiram aplicações de rastreamento e comércio móvel. No início desse século experiências e pesquisas foram realizadas na área da localização indoor, resultando em alguns trabalhos como os de Hightower J.; Borriello (2000) e o Landmarc de Ni et al. (2004), trabalhos esses considerados clássicos na área de sistemas de localização indoor. Após uma queda na quantidade de pesquisas realizadas com RFID por volta de 2007, um crescimento tem ocorrido nos últimos anos, agora na área da cadeia de suprimentos onde se passou a explorar o uso de RFID com sensores.

Com base nas pesquisas realizadas, apresentamos a Tabela 1, baseada em uma compilação de Landt (2005) que resume a evolução do RFID a partir de sua invenção até os dias atuais.

Tabela 1: Resumo da evolução do RFID

Década	Evento
1940 - 1950	Radar refinado e utilizado, grande esforço de desenvolvimento na Segunda Guerra mundial. RFID inventado em 1948.
1950 - 1960	Primeiras explorações da tecnologia RFID, experimentos de laboratório.
1960 - 1970	Desenvolvimento da teoria do RFID. Início de testes de aplicação em campo.
1970 - 1980	Explosão no desenvolvimento do RFID. Testes aceleram e primeiros usos práticos são realizados.
1980 - 1990	Surgem aplicações comerciais utilizando o RFID
1990 - 2000	Surgimentos de padrões, RFID amplamente utilizado. RFID começa a fazer parte do cotidiano.
2000 - 2010	Vários estudos realizados no uso de RFID em sistemas de localização.
2010 - 2012	Após uma redução na quantidade de pesquisas, ocorre uma explosão na área da cadeia de suprimentos e utilização de sensores.

Fonte: Elaborada pelo autor, adaptada de Landt (2005).

2.2.2 Tecnologia RFID

Segundo Finkenzeller (2010), um sistema RFID é basicamente composto por dois componentes: uma tag, transponder, ou identificador; e uma leitora ou interrogadora. A tag é utilizada nos objetos que se deseja identificar enquanto a leitora realiza a leitura ou escrita e leitura dos dados conforme a tecnologia utilizada. Leitoras tipicamente contém um módulo de rádio frequência transmissora e receptora, uma unidade de controle e um elemento de acoplamento para as tags. As tags são basicamente constituídas por um elemento de acoplamento e um microchip eletrônico. Tags normalmente não possuem suprimento de energia e só estão ativas quando inseridas na zona de interrogação de uma leitora. A unidade de acoplamento, que é *contactless* - sem contato, fornece o suprimento de energia necessário.

O acoplamento é realizado utilizando indução magnética e captura de ondas eletromagnéticas e segundo Want (2006), Roussos e Kostakos (2009) e Finkenzeller (2010), as *tags* RFID podem ser divididas em duas categorias principais que variam conforme o método de transferência de energia entre a leitora e a *tag*. *Tags* que utilizam o componente magnético da onda de rádio são denominadas de *near-field*, enquanto as que usam o componente elétrico são as *far-field*.

Conforme Want (2006), *tags near-field* realizam a comunicação com a leitora utilizando modulação de carga, esta técnica consiste em a leitora captar a energia magnética enviada pela leitora e utilizar essa energia para retornar um sinal alterado através da antena da *tag*; a leitora ao receber o sinal alterado o interpreta como sendo uma mensagem. Há uma distância limite para a utilização do *near-field*, esta distância pode ser conhecida por meio da fórmula:

$$c/2\pi f \tag{2.1}$$

onde:

- c é uma constante referente à velocidade da luz;
- f é a frequência operada pela leitora.

Quanto maior for a frequência de operação, menor vai ser o raio de operação. Outra limitação é a energia disponível para indução, a redução do campo magnético cai em uma proporção de

$$1/r^3,$$

onde r é distância entre a *tag* e a leitora.

As *tags far-field*, por sua vez, utilizam uma técnica chamada de retroespalhamento (backscattering). Esta técnica concentra a antena em uma frequência específica e reflete o sinal recebido nesta antena alterando a impedância da frequência para transmitir dados.

Da mesma maneira que o *near-field*, ocorre um enfraquecimento de energia na medida que a distância entre a *tag* e a leitora vai aumentando, o campo eletromagnético sofre duas atenuações, uma quando o sinal sai da leitora para a *tag* e outra no sentido inverso. A energia no retorno é de cerca de

$$1/r^4,$$

onde r é a distância entre a *tag* e a leitora.

Tags que operam em *far-field* operam em frequências acima de 100MHz, geralmente em UHF. Abaixo dessa faixa de frequência operam as tags RFID baseadas em *near-field*.

Basicamente são dois os tipos de tags RFID mais utilizados, mas um terceiro tipo também é citado na literatura. Conforme Finkenzerler (2010), o primeiro tipo é composto pelas tags RFID passivas, estas tags não possuem bateria e obtêm a energia que necessitam para funcionar do campo magnético ou do campo eletromagnético emitido pelas antenas das leitoras. As tags passivas são normalmente do tipo near-field e só serão ativadas quando estiverem dentro do raio de ação da leitora. O segundo tipo de tag são as tags ativas, elas possuem seu próprio suprimento de energia que normalmente é composto por uma bateria. Esta qualidade permite que o alcance da tag seja maior do que o de uma tag passiva. As tags ativas apresentam algumas vantagens significativas com relação às tags passivas. Como elas possuem uma fonte de energia, é possível adicionar módulos de memória, sensores e até capacidade de processamento. A única desvantagem das tags ativas com relação às passivas é que elas possuem um tempo de vida limitado, por volta de 5 a 10 anos, contra uma capacidade de uso de tempo ilimitado das tags passivas (LI; BECERIK-GERBER, 2011). O terceiro tipo de tag RFID existente é chamado de semi-passivo. Da mesma forma que uma tag ativa, esta tag é dotada de uma fonte de energia, porém só é funcional quando inserida no raio de leitura de uma leitora, de forma idêntica a uma tag passiva.

Existem algumas limitações na tecnologia RFID. Roussos e Kostakos (2009) e Li e Becerik-Gerber (2011) citam que há uma significativa taxa de erros em transmissões de sistemas RFID causada por interferências ou colisões. Interferências nas ondas de rádio podem ser causadas pela proximidade com metais e líquidos, podendo reduzir significativamente o alcance de lei-

Tabela 2: Características de cada frequência

	LF	HF	UHF	Microondas
Raio da frequência	125- 135 kHz	13.56 MHz	400-960 MHz	2.45-5.8GHz
Raio de leitura	<0.5m (pas- sivo)	<1.0m (pas- sivo)	<10m (pas- sivo), >10m (semi-passivo e ativo)	> 100m (ativo)
Padrões	ISO 11784/5, 14223, 18000- 2	ISO 14443, 15693, 18000- 3	ISO 18000- 6/7, EPCGen1 e 2	ISO 18000-4/5
Impacto causado por Metais/Fluidos	Muito Baixo	Baixo	Alto	Alto
Taxa de transferência de dados	Baixo	Médio	Alto	Alto
Energia e transmissão de dados para tags pas- sivas	Acoplamento por indução	Acoplamento por indução	Acoplamento por propaga- ção	N/A
Aplicação na indústria	Agricultura, segurança e cervejaria	Farmácia e saúde	Produção, lo- gística e cons- trução	Exército, na- vegação e ae- ronáutica

Fonte: Adaptado de Li e Becerik-Gerber (2011).

tura e diminuir a taxa de transferência de dados, ou pela existência de fontes de radio frequência externas como outras leitoras e até mesmo uma rede sem fio. Já as colisões ocorrem quando duas ou mais tags tentam responder às leitoras ao mesmo tempo, podendo gerar perda de comunicação, leituras parciais ou fantasmas.

De acordo com Li e Becerik-Gerber (2011), as tags RFID operam em diversos tipos de frequências, elas incluem baixa frequência (LF), alta frequência (HF), ultra-alta frequência (UHF). Super-alta frequência (SHF) e micro-ondas também podem ser utilizadas. Normalmente, UHF é a mais utilizada porque tags passivas UHF fornecem soluções simples e baratas e a maioria das tags ativas operam em UHF. A Tabela 2 apresenta um resumo das características de cada frequência, nela é possível verificar a faixa de frequência em que cada tipo opera, o alcance para cada tipo de tag, padrões ISO de cada uma delas, o impacto causado por metais e fluidos sobre as transmissões, taxa de transferência de dados, método de transmissão de energia e dados para as tags passivas e áreas de uso para cada tipo de frequência.

Alguns tipos de sistemas RFID são apresentados por Finkenzeller (2010), que são denominados conforme a distância entre a leitora e as *tags*. O primeiro tipo citado pelo autor é o sistema de acoplamento próximo (*close-coupling system*). Este tipo de sistema atua sobre um alcance de no máximo 1cm e possui utilidade na realização de operações onde é necessário uma maior segurança. O segundo tipo de sistema apresentado atua sobre um raio de até 1 metro e é conhecido por sistema de acoplamento remoto (*remote coupling system*) ou sistema de rádio indutivo (*inductive radio system*). O último tipo de sistema descrito atua acima de 1 metro e

é conhecido por sistema de longo alcance (*long-range system*), ele utiliza frequências UHF e chega a 3m de alcance com *tags* passivas e 15m com *tags* ativas.

Roussos e Kostakos (2009) citam que o RFID é empregado na cadeia de suprimentos, bilhetagem, controle de ativos, manutenção, varejo e identificação pessoal. Também informam que o RFID é utilizado na área da computação pervasiva servindo como um meio de identificar objetos de maneira universal, auxiliar no gerenciamento de dados, fornecer suporte a sensibilidade ao contexto, entre outras possibilidades.

2.2.3 RFID e Sistemas de Localização Indoor

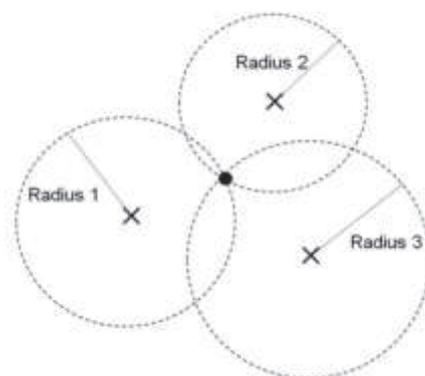
Aqui, é realizada uma apresentação sobre o que é e como funciona um sistema de localização indoor, seguido por uma descrição do papel desempenhado pelo RFID nesse tipo de sistema.

Gu, Lo e Niemegeers (2009) descrevem um Sistema de Localização Indoor (SLI) como um sistema que continuamente e em tempo real pode determinar o posicionamento de algo ou alguém em um espaço físico pertencente a uma construção como uma escola ou hospital.

Existem quatro técnicas para a estimativa de posição em ambientes fechados (GU; LO; NIEMEGEERS, 2009): triangulação, *fingerprint*, análise visual e proximidade.

A triangulação utiliza a distância ou direção conhecida entre a entidade em análise com três elementos de referência que possuam sua localização definida para calcular a posição absoluta dessa entidade (Figura 2).

Figura 2: Técnica de triangulação



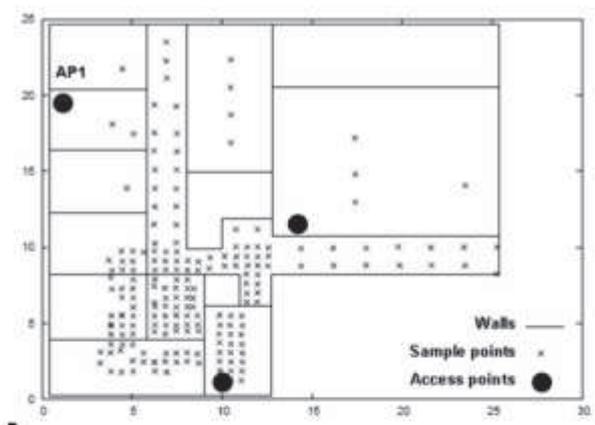
Fonte: Bouet e Santos (2008)

O *fingerprint* é uma técnica que propõe melhorar a precisão da medição do posicionamento indoor utilizando dados já conhecidos do ambiente. Inclui duas fases: treinamento (*offline*) e determinação de posicionamento (*online*) que utiliza o algoritmo *K-nearest-neighbours*, aproximação probabilística, redes neurais, *support vector machine (SVM)* ou *smallest M-vertex polygon (SMP)*.

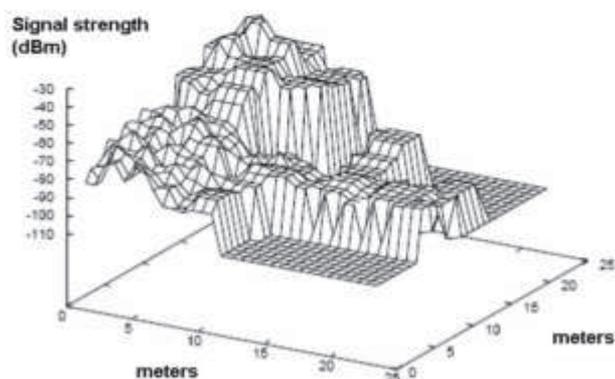
A Figura 3 (a) representa um ambiente que possui três sensores, no modo *offline* foram

marcados pontos que deram origem a um gráfico que representa o mapa de potência do sinal obtido (b).

Figura 3: Técnica de fingerprint



(a) The Experiment Environment



(b) Signal Strength for AP1

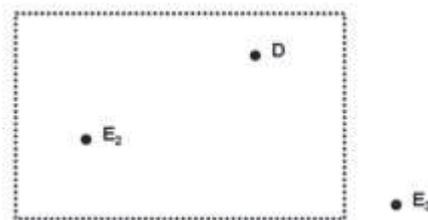
Fonte: Liu et al. (2007)

Análise visual estima a localização a partir de uma imagem obtida por um ou mais pontos. Os alvos são identificados nas imagens e sua posição é estimada a partir de uma base de dados já preparada previamente.

Localização por proximidade compara a localização de um objeto alvo com uma posição ou área conhecida. Se um sensor detectar um objeto alvo, este alvo é considerado na mesma posição do sensor. Caso o alvo seja detectado por mais de um sensor se assume que o alvo está na posição do sensor com o sinal mais forte (BOUET; SANTOS, 2008). Na Figura 4 E^2 e E^3 são alvos rastreáveis e D o sensor. A área demarcada pelos pontos é a reconhecida pelo sensor. Nela é possível verificar que o sensor detectou a aproximação do alvo E^2 .

Triangulação e *fingerprint* podem determinar posicionamento, absoluto, relativo ou aproximado de uma entidade. Já a localização por proximidade apenas posicionamento aproximado.

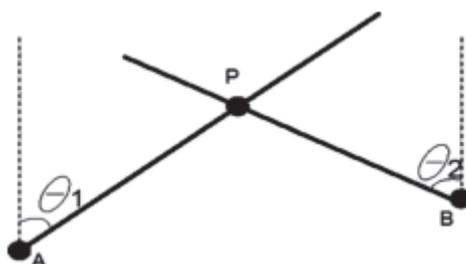
Conforme Liu et al. (2007) as técnicas utilizadas pela triangulação podem ser divididas

Figura 4: Técnica de aproximação

Fonte: Liu et al. (2007)

em dois grupos. No primeiro estão as que estipulam a posição de uma entidade com base na distância, esse grupo é denominado de técnicas de medições de alcance. Nele estão o tempo de chegada (TOA) e a intensidade do sinal recebido (RSS). No segundo grupo estão as técnicas angulares e nele está a técnica de ângulo de chegada (AOA) (Figura 5).

- **TOA:** A distância entre o ponto de referência e o alvo é proporcional ao tempo de propagação do sinal. São necessários três referências para definir o posicionamento em 2D e também que todos os emissores e receptores estejam sincronizados e utilizando time-tamps para facilitar a avaliação das distâncias.
- **RSS:** Também necessita de três pontos de referência para determinar o posicionamento de uma entidade. Estima o posicionamento utilizando a atenuação da potência do sinal emitido.
- **AOA:** Necessita apenas de duas referências para determinar o posicionamento 2D de uma entidade ou três referências para o 3D. Determina o posicionamento de uma entidade através da intersecção de duas retas, cada uma pertencendo a um receptor distinto.

Figura 5: Técnica de AOA

Fonte: Liu et al. (2007)

Devido ao baixo custo de implementação, o RFID se tornou popular no desenvolvimento de sistemas de localização indoor. Ele é amplamente utilizado tanto como única tecnologia empregada quanto em conjunto com outras, atuando como um recurso de apoio para a melhoria na definição da precisão de um objeto nesse tipo de sistema. Quando utilizado em conjunto com

outros tipos de tecnologias, como sensores, o RFID assume o papel como um dos elementos que auxiliam na definição de contextos. Esse tema será discutido separadamente logo a seguir sob o título de RFID e Sensibilidade ao Contexto.

2.2.4 RFID e Sensibilidade ao Contexto

Conforme Roussos e Kostakos (2009), o papel do RFID na sensibilidade ao contexto é como uma ferramenta de suporte para identificar entidades próximas e o sistema fornecer serviços. Um exemplo disso é o emprego do RFID em casas inteligentes onde, ao haver aproximação de um determinado item, o ambiente pode reagir de uma maneira específica.

O RFID é efetivo em identificar a presença de entidades marcadas, mas não pode fornecer nenhuma informação sobre a sua situação. Uma solução para essa condição é incorporar sensores às *tags* RFID. Infelizmente, os sensores necessitam de uma fonte de energia e embora *tags* RFID ativas possam suprir essa necessidade, não é o que ocorre com *tags* passivas.

3 TRABALHOS RELACIONADOS

Na literatura científica, existe uma ampla variedade de publicações que abordam o uso de RFID e contexto para diversas finalidades. Durante as pesquisas realizadas não foram encontrados trabalhos que fossem diretamente ao encontro da proposta desta dissertação, o de gerar modelagem de contexto baseado em informações fornecidas por *tags* RFID com sensores. Desta forma, foram selecionados trabalhos que apresentassem alguma relevância para o objetivo proposto e divididos em dois grupos. O primeiro grupo apresenta uma relação de trabalhos que tratam informações obtidas por meio de RFID, realizando modelagem de contexto resultando em informações semânticas. O segundo grupo descreve trabalhos que exploram o uso de sensores em conjunto com RFID para adquirir informações que possam exercer alguma influência sobre entidades que estejam sendo monitoradas.

3.1 Grupo 1: Modelagem de contexto baseado em RFID

Este grupo de trabalhos foca na modelagem de contexto com base em informações obtidas por meio de RFID. Os modelos aqui apresentados estão concentrados em como os dados recebidos a partir das *tags* RFID são tratados e processados para que possam gerar informação contextual e serem utilizadas. Estes trabalhos foram selecionados porque procuram realizar a extração de informações semânticas dos dados obtidos.

3.1.1 *A RFID-based Context-Aware Service Model*

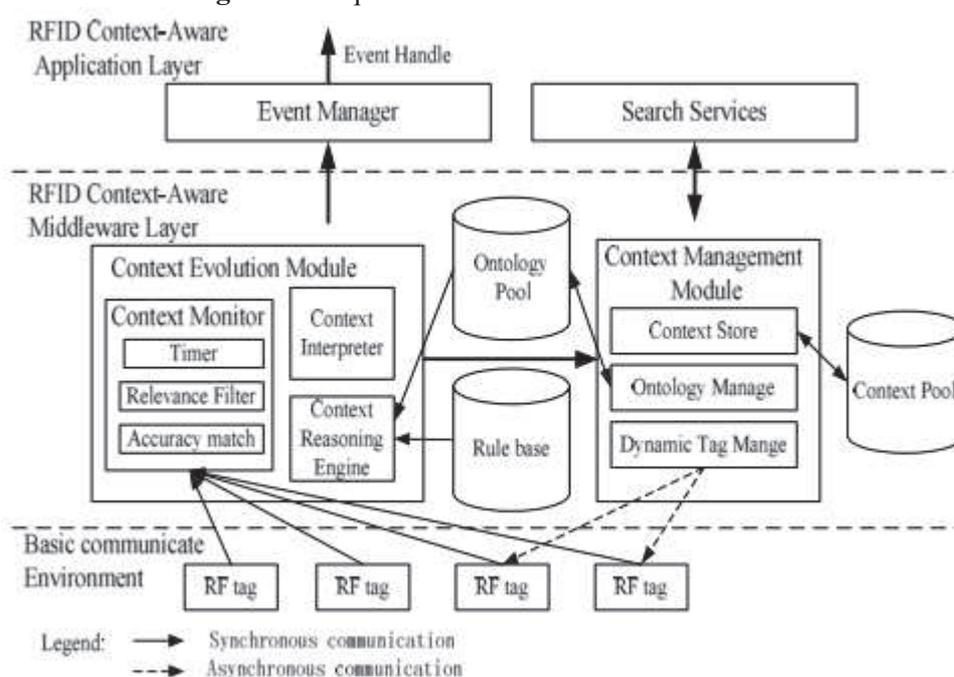
Sheng et al. (2011) propõem um modelo de framework para serviços sensíveis ao contexto baseado em RFID. Para atingir o objetivo traçado, os autores combinaram na modelagem de contexto ontologia SOCAM (GU; PUNG; ZHANG, 2004), um modelo formal de ontologia baseado em OWL (WANG et al., 2004), como FOAF¹ (*Friend of a Friend*) e CC/PP (*Composite Capabilities/Preference Profiles*) (KIM; LEE, 2006). O modelo apresentado utiliza ontologia para descrever entidades computacionais no ambiente pervasivo, enquanto usa FOAF para descrever a relação entre cada usuário do sistema e CCPP para descreve as escolhas do usuário, a capacidade de compreensão e desempenho do equipamento no ambiente computacional.

Na modelagem de contexto, as informações são inferidas por meio de uma combinação de OWL² e uma base de regras. Essas informações são divididas em quatro categorias: entidade computacional, entidade de usuário, entidade de localização e entidade de atividade. Essas quatro categorias são ainda divididas de acordo com os seus domínios e pode existir alguma relação entre elas. A representação do contexto é realizada por meio de uma tupla de seis elementos (*EntitySubject, predicate, EntityObject, time, area, visits*). *EntitySubject* indica o corpo principal do contexto; *EntityObject* aponta o objeto do contexto; *predicate* dá a relação semântica entre as entidades *EntitySubject* e *EntityObject*; *time* atribui o momento em que o

contexto foi gerado; *area* informa a localização do contexto e, por fim, *visits* denota o registro de ocorrências desse contexto na localização especificada.

O modelo segue uma arquitetura de três camadas (Figura 6). A *RFID Context-Aware Application Layer* realiza os tratamentos necessários nas mensagens recebidas dos RFIDs para a camada de *Middleware* ou utiliza a camada de *Middleware* para oferecer um serviço de resumo em alto nível. A *Middleware Layer* obtém as informações de contexto da *Basic Communication Environment*. Além disso, ela pode detectar mudanças nessas informações e fornecer acesso a elas ou informá-las para as aplicações sensíveis ao contexto. Essa camada também é responsável por manter as informações relevantes aos usuários. Ela é dividida em dois módulos: *Context Evolution Module* e *Context Management Module*. O *Context Evolution Module* é responsável por capturar e fornecer as alterações nas informações de contexto enquanto o *Context Management Module* é responsável por gerenciar os conjuntos de contextos, visando facilitar o uso futuro. A terceira camada, a *Basic Communication Environment*, é composta por *tags* RFID, leitoras, e outros sensores, e é responsável por fornecer as informações de contexto para o restante do modelo. As *tags* RFID utilizam EPC para codificar as suas informações e garantir a sua legibilidade.

Figura 6: Arquitetura em três camadas do modelo



Fonte: Sheng et al. (2011)

O exemplo utilizado por Sheng et al. (2011) para avaliar o seu modelo foi a aplicação na área de mineração, onde foi realizado um acompanhamento das condições físicas dos trabalhadores. Um funcionário teve sua pressão sanguínea e pulso monitorados. Se a pressão ou batimentos

¹The Friend Of A Friend (FOAF) Project: <http://www.foaf-project.org/> acessado em 6/10/2012

²<http://w3.org/TR/2004/RDC-owl-features-20040210/>

cardíacos chegassem ao limite especificado no sistema, a *tag* RFID localizada no trabalhador teria o seu estado atualizado para que quando o sistema lesse essa *tag*, ele pudesse enviar um alerta recomendando que este o funcionário não continuasse a trabalhar.

3.1.2 *A Context-Aware Smart Infrastructure based on RFID Sensor-tags and its Application to the Health-Care Domain*

No trabalho de Catarinucci et al. (2009), o objetivo é a elaboração e desenvolvimento de um hardware e de um software para a aquisição de dados por meio de monitoramento sensorial. O hardware, que foi denominado *S-Tag*, consiste em uma *tag* RFID passiva que pode ser acoplada a um sensor genérico. A *tag* RFID informa quando requisitados os parâmetros físicos medidos pelo sensor.

O software, por sua vez, é constituído por uma arquitetura de três camadas (Figura 7). A camada mais baixa é a *Context Sources* a qual está subdividida em dois grupos: sensores físicos e sensores virtuais. Os sensores físicos identificam dispositivos que podem adquirir informações de contexto, tal como RFID e sistemas de posicionamento. Sensores virtuais são constituídos por bancos de dados e interfaces para entrada de dados dos usuários.

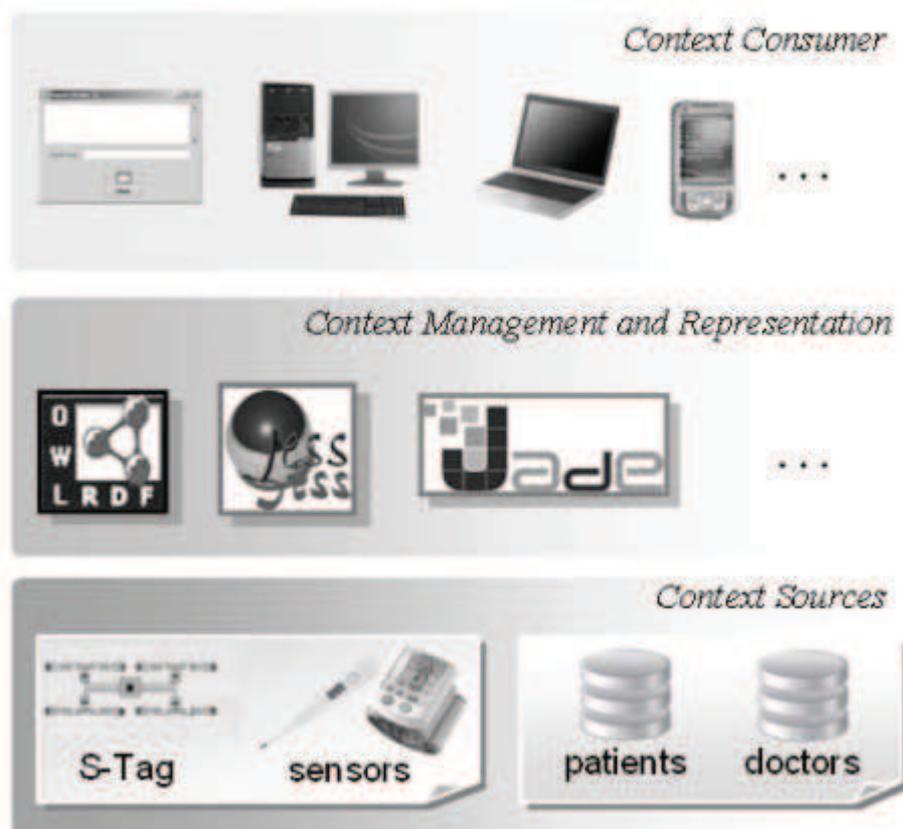
Acima da camada de *Context Sources* está a *Context Management and Representation*. Esta camada possui os recursos para o processamento das informações de contexto adquiridas. Para isso, realiza a integração de três tecnologias: ontologia, para a formalização da informação contextual em uma codificação reutilizável e compartilhável; paradigma multiagente, que organiza o sistema como uma interação coordenada de entidades de raciocínio autônomo; e lógica baseada em regras, que suporta agentes na implementação de raciocínio avançado e na derivação de conceitos de alto nível a partir da informação sentida.

A ontologia utilizada foi desenvolvida com OWL, a base de regras foi implementada com *Jess* (FRIEDMAN-HILL, 2003) e os agentes por meio de *Java Agent Development Environment* (JADE)³. Segundo os autores, a informação contextual recebida pelos sensores, denominada de baixo nível, é utilizada pelo sistema para inferir uma informação contextual de alto nível também denominada "evento". Aplicando regras lógicas e convertendo os eventos em acontecimentos, é possível realizar um evento maior, como, por exemplo, executar um alarme do sistema.

O exemplo de aplicação utilizado pelos autores é um caso de uso em um domínio de cuidados de saúde. No cenário explorado, diversos pacientes estão sendo monitorados por sensores integrados com o *S-Tag*. Se as informações adquiridas pelos sensores, combinados com as informações existentes dos pacientes, gerar um evento que atenda uma regra do sistema, por exemplo, temperatura ou glicose alta, um alarme é gerado alertando o médico mais indicado para atender a situação identificada.

³Jade web site: <http://jade.cselt.it> acessado em 6/10/2012

Figura 7: Arquitetura em três camadas utilizado pelo modelo



Fonte: Catarinucci et al. (2009)

3.1.3 An Extended Context Model in a RFID-based Context-Aware Service System

O objetivo do trabalho de Zhang, Zheng e Liu (2008) é propor um modelo formal de contexto em ambientes ubíquos. Para alcançar o objetivo traçado, os autores utilizaram o OWL-S⁴, uma versão de OWL para serviços, para descrever serviços WEB semânticos.

O modelo de serviço sensível ao contexto proposto procura combinar a abordagem centrada no usuário, em que as preferências do usuário são utilizadas para gerar serviços, e a abordagem centrada em serviço, onde os serviços oferecidos são adaptados conforme o contexto é alterado.

Os dados adquiridos pelo OWL-S são complementados com informação contextual. Essa informação é decomposta em atributos de contexto de alto nível, e precondições contextuais e efeitos. Os atributos de contexto indicam se o serviço é sensível à informações de contexto, enquanto as condições contextuais indicam as condições que devem ser cumpridas para que o serviço seja executado.

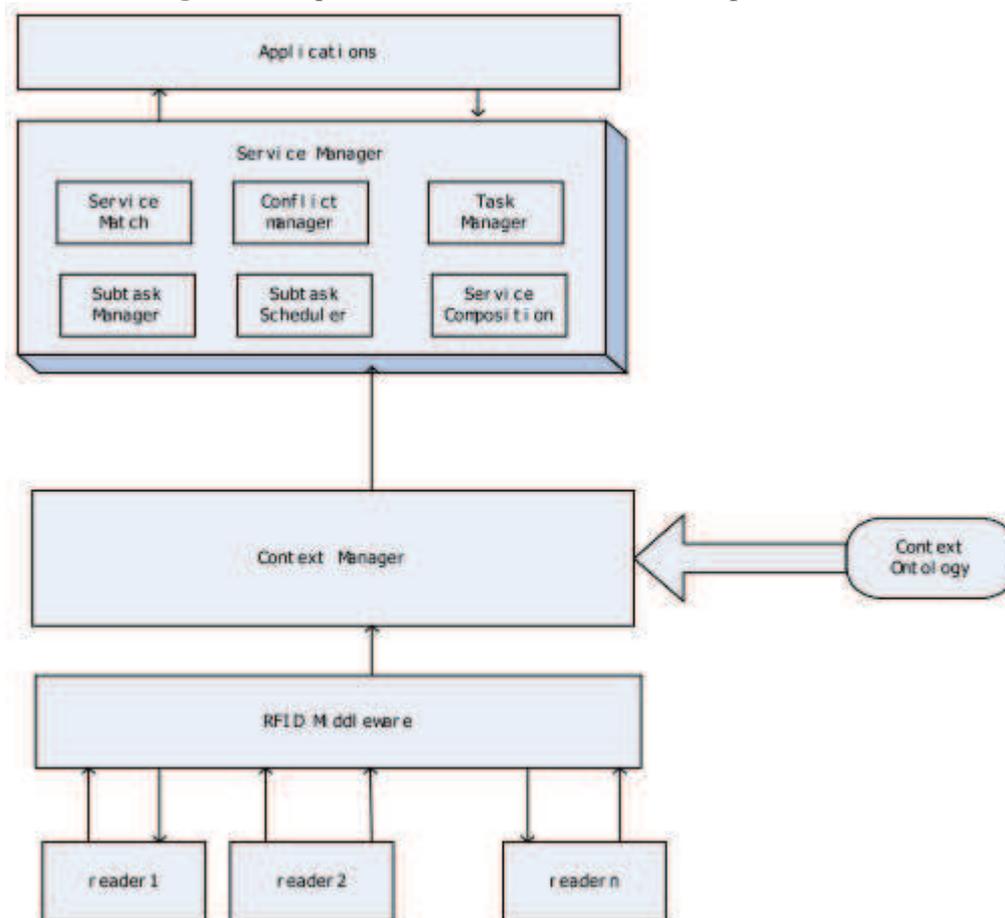
Os autores especificam um sistema de serviço de contexto baseado em RFID (Figura 8). A camada de *middleware RFID* envia os dados coletados pelas leitoras RFID para a camada *Context Manager*. Entre os dados obtidos podem estar diversas informações contextuais obtidas

⁴<http://www.w3.org/Submission/OWL-S/> acessado em 6/10/2012

juntamente com as leitoras RFID. A camada *Context Manager* coleta informações de diferentes entidades que afetem o contexto. Ela utiliza ontologia de contexto para fornecer informação contextual. A camada de *Service Manager* foi projetada para administrar diversos serviços. Ela possui três módulos para gerenciar as três camadas de serviços, encontrar conflitos de serviços e resolver problemas de conflitos.

Este trabalho não apresentou uma avaliação pois tratava de um projeto em desenvolvimento.

Figura 8: Arquitetura em três camadas utilizado pelo modelo



Fonte: Zhang, Zheng e Liu (2008)

3.1.4 Discussão sobre os trabalhos do grupo 1

Como já citado anteriormente, os trabalhos apresentados neste grupo um estão focados no tratamento realizado da informação adquirida de uma ou mais *tags* RFID com o objetivo de gerar informação contextual. A Tabela 3 apresenta uma compilação das características destacadas dos trabalhos apresentados. Nesta tabela, são exibidas informações relacionadas ao tipo de ontologia utilizada, qual o tipo de tecnologia RFID empregada, se a tecnologia RFID utiliza o padrão EPC, quais os tipos de sensores suportados, qual a arquitetura utilizada em suas propostas, o tipo de modelagem de contexto empregada e qual o meio de representação de contexto

Tabela 3: Trabalhos que geram contexto a partir de informações obtidas por RFID

Trabalho	Sheng et. al (2011)	Catarinucci et. al (2009)	Zhang, Zheng, Liu (2008)
Tipo de RFID utilizado	Não específica	Passiva	Não específica
Padrão EPC	Sim	Não	Não
Tipo de Sensores	Variados	Variados	Não específica
Arquitetura	3 camadas	3 camadas	SOA
Modelagem de Contexto	Ontologia/Baseada em regras	Ontologia/Baseada em regras/Multi Agente	Ontologia
Representação de Contexto	Tuplas		
Formato de Representação da Ontologia	SOCAM/OWL	OWL	OWL-S

Fonte: Elaborada pelo autor.

utilizado.

Os três trabalhos utilizam o formato OWL para representar a ontologia, sendo que Zhang, Zheng e Liu (2008) utilizam OWL-S, uma versão de OWL voltada a geração de serviços. Apenas Catarinucci et al. (2009) especificam o tipo de *tag* RFID usados nas validações, os demais trabalhos apenas especificam de forma genérica que trabalha com *tags* RFID. Sheng et al. (2011) foram os únicos que procuraram projetar o seu modelo utilizando o padrão EPC. Sheng et al. (2011) e Catarinucci et al. (2009) indicam que seus modelos suportam tipos variados de sensores e também descrevem suas arquiteturas em 3 camadas. Já Zhang, Zheng e Liu (2008) não informaram sobre o uso de sensores em seu modelo e descreveram a sua arquitetura orientada a serviços (SOA).

Os três trabalhos fazem a modelagem de contexto Orientada à Ontologia, com a diferença que Sheng et al. (2011) utilizam sua modelagem baseada em regras, e Catarinucci et al. (2009), além de usarem modelagem baseada em regras, também utilizam multiagentes. Por fim, Catarinucci et al. (2009) e Zhang, Zheng e Liu (2008) representam contexto por meio de ontologias, enquanto Sheng et al. (2011) utilizam tuplas. Neste grupo, os trabalhos apresentados procuram extrair informações semânticas a partir de dados provenientes de *tags* RFID. Exceto por Catarinucci et al. (2009), estes trabalhos não abordam os tipos de RFID utilizados, mesmo especificando-os em seus modelos.

3.2 Grupo 2: Exploração de dados obtidos por meio de RFID com sensores

Neste grupo, os trabalhos apresentados procuram explorar o uso de sensores combinados com RFID para monitorar a condição de entidades em situações específicas. A escolha desses trabalhos foi por serem primeiramente baseados em RFID e por abordarem a questão do

monitoramento e controle como vitais para o sucesso de seus modelos.

3.2.1 *The Promise of RFID-Based Sensors in the Perishables Supply Chain*

Delen, Sharda e Hardgrave (2011) abordam a combinação do RFID com sensores e sua aplicação em objetos ambientalmente sensíveis, por exemplo, produtos farmacêuticos, perecíveis e matérias-primas. O objetivo do artigo é poder informar não só a localização de um produto, mas também qual a situação em que ele se encontra. No artigo, os autores focam no uso específico de RFID com sensores de temperatura.

O artigo apresenta um sistema com arquitetura em três camadas (Figura 9). Duas delas são consideradas como principais; a camada de *front-end* e a camada de *back-end*. A terceira camada é denominada *gateway* e faz a ligação entre as duas.

Na camada *front-end* estão localizados os sensores baseados em RFID, leitoras e os mecanismos de armazenamento de dados temporários. Os sensores baseado em RFID basicamente são compostos por *tags* RFID semi-passivas e utilizam um método de aquisição de dados por meio de um intervalo de tempo específico chamado de *Periodic polling*. Este método é utilizado pelos sensores para reduzir o consumo de energia da bateria.

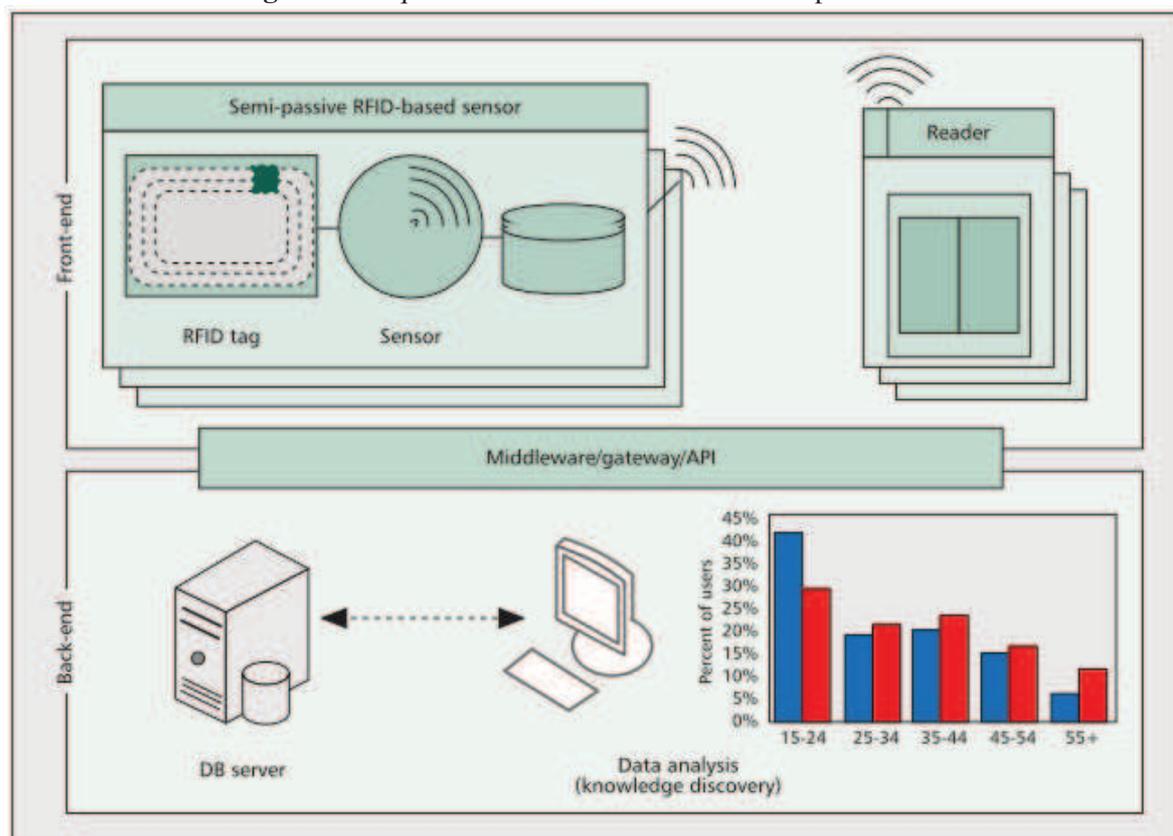
A camada de *back-end* é composta por um banco de dados e diversos módulos de software de análise de negócios. Os dados do banco de dados são tratados e organizados na forma de séries temporais. E os módulos processam os dados e procuram extrair informações e conhecimento que tenham valor para o negócio.

A camada de Gateway realiza a ligação entre as camadas de *front-end* e *back-end*. Além disso, permite que sejam feitas troca de componentes em ambas as camadas sem que seja necessário reconfigurar a camada de Gateway.

Com o intuito de testar a utilidade dos sensores baseados em RFID, os autores realizaram dois testes de campo. No primeiro teste, os sensores com RFID foram colocados em todas as caixas de um carregamento de banana. Esse carregamento viajou da America do Sul para os Estados Unidos, sendo a temperatura da carga monitorada no decorrer da viagem. Durante o trajeto percorrido até um depósito no Arkansas - EUA, seu destino final, a carga foi transportada por caminhão, navio e trem. Como resultado do monitoramento realizado, um perfil da temperatura da carga foi traçado e também foi possível verificar que o container onde as caixas viajaram não possuía um fluxo de ar uniforme, resultando em temperaturas mais elevadas em alguns pontos do container e, conseqüentemente, em algumas das caixas de bananas.

O segundo teste envolveu monitorar um carregamento de vegetais durante quatro dias em uma viagem de costa a costa, nos Estados Unidos. *tags* de temperatura foram colocadas em cada caixa a ser transportada e um mapeamento da temperatura do interior do container foi realizado. O perfil traçado permitiu concluir que o ar resfriado não abrangia o container da forma esperada, ficando concentrado na parte superior e não chegando ao fundo, resultando em uma maior variação de temperatura nas caixas na parte de baixo e central do container.

Figura 9: Arquitetura em três camadas utilizado pelo modelo



Fonte: Delen, Sharda e Hardgrave (2011)

Após uma análise dos testes realizados, os autores concluíram que é possível conhecer as condições dos produtos transportados com base nos dados adquiridos pelos sensores.

3.2.2 A simulation approach for optimal design of RFID sensor tag-based cold chain system

No artigo de Kang et al. (2012) os autores procuram aplicar o uso de RFID com sensores em um sistema de cadeia fria, do inglês *Cold Chains System* (CCS), onde há necessidade não apenas de saber quando e onde um produto está, mas também o quanto o ambiente está lhe influenciando. Para alcançar os seus objetivos, os autores desenvolveram um *Smart Cold Chains System* (SCCS), um simulador baseado no DEVS (ZEIGLER B.P., 2000) - O DEVS é um formalismo que permite modelar e analisar sistemas de eventos discretos, ele representa cada componente do sistema como um modelo atômico e descreve os relacionamentos e estruturas hierárquicas entre eles utilizando um modelo acoplado.

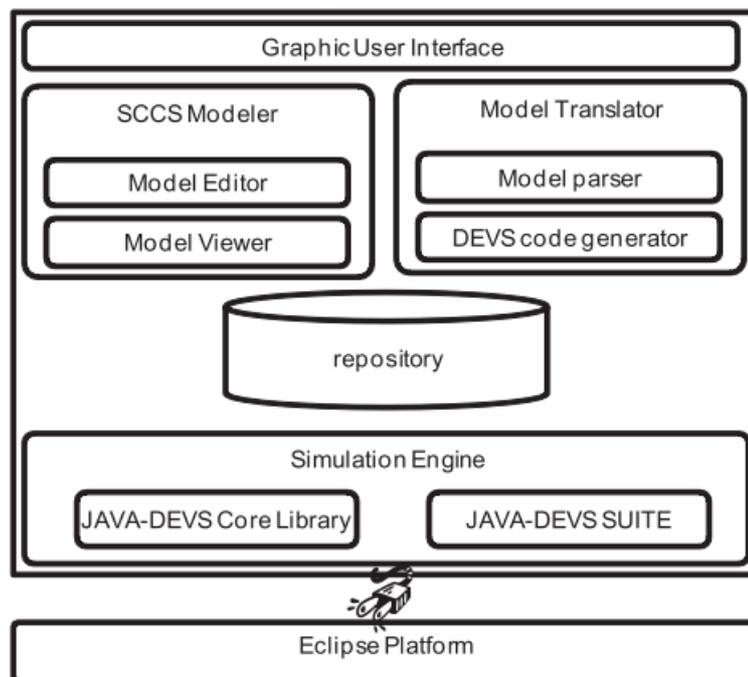
Cinco modelos atômicos foram elaborados para o desenvolvimento do simulador utilizando DEVSJAVA (ACIMS, 2009): *Sensor_Tag_Node* corresponde a *tag* sensor RFID; *Carrier_Node* representa qualquer local de armazenamento ou transporte; *Reader_Node* diz respeito às leitoras RFID e *Middleware*; *Database_Node* refere-se a uma base de armazenamento externa; e

Start_End_Node indica quando uma simulação deve começar ou encerrar.

O processo de coleta e armazenamento de dados planejado pelos autores é feito por um sensor que avalia cada pequena alteração no ambiente e armazena essa informação no espaço de memória da *tag* sensor RFID. A leitora, ao obter a informação da *tag*, envia os dados para um sistema de informação. As *tags* utilizam EPC e os dados são enviados por meio de uma rede CDMA.

O simulador sugerido por Kang et al. (2012) foi desenvolvido através do *Eclipse Rich Client Platform* (RCP), utilizando Java e *Eclipse Graphical Modeling Framework* (GMF), e é composto por um modelador SCCS, um tradutor modelo, um motor de simulação e um modelo de repositório distribuídos em três camadas. A Figura 10 ilustra a estrutura do simulador.

Figura 10: Arquitetura em três camadas



Fonte: Kang et al. (2012)

O modelador SCCS apresenta um conjunto de recursos gráficos para facilitar a elaboração da CCS, permitindo arrastar componentes e configurá-los com um simples duplo-clique de mouse. O tradutor modelo traduz as conexões entre os componentes para códigos Java ou modelos acoplados. O repositório serve para armazenar os resultados das simulações. O motor de simulação é responsável por executar os testes.

O simulador foi avaliado com o auxílio de uma empresa logística da Coreia do Sul. O objetivo do teste era determinar o melhor "intervalo sensível" em que os sensores de temperatura deveriam adquirir os dados. O alvo dos testes do sistema foi o monitoramento de temperatura para comidas congeladas. Sensores RFID foram colocados nas pallets enquanto *tags* passivas foram colocadas em cada caixa de comida. As informações eram lidas periodicamente quando as caixas estavam estocadas ou em seções de transporte. As informações coletadas eram trans-

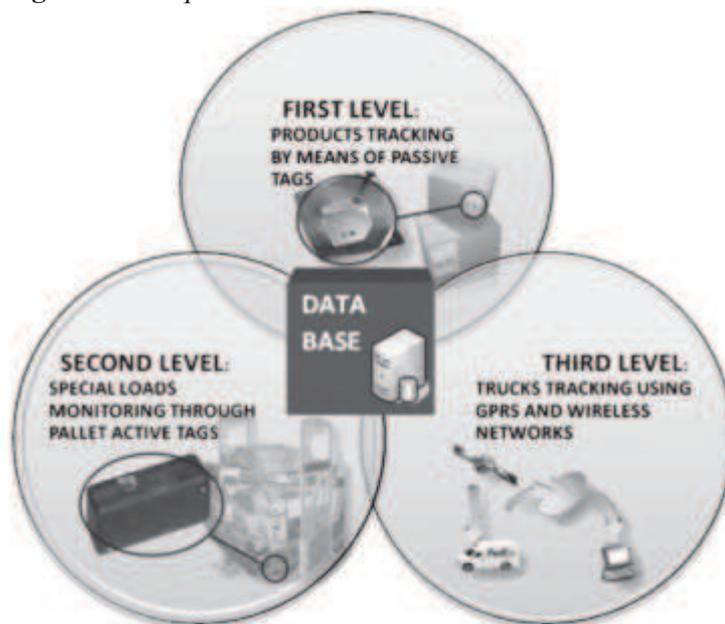
mitidas para o sistema por meio de dispositivo móvel com conexão CDMA equipados nos caminhões. Os dados lidos eram ID e temperatura. Durante uma viagem, o intervalo de envio era de cerca de 10 minutos ou quando a temperatura chegava a uma situação de limite onde era enviada imediatamente. Um perfil da temperatura da carga era traçado conforme o progresso da viagem, registrando as alterações durante o percurso.

O monitoramento permitiu encontrar situações nas quais os alimentos foram expostos a temperaturas além do limite estabelecido: operações de carregamento/descarregamento da carga, quando a temperatura ambiente estava muito alta, refrigeração quebrada, e também quando havia uma fonte de calor no caminhão. Também foram detectados que motoristas desligavam o ar condicionado durante o trajeto percorrido para economizar combustível.

3.2.3 *Improvement in the tracking of special loads by using a three-level RFID system*

O trabalho de Ga-Escribano et al. (2012) apresentam um sistema de três camadas (Figura 11), as quais foram denominadas de níveis, que utiliza a combinação de RFID ativo e passivo para determinar a localização e as condições, tanto de conjuntos de produtos, quanto de produtos individuais, em todas as etapas da cadeia de suprimentos.

Figura 11: Arquitetura em três camadas utilizado no modelo



Fonte: Ga-Escribano et al. (2012)

O primeiro nível é constituído por itens que podem ser produtos ou caixas. Cada um desses itens possui uma *tag* RFID passiva e a função dessas *tags* é informar a sua identificação para as antenas. O segundo nível é composto por pallets de caixas ou produtos que podem ser compostas de forma não homogênea, por tipos diversificados. *tags* RFID ativas estão presentes nas pallets, e sua função é armazenar todas as informações das *tags* passivas neles contidos. O

terceiro nível é constituído pelos containers capazes de transportar as pallets de produtos. Nestes containers, leitoras gerenciam a informação do seu conteúdo, lendo ou escrevendo informações nas *tags* ativas. A comunicação entre as leitoras é realizada por meio de uma rede ZigBee (ALLIANCE, 2008). Cada leitora pode estar em um container e um coordenador desta rede pode atualizar informações das unidades da rede. Este coordenador possui conexão GPRS e é capaz de enviar informações em tempo real mantendo disponíveis as informações sobre os carregamentos e a possibilidade de determinar o posicionamento da carga continuamente.

O sistema foi implementado utilizando *tags* EPC e adicionando sensores às *tags* ativas contidas nas pallets para adquirir informações das condições ambientais tais como temperatura e umidade. Ele foi testado utilizando dois cenários de rastreamento de produtos.

O primeiro teste foi realizado em uma empresa de engarrafamento espanhola para rastreamento de cargas especiais. A empresa obteve um crescimento de demanda em poucos anos e passou a ter de realizar entregas de vinhos e sucos de frutas em pallets com conteúdo misto, necessitando ter melhor controle sobre as suas cargas. A implementação do sistema resultou em uma queda na quantidade de erros nos pedidos, alcançando 99,5% de eficácia. Também resultou em uma maior agilidade no processamento de pedidos e reduziu custos de produção e distribuição. Os sensores instalados nas *tags* ativas se provaram úteis para a preservação das propriedades dos produtos transportados.

O segundo teste foi aplicado na produção de painéis de concreto pré-fabricados. Uma drástica queda na quantidade de erros na identificação e perda dos painéis foi obtida como resultado da implementação do sistema. Além disso, houve um ganho na rapidez na busca de painéis, o que tornou flexível e ágil o processo de despacho das peças.

Ga-Escribano et al. (2012) concluem que o sistema proposto oferece uma boa confiança e controle de qualidade, também oferece a habilidade de detectar erros em qualquer ponto da cadeia. O sistema apresentou uma melhora no número de erros em ambos os testes aplicados, reduzindo as taxas, que variavam entre 20 e 40%, para menos de 5%, em ambos os casos.

3.2.4 Discussão sobre os trabalhos do grupo 2

Os trabalhos apresentados neste grupo abordaram a utilização de sensores com RFID para obter as condições de entidades em determinados momentos. A tabela 4 compila as características em destaque dos trabalhos deste grupo, acentuando aquelas que mais interessam para o desenvolvimento deste trabalho. É citado nesta tabela qual o tipo de RFID empregado, se as *tags* usam o padrão EPC, os tipos de sensores utilizados nas avaliações dos modelos, a arquitetura empregada no desenvolvimento dos sistemas, se utilizam perfil em algum ponto do sistema, qual o método de captura de dados empregado, se há geração de alarmes de monitoramento e qual a função de armazenamento do RFID.

Delen, Sharda e Hardgrave (2011) propõem o uso de *tags* RFID semi-passivas para trabalhar com os sensores, enquanto Kang et al. (2012) e Ga-Escribano et al. (2012) optaram pelo uso de

Tabela 4: Trabalhos que utilizam RFID com sensores

Trabalho	Delen et. al (2011)	Kang et. al (2012)	Escribano et. al (2012)
Tipo de RFID utilizado	Semi-Passivo	Passivo / Ativo	Passivo / Ativo
Padrão EPC	Sim	Sim	Sim
Tipo de Sensores	Temperatura	Temperatura	Temperatura / Umidade
Arquitetura	3 camadas	3 camadas	3 camadas
Uso de perfil	Temperatura	Temperatura	Não utiliza
Método de coleta de dados	Periodoc Pooling	Periodoc Pooling	
Alarmes de monitoramento	Sim	Não	Não
Função do Armazenamento no RFID	Registrar Valores de temperatura (quantidade não informada)	Registrar valores de temperatura (Até 64 registros)	<i>tags</i> ativas armazenam dados das cargas

Fonte: Elaborada pelo autor.

tags ativas para trabalhar com os sensores e o uso de *tags* passivas para auxiliar no controle das entidades envolvidas em suas avaliações. Ga-Escribano et al. (2012) justificam o uso de *tags* ativas em vez de *tags* semi-passivas, devido a necessidade de que as semi-passivas precisam estar dentro da área de alcance de uma leitora para poder utilizar os sensores.

Todos os trabalhos apresentados utilizam o padrão EPC para identificar as entidades envolvidas e todos eles exploraram a utilização de sensores de temperatura em suas avaliações, exceto por Ga-Escribano et al. (2012) que utilizaram também sensores de umidade. Os três trabalhos também utilizaram em seus projetos arquiteturas em três camadas, com exceção de Ga-Escribano et al. (2012) que utilizaram além das três camadas um banco de dados separado.

Os trabalhos de Delen, Sharda e Hardgrave (2011) e Kang et al. (2012) realizam a criação e gerenciamento de um perfil da temperatura do ambiente que está sendo monitorado para fazer uma avaliação sobre as condições das entidades que estão sendo transportadas.

Delen, Sharda e Hardgrave (2011) e Kang et al. (2012) fazem as coletas de informações dos sensores ou *tags* RFID por meio de um método denominado *Periódic Pooling*, que obtém os dados em um intervalo de tempo pré-determinado. Ga-Escribano et al. (2012) não especificou o procedimento de aquisição de dados.

O trabalho de Delen, Sharda e Hardgrave (2011) é o único que faz uma análise dos dados coletados e em caso de necessidade gera um alarme de monitoramento quando eles não estão consistentes.

Cada um dos trabalhos utiliza a capacidade de armazenamento das *tags* RFID de uma forma diferente. Os mais similares foram Delen, Sharda e Hardgrave (2011) e Kang et al. (2012), ambos os trabalhos utilizam as *tags* para manter armazenados os valores das temperaturas obtidas pelos sensores. O diferencial é que Delen, Sharda e Hardgrave (2011) informam que todos

os registros são armazenados até o limite de 64 registros que é o limite da *tag* enquanto Kang et al. (2012) não especificam a quantidade de registros nem o critério para armazenamento. Ga-Escribano et al. (2012) utilizam as *tags* ativas do modelo para armazenar as informações das cargas.

Esses três trabalhos apresentados são frutos de pesquisas realizadas para a área de cadeia de suprimentos onde as condições dos produtos transportados devem ser monitoradas para que haja uma maior qualidade. Por esse motivo, o foco destes trabalhos está apenas na obtenção dos dados, não fazendo menção a posteriores tratamentos.

3.3 Considerações sobre os trabalhos relacionados

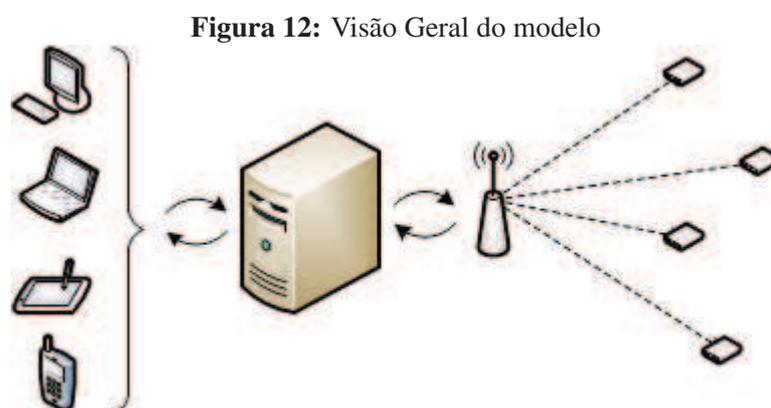
Seis trabalhos foram destacados, três que fazem uso de dados provenientes de *tags* RFID para gerar contexto, e três que utilizam RFID com sensores para obter informações relevantes a entidades monitoradas. Através desses trabalhos, foi possível observar que ontologias são o meio mais utilizado para realizar a geração e representação de contexto por meio do uso de OWL. Também foi possível notar que a arquitetura mais empregada é a de três camadas, sendo basicamente uma onde há a coleta de dados por meio das leitoras e *tags* RFID, outra para processamento dos dados obtidos, onde é extraído conhecimento, e a terceira para fornecimento de serviços. Apesar de terem explorado, em sua maioria, sensores de temperatura, alguns dos trabalhos citam a possibilidade de utilizar outros tipos de sensores, permitindo ter uma maior variedade no fornecimento de dados para obter um contexto mais preciso no monitoramento das condições de entidades. O uso de perfil ocorreu conforme a preferência e a necessidade encontrada pelos autores de cada trabalho. Houve trabalhos em que foi considerado manter um histórico da variação da temperatura no decorrer de uma viagem, já em outro se considerou mais importante possuir as informações referentes ao que estava sendo transportado.

No próximo capítulo, estaremos apresentando um modelo completo para a geração de contexto a partir de RFID combinados com sensores, com base nas informações obtidas ao analisar os trabalhos relacionados. Este modelo surge a partir da não existência de um modelo que permita a fácil integração e extração de informações em projetos de diversas áreas, onde, conforme os trabalhos estudados há normalmente a necessidade de elaborar uma solução que normalmente resolve um problema pontual.

4 ONISCIENTE

Após a seleção e estudo dos trabalhos relacionados, este capítulo possui como finalidade apresentar a especificação de um modelo que faça uso das tecnologias abordadas na criação de contexto. No decorrer do capítulo será apresentada uma visão geral do modelo, seguido de seus requisitos funcionais e os casos de uso envolvidos. Logo em seguida, será descrita a especificação da arquitetura dos elementos que compõem o modelo, junto com alguns diagramas de atividade que demonstram o procedimento que deve ser realizado para resolver cada caso de uso especificado.

4.1 Visão geral do modelo



Fonte: Elaborada pelo autor.

O Onisciente deverá utilizar dados adquiridos por meio de *tags* RFID para criar informações contextuais que informem as condições de entidades que estejam sendo monitoradas. A Figura 12 apresenta uma visão geral dessa ideia. Usuários poderão cadastrar as entidades que desejam monitorar por meio de uma interface fornecida pelo cliente. Neste cadastro de entidade os usuários especificarão um perfil, que abrange todas as características da entidade e onde estarão presentes as limitações às quais ela pode ser exposta. Por exemplo, uma carga que possui uma faixa de temperatura máxima e mínima na qual deve ser mantida, ou um paciente em um hospital que não possa ultrapassar um determinado valor na sua taxa de glicose.

Um recurso oferecido pelo Onisciente é o de realizar acompanhamento de entidades por meio de sessões. Uma sessão é um espaço controlado pelo servidor que pertence unicamente ao usuário que a criou em que ele pode selecionar as entidades que deseja monitorar e por meio dela receber as informações relacionadas a essas entidades. O modelo fornecerá aos usuários não apenas informações que indiquem as condições atuais das entidades monitoradas, mas também recursos que disponibilizem a possibilidade de verificar as situações em que as entidades monitoradas foram expostas desde o início do processo de monitoramento.

As informações contextuais geradas deverão proporcionar, no mínimo, a localização da

entidade, visto que, existindo uma simples *tag* RFID, é possível determinar a sua localização por meio de técnicas de triangulação ou medição de intensidade de sinal. A qualidade do resultado obtido na geração do contexto dependerá da quantidade de sensores envolvidos no processo, quanto mais diferentes tipos de sensores utilizados e conseqüentemente tipos de dados obtidos, mais completo será o resultado. Portanto, a quantidade de recursos envolvidos no processo tenderá ao infinito conforme a disponibilidade.

Devido a necessidade de utilizar sensores junto as *tags* RFID, o Onisciente utilizará *tags* RFID ativas porque elas possuem fonte de energia própria que alimentarão os sensores envolvidos, mesmo quando as *tags* não estiverem ao alcance de uma leitora. Para estas situações onde não há comunicação, os dados coletados serão armazenados na *tag* para quando houver contato com uma leitora, eles possam ser transmitidos. Sensores que não estejam conectados às *tags* RFID também poderão ser utilizados, desde que eles apresentem dados relevantes para a entidade a qual o contexto será construído. Por exemplo, sensores que estejam monitorando as condições de um ambiente específico onde a entidade esteja presente.

A geração do contexto será realizada e representada por meio de ontologia. Seguindo a tendência identificada ao analisar os trabalhos relacionados, a ontologia em OWL aparenta ser a mais indicada. Possuindo o resultado do contexto criado, as informações obtidas estarão disponíveis para os usuários por meio de serviços fornecidos pelo servidor que realizou o processamento dos dados. Estes serviços poderão ser consumidos tanto por estações de trabalho, quanto por notebooks e outros dispositivos móveis, tais como *smartphones* e *tablets*. Um cliente web estará disponível para estações de trabalhos tradicionais e notebooks realizarem o monitoramento e poderem efetuar a visualização dos resultados. Já para os dispositivos móveis, um cliente realizará o processo de consulta de informações junto ao servidor para a exibição dos resultados. Ambos os clientes possuem a mesma base e apresentam as mesmas funcionalidades, o diferencial entre eles é a maneira como será implementada cada versão, o objetivo é aproveitar os recursos oferecidos pelos dispositivos móveis.

4.2 Requisitos

Para que o Onisciente atenda a todas as especificações descritas na visão geral, alguns requisitos funcionais foram definidos, são eles:

- Utilizar uma arquitetura cliente servidor para a utilização de serviços;
- Utilizar ontologias para a criação e representação de contextos;
- Permitir a visualização da situação de uma entidade em qualquer momento em que ela esteve sendo monitorada;
- Ser capaz de suportar tipos variados de entidades;

- Fornecer suporte a diversos tipos de dispositivos móveis como notebooks, tablets e smartphones, além dos tradicionais desktops;
- Utilizar uma quantidade indefinida de fonte de dados para a criação dos contextos;
- Permitir o uso de tipos variados de sensores para a aquisição de dados relacionados à entidade monitorada;
- Utilizar *tags* RFID ativas para fazer o monitoramento de entidades;
- Permitir o cadastro e edição de um perfil da entidade a ser monitorada;
- Detectar situações de risco para uma entidade;
- Notificar usuários sobre situações de risco de uma entidade.

4.3 Casos de Uso

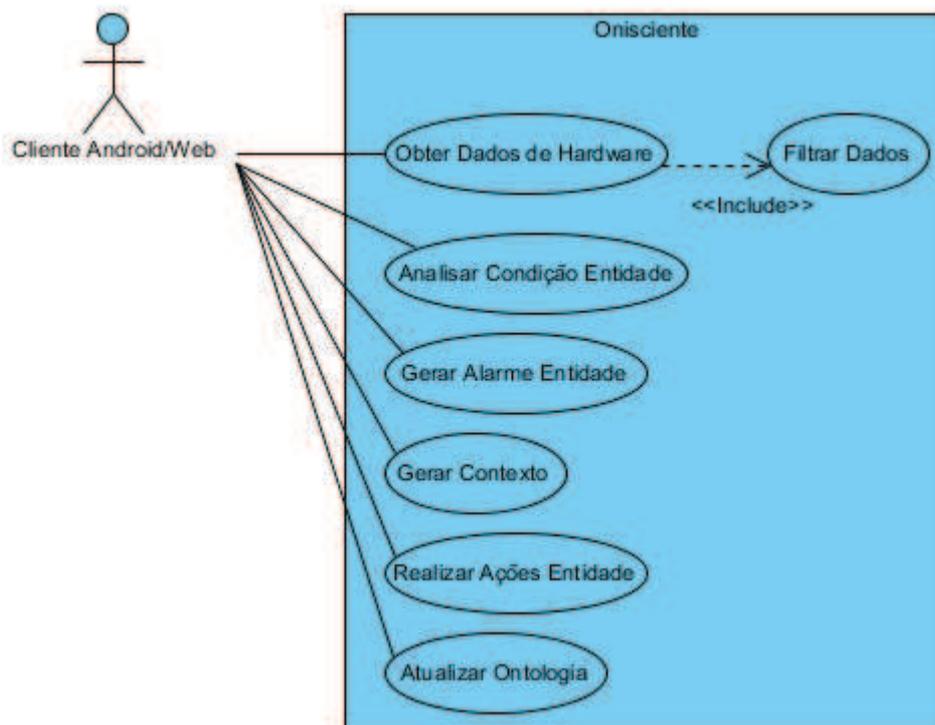
Com base nos requisitos e a especificação do modelo, é possível definir os atores envolvidos e seus casos de uso. O primeiro ator identificado é o Usuário, ele representa as pessoas que irão interagir com os clientes Android e Web do Onisciente. O segundo ator representa os clientes Android e Web onde são realizadas as operações relacionadas às entidades, ele representa cada dispositivo que interage com o servidor do Onisciente. Cada um desses atores possui uma quantidade de operações que são ou podem ser realizadas, conforme a necessidade encontrada.

4.3.1 Casos de uso do servidor

Os casos de uso do servidor podem ser visualizados na Figura 13, são eles:

- **Obter Dados:** Funcionalidade que realiza a interação com as leitoras RFID, obtendo os dados coletados. É executada conforme configuração prévia ou solicitação do processo de filtragem de dados;
- **Filtrar Dados:** Funcionalidade utilizada logo após uma coleta de dados que realiza uma verificação e validação no conteúdo adquirido. Elimina os dados redundantes e atua, em caso de necessidade, arbitrando quando houver ocorrência de conflitos de informação. Por exemplo, dois sensores de temperatura informando valores diferentes. Utiliza como critério de definição as últimas ocorrências registradas para o mesmo atributo em julgamento. Caso não exista um registro do atributo em questão, utiliza a média dos valores encontrados e requisita uma atualização do valor do atributo para a funcionalidade de obtenção de dados;
- **Gerar Contexto:** Aplica os dados já filtrados junto a ontologia para a criação dos contextos. Grava os resultados obtidos na base ontológica;

Figura 13: Casos de Uso do Servidor



Fonte: Elaborada pelo autor.

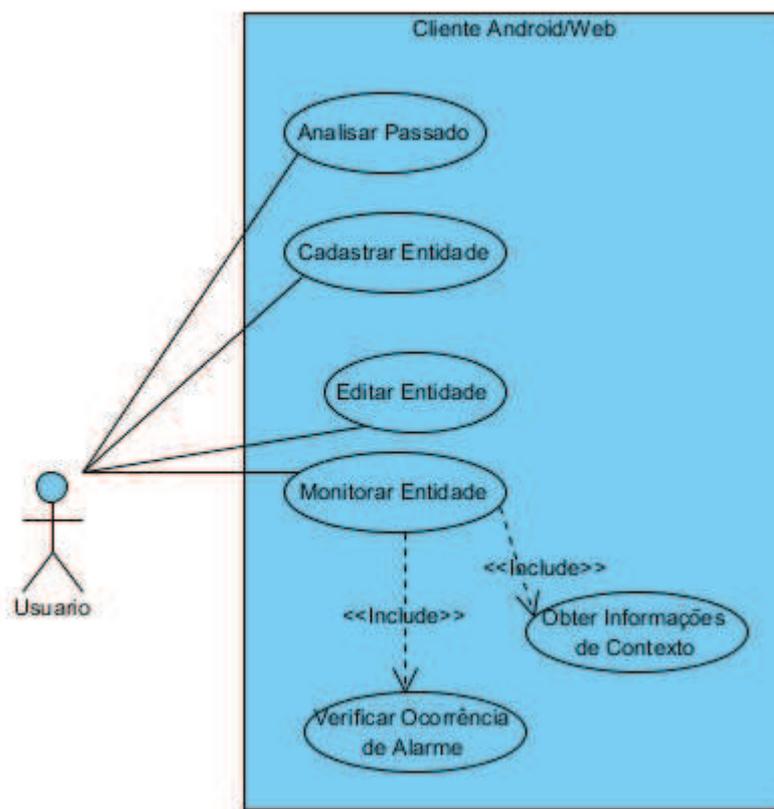
- **Analisar Condição de Entidade:** Cruza as informações de contexto geradas com os perfis da entidade em questão para determinar se existe alguma discrepância na entidade. Em caso de alguma ocorrência pode disparar a geração de um alarme;
- **Gerar Alarme de Entidade:** Esta funcionalidade é executada quando ocorre alguma violação ou ocorrência inesperada com a entidade relacionada, ela cria uma mensagem que será exibida nos clientes que estiverem acompanhando o monitoramento. Na mensagem consta a descrição da situação que ocorreu junto com um valor que indica o nível de gravidade da ocorrência;
- **Realizar Ações de Entidade:** Por meio desta funcionalidade são efetuadas as operações de cadastro e edição de entidades realizadas pelos usuários.

4.3.2 Casos de uso do cliente

A Figura 14 apresenta os casos de uso do cliente que são comentados a seguir:

- **Cadastrar Entidade:** Funcionalidades básicas do modelo para a entrada de dados relacionados às entidades a serem monitoradas. O cadastro consiste em identificar a entidade e especificar uma lista de atributos e intervalos de valores para cada um desses atributos

Figura 14: Casos de Uso do Cliente



Fonte: Elaborada pelo autor.

que serão utilizados para analisar as condições da entidade durante o processo de monitoramento;

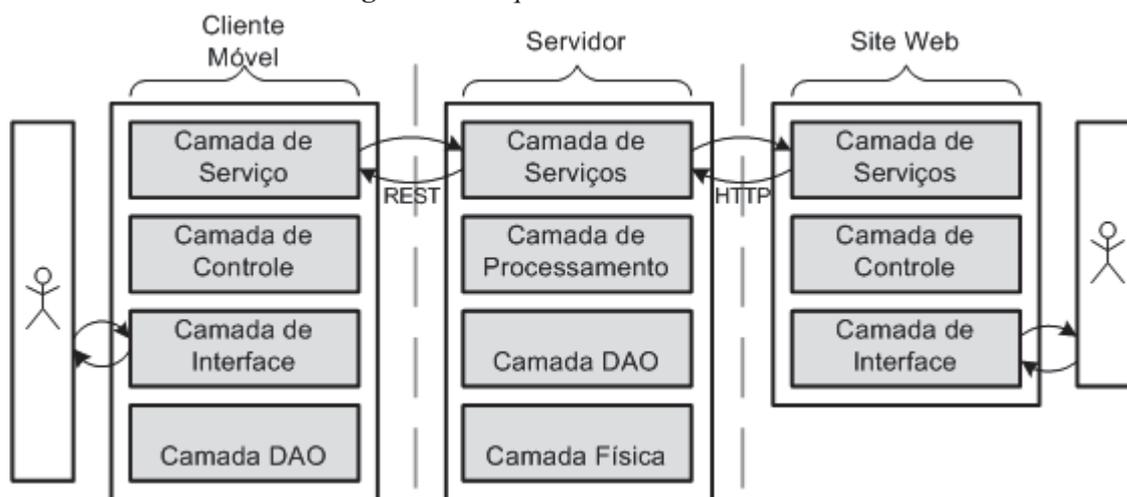
- **Editar Entidade:** Por meio dessa funcionalidade, o usuário terá a opção de editar uma entidade já cadastrada, permitindo alterar a quantidade, descrição e valores de seus atributos;
- **Monitorar Entidade:** Funcionalidade que gerencia o processo de monitoramento de uma ou mais entidades em tempo real. Cria uma sessão que fica constantemente requisitando as informações de contexto das entidades que foram selecionadas pelo usuário através de uma lista;
- **Analisar Passado de Entidade:** Esta funcionalidade permite verificar entidades que foram monitoradas em algum momento no passado;
- **Obter Informação de Contexto:** Esta funcionalidade é utilizada pelos clientes para adquirir as informações de contexto de uma entidade para posteriormente exibi-las para os usuários;
- **Verificar Ocorrência de Alarme:** Nesta funcionalidade é realizado o controle de ocorrências de alarmes de uma sessão de monitoramento. Se existir um registro, este será

exibido para o usuário.

4.4 Arquitetura

A arquitetura do Onisciente (Figura 15) foi projetada utilizando o modelo TAM da SAP (SAP, 2007), e é definida como cliente servidor fornecendo *web services* para diversos tipos de dispositivos móveis além do tradicional desktop. O servidor é responsável por adquirir e processar os dados das entidades monitoradas, convertendo-os em informação contextual. O cliente por sua vez consome a informação contextual gerada além de disponibilizar recursos para a interação do usuário com o servidor. Ele está apresentado em duas opções, um site web que pode ser acessado por computadores pessoais, notebooks e dispositivos móveis, e um aplicativo para dispositivos móveis.

Figura 15: Arquitetura Geral do modelo



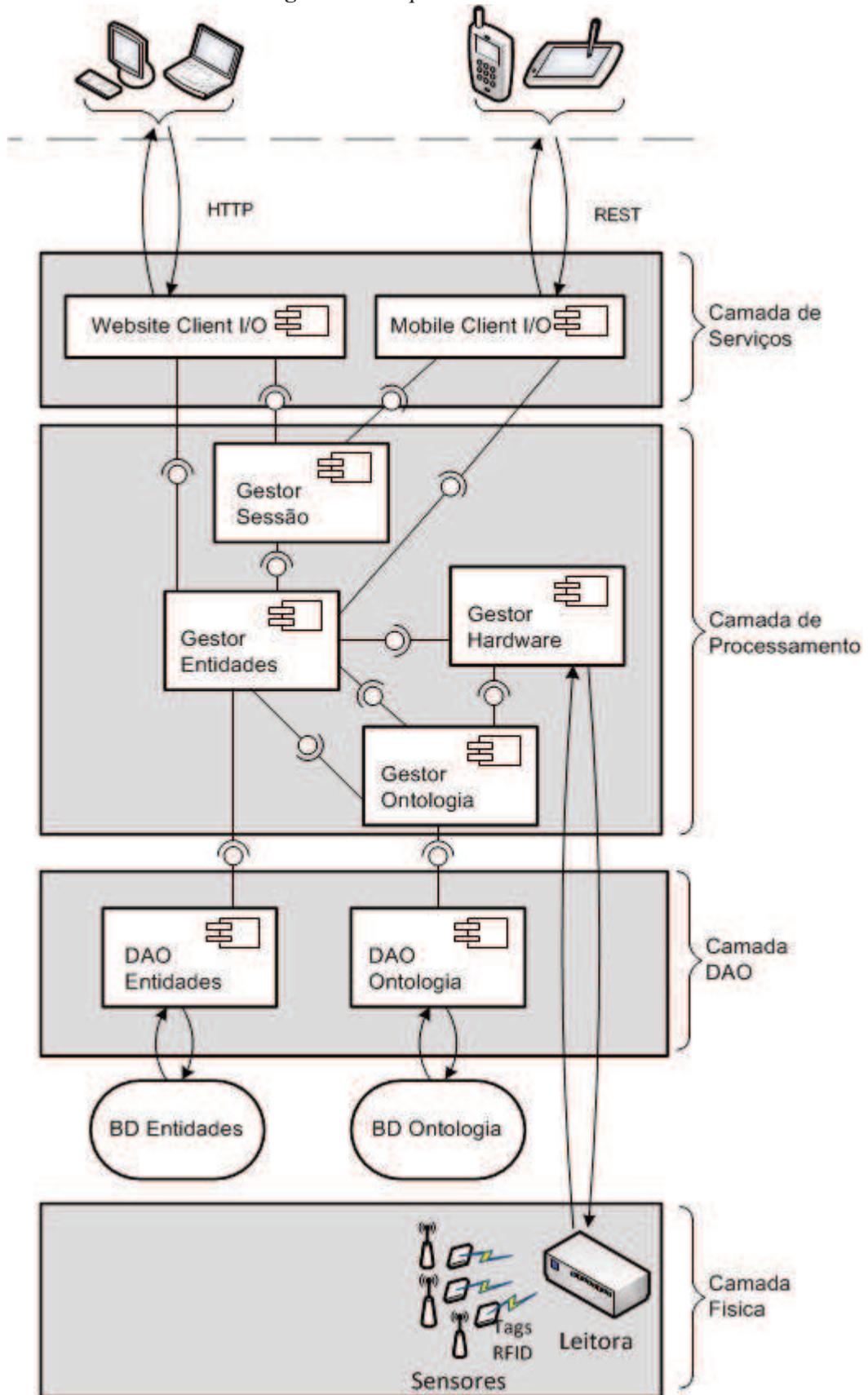
Fonte: Elaborada pelo autor.

4.4.1 Servidor

O servidor utiliza uma estrutura em quatro camadas (Figura 16), modelada segundo o modelo TAM da SAP (SAP, 2007). A camada mais baixa, denominada de *Camada Física*, é composta pela estrutura de hardware que realiza a aquisição de dados por meio do RFID. Nela estão inseridas as *tags* RFID ativas e as leitoras, e os sensores que estiverem disponíveis.

A segunda camada é composta pelos elementos que converterão dados brutos adquiridos pela primeira camada. Esta camada foi denominada de *Camada de Processamento*, ela é composta por três módulos que realizam funções distintas. O primeiro módulo é o *Gestor de Hardware*, sua função é receber os dados das leitoras RFID (Caso de uso Obter dados de hardware) e tratá-los de forma que não haja redundâncias nem conflitos de informação (Caso de uso Filtrar dados) para que possam ser aplicados no segundo módulo. O segundo módulo é o *Gestor de*

Figura 16: Arquitetura do Servidor



Fonte: Elaborada pelo autor.

Ontologia, ele é abastecido pelo *Gestor de Entidades* e recebe os dados processados do *Gestor de Dados* para aplicá-los na Ontologia resultando em uma informação contextual (Caso de uso Gerar Contexto) que será utilizada pelo terceiro módulo (Caso de uso Analisar Condição de Entidade). O terceiro módulo é o *Gestor de Entidades*, por meio dele o servidor interage com os clientes, recebendo e efetuando as operações de cadastro ou edição de entidades (Caso de uso Realizar Ações de Entidade), também abastece o *Gestor de Ontologia* para manter a base ontológica atualizada (Caso de uso Atualizar Ontologia). Outra funcionalidade deste módulo é a geração de alarmes quando necessário (Caso de uso Gerar Alarme de Entidade). Os alarmes possuem diversos níveis de gravidade e ocorrem quando os limites de um ou mais atributos descritos no perfil de uma entidade são ultrapassados por algum dado existente no contexto. Um alarme pode variar de uma simples notificação de alerta ou de situação crítica. Um quarto módulo, denominado *Gestor de Sessão*, gerencia as requisições de monitoramento de um usuário, ele mantém as listas de entidades monitoradas e interage com o *Gestor de Entidades* para consultar a situação destas entidades, notificando o usuário caso exista algum alarme.

A terceira camada é a *Camada DAO*, ela é composta por dois módulos e fornece recursos para que os módulos da Camada de Processamento tenham acesso aos dados armazenados. O primeiro módulo é o *DAO Entidades*, ele é responsável por realizar a persistência dos dados relacionados às entidades e sessões criadas pelos usuários na base no banco de dados BD Entidades. O segundo módulo é o *DAO Ontologia* e realiza persistência de dados no BD Ontologia. Este módulo é de uso exclusivo do Gestor de Ontologia.

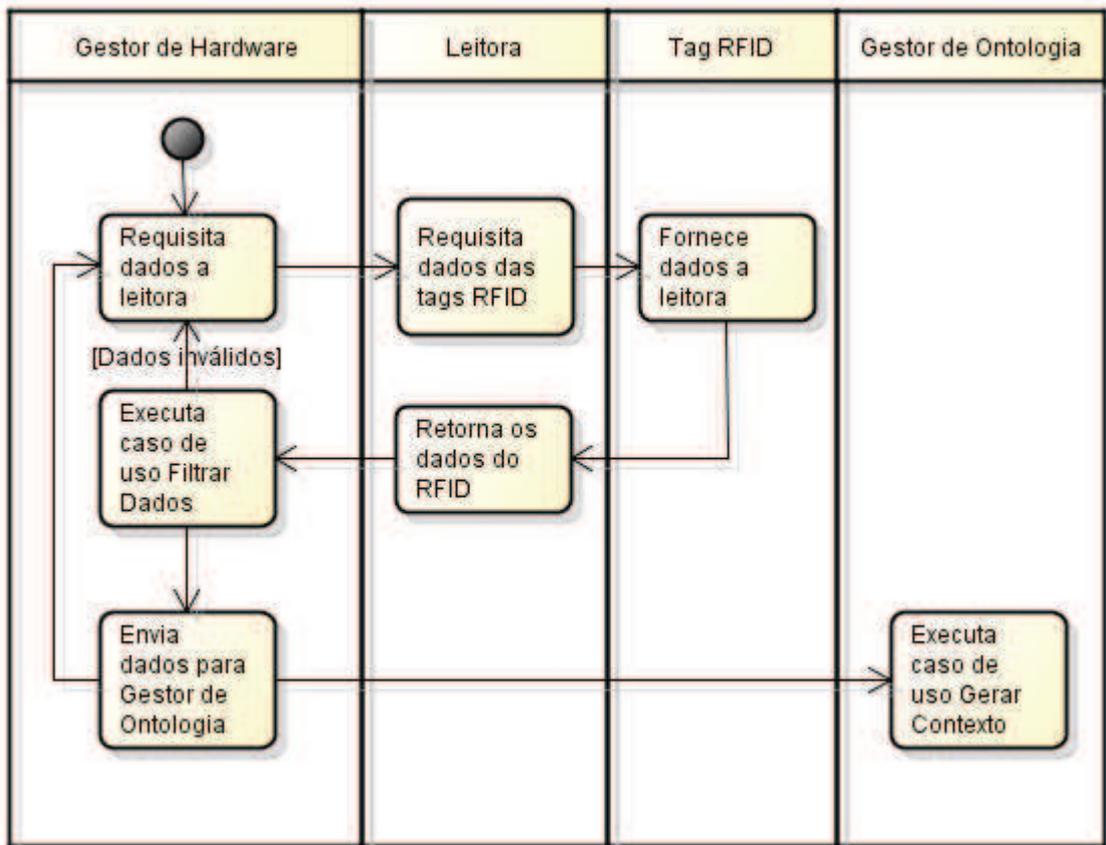
A última camada é a *Camada de Serviços*, ela fornece recursos para a comunicação entre os clientes e o servidor. Ela está dividida em dois grupos, o primeiro é constituído de apenas um módulo nomeado *Interface Física E/S*, ele contém um conjunto de métodos de permite com que a camada física possa interagir com o servidor realizando o envio de dados. O segundo grupo é composto por diversos módulos que realizam a interação entre o servidor e os clientes do Oniciente por meio do protocolo REST. Estes módulos estão divididos por tipo de controle que eles realizam. O *Modulo de Entidade* realiza operações relacionadas aos procedimentos de CRUD de entidades, o *Módulo de Sessão* ao controle de sessões e entidades que estão sendo monitoradas.

4.4.2 Diagramas de Atividade do Servidor

Para um melhor entendimento de como são realizados os procedimentos do Oniciente, um diagrama de atividade para cada caso de uso especificado é apresentado.

O primeiro diagrama apresentado (Figura 17) refere-se ao procedimento de obtenção dos dados que irão compor os contextos. Em um primeiro momento, o *Gestor de Hardware* requisita às leitoras que estiverem a sua disposição os dados das *tags* RFID. Estes dados após chegarem ao *Gestor de Hardware* são aplicados em um filtro e o resultado é enviado ao *Gestor de Ontologia* para processamento do contexto. Se o resultado do filtro não for válido então

Figura 17: Diagrama de Atividades - Obter dados de Hardware



Fonte: Elaborada pelo autor.

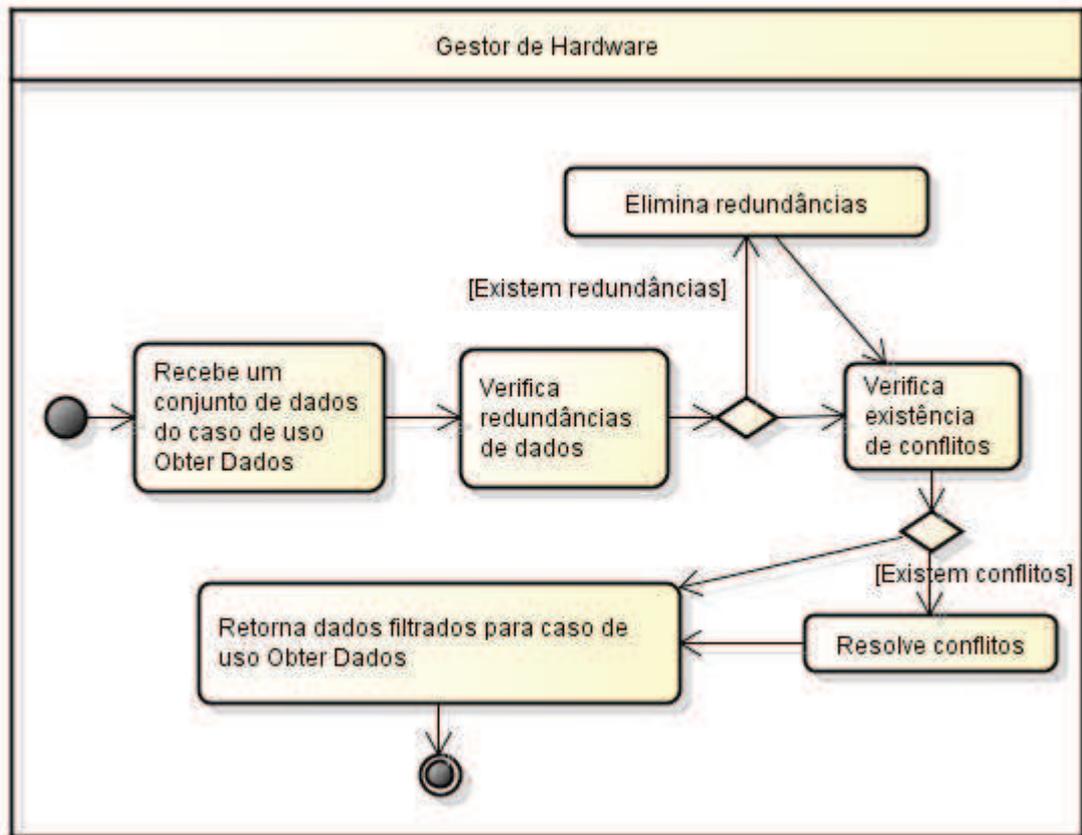
os dados são eliminados e uma nova consulta é realizada. O processo de requisição de dados ocorre indefinidamente com cada execução da rotina iniciando após um pequeno período de tempo depois dos dados serem enviados para o *Gestor de Ontologia*.

O próximo diagrama (Figura 18) descreve o processo realizado pelo caso de uso Filtrar Dados. Para cada conjunto de dados recebidos é realizada uma busca para a existência de redundâncias de dados, que são descartados se forem identificados. Em um segundo instante, é feita uma verificação para a ocorrência de conflitos, que devem ser resolvidos antes dos dados serem finalmente retornados para o caso de uso Obter Dados de Hardware. Caso os conflitos não possam ser resolvidos, o retorno é marcado como inválido para que os dados sejam excluídos posteriormente.

O diagrama da Figura 19 representa o processo de geração de contexto. Os dados recebidos através do *Gestor de Hardware* são aplicados na ontologia do modelo. Após o processo de criação, o resultado obtido, o contexto, é enviado para o *Gestor de Entidades* que executa o procedimento que avalia as condições da entidade.

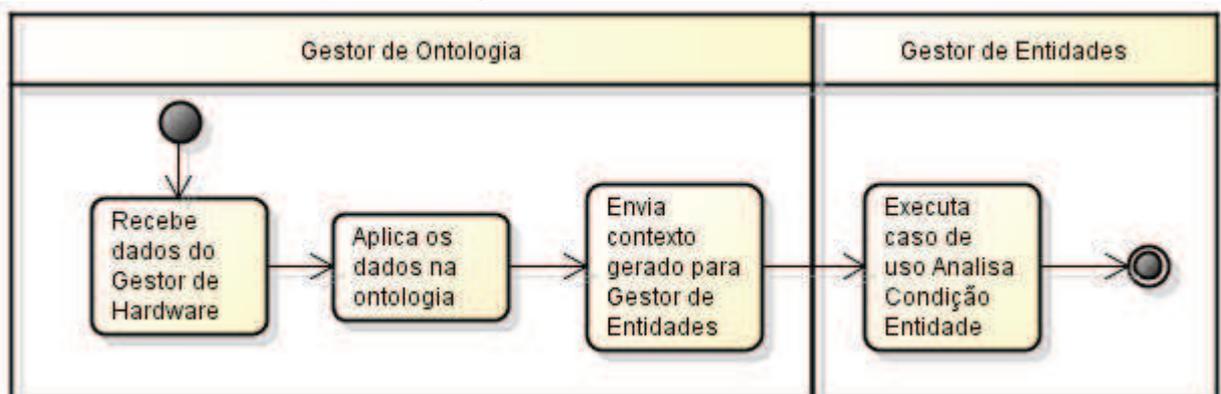
Quando um contexto é criado, ele logo em seguida é enviado para um procedimento denominado *Análise de Condição de Entidade*, a apresentação deste procedimento pode ser visualizada na Figura 20. Após o *Gestor de Entidade* receber o contexto, seu conteúdo é registrado junto à

Figura 18: Diagrama de Atividades - Filtrar Dados



Fonte: Elaborada pelo autor.

Figura 19: Diagrama de Atividades - Gerar Contexto

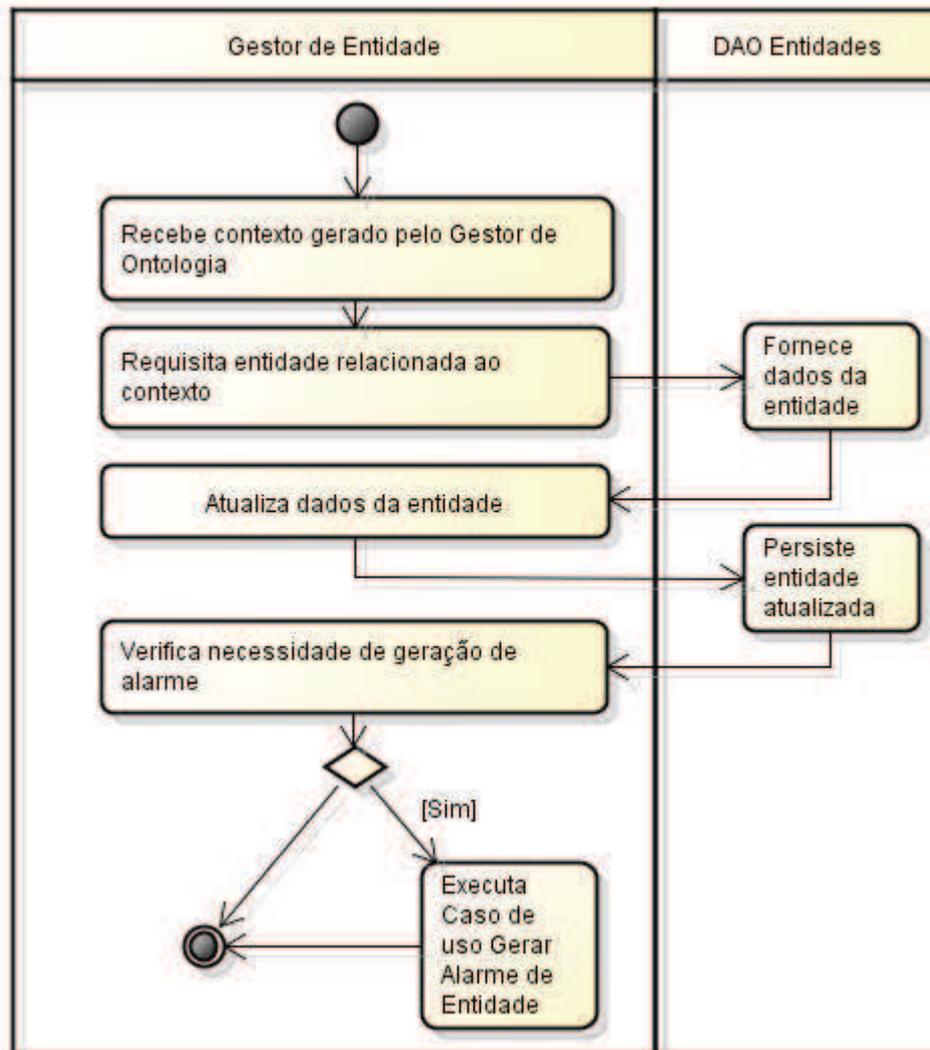


Fonte: Elaborada pelo autor.

entidade a qual ele pertence e persistido no banco de dados para que se tenha um histórico dos registros adquiridos. Por fim, o conteúdo do contexto é analisado e, se houver necessidade, gera um registro de alarme que será apresentado aos clientes na forma de uma mensagem.

Como mencionado anteriormente, registros de alarmes podem ser criados para serem exibidos aos usuários que estejam monitorando a entidade relacionada. No contexto criado, um dos campos existentes apresenta uma descrição da situação em que as entidades se encontram.

Figura 20: Diagrama de Atividades - Analisar Condição de Entidade

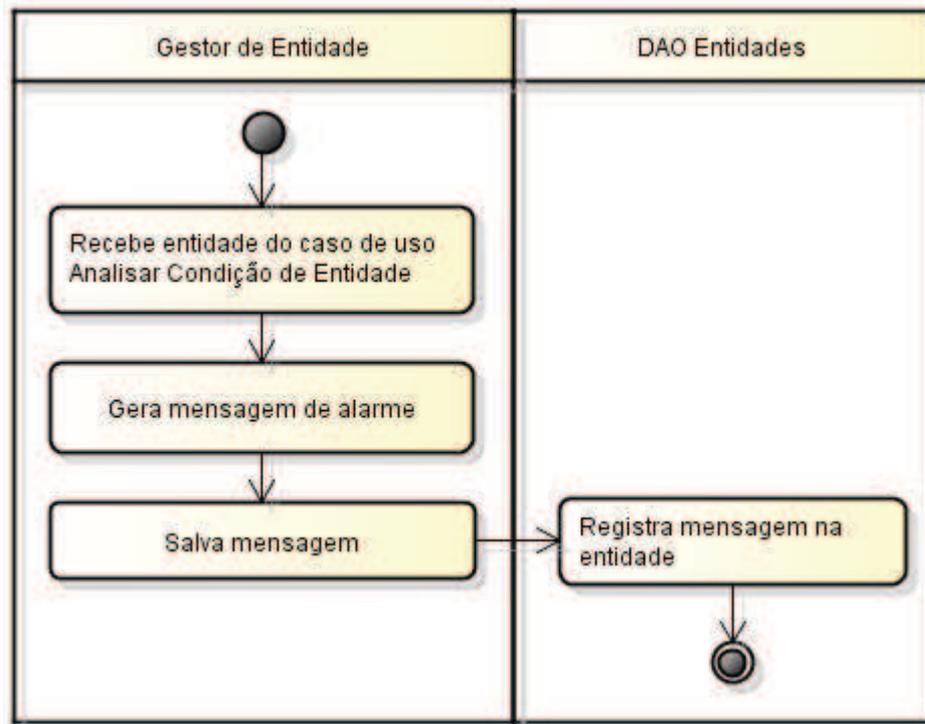


Fonte: Elaborada pelo autor.

Em alguns casos essa descrição informa que algum parâmetro especificado no momento do cadastro da entidade foi violado. Quando isso acontece o usuário deve ser informado e o caso de uso Gerar Alarme, representado pelo Diagrama de Atividades da Figura 21, trata essa situação. Pela descrição da figura, quando ocorrer a situação descrita, uma mensagem é criada e anexada à entidade, para futuramente, quando requisitada por um cliente, ela possa ser exibida para o usuário.

Um caso de uso denominado Realizar Operação de Entidade foi definido para que atue como um centralizador de operações relacionadas às entidades. Isso significa que processos de criação e edição de entidades são realizados por esse procedimento. Conforme a Figura 22 a *Camada de Serviços* envia para a *Gestor de Entidades* os dados e o tipo de operação que deverá ser efetuada, o gestor faz uma análise desses dados para descobrir se eles estão consistentes e logo em seguida faz o procedimento de persistência dos dados no banco de dados. Por fim envia um retorno para a *Camada de Serviços* que por sua vez irá apresentar o retorno de operação para

Figura 21: Diagrama de Atividades - Gerar Alarme



Fonte: Elaborada pelo autor.

o cliente e também fará uma atualização na ontologia, adicionando ou atualizando os dados de uma entidade.

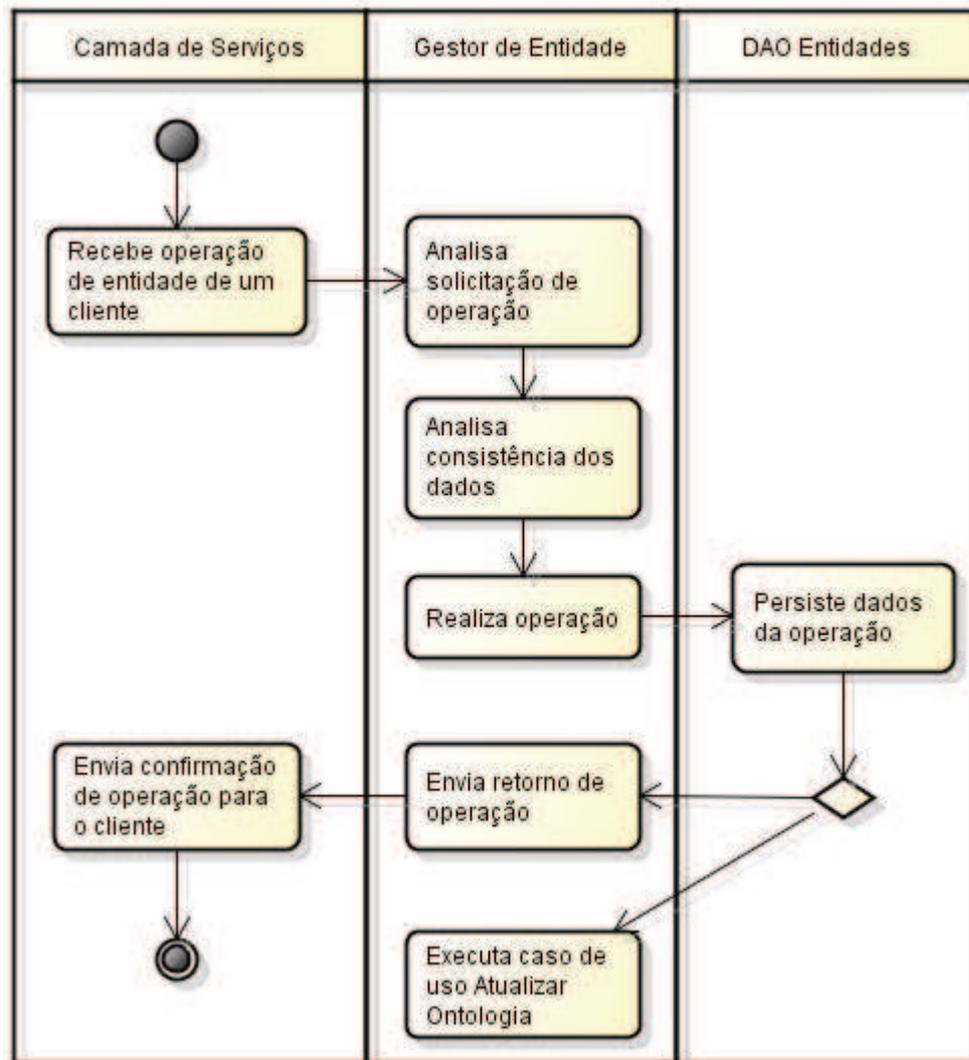
O procedimento de atualização de uma ontologia é apresentado na Figura 23. Para cada entidade cadastrada ou atualizada pelo *Gestor de Entidade*s, uma operação semelhante é realizada na base ontológica. Quando houver um cadastro de entidade, os mesmos dados deverão ser adicionados na base ontológica, já quando ocorrer uma atualização, os dados deverão ser analisados e posteriormente atualizados na base de dados. Esse espelhamento de informação é para quando for realizada a criação de um contexto, o resultado não seja baseado em dados inválidos o que poderia gerar complicações na avaliação das entidades envolvidas.

4.4.3 Cliente Site Web

Objetivando permitir acesso as informações produzidas, é oferecida uma opção de site web para que seja possível monitorar entidades em qualquer tipo de dispositivo que possua conexão com a internet e um meio de fazer a exibição das informações recebidas.

A arquitetura do site web foi modelada conforme o modelo TAM da SAP (SAP, 2007) e está estruturada em três camadas (Figura 24). A primeira camada é a *Camada de Serviços*, onde estão localizadas as funções que fazem a comunicação com o servidor através de requisições para a realização do monitoramento de entidades ou envio de comandos para a concretização das operações relacionadas às entidades.

Figura 22: Diagrama de Atividades - Realizar Operação de Entidade

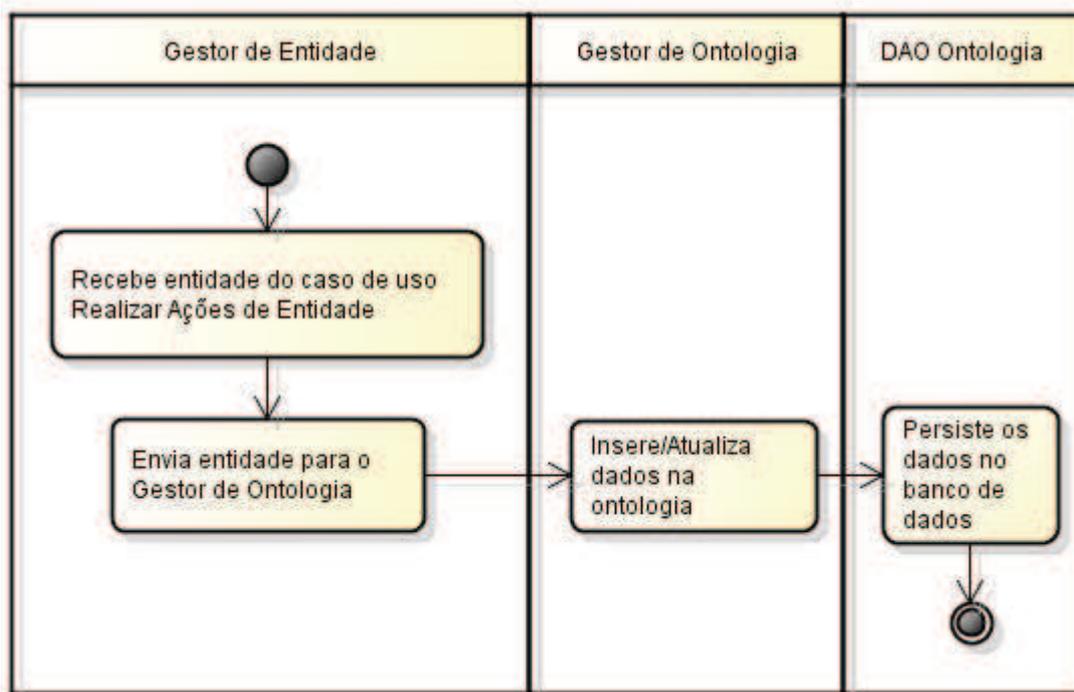


Fonte: Elaborada pelo autor.

A segunda camada é a *Camada de Controle*, ela apresenta os elementos de interação com o usuário, necessários para o funcionamento correto do modelo. O conteúdo de controle está subdividido em três módulos que abrangem cada, um conjunto de operações do modelo. O primeiro módulo é o *Gestor de Acesso*, ele interage com o servidor realizando controle de acesso de usuários. O segundo módulo é o de *Gestor de Entidade*. Este módulo é composto pelas operações de validação, cadastro e edição de entidades. O terceiro módulo é o *Gestor de Monitoramento*, ele faz o controle do processo de monitoramento de entidades, desde a seleção de entidades até a geração de relatórios de monitoramento geral e individual de cada entidade.

A terceira camada é composta pelas páginas web por onde o usuário pode interagir. Está subdividida em dois grupos: O primeiro é o de *Páginas de monitoramento*, que abrange as páginas que permitem a seleção de entidades, o monitoramento em si, e apresentação de relatórios; o segundo grupo é o de *Páginas de CRUD de entidades*, ele contém as páginas que afetam uma entidade, entre suas opções estão as páginas de cadastro e edição de entidades.

Figura 23: Diagrama de Atividades - Atualizar Ontologia



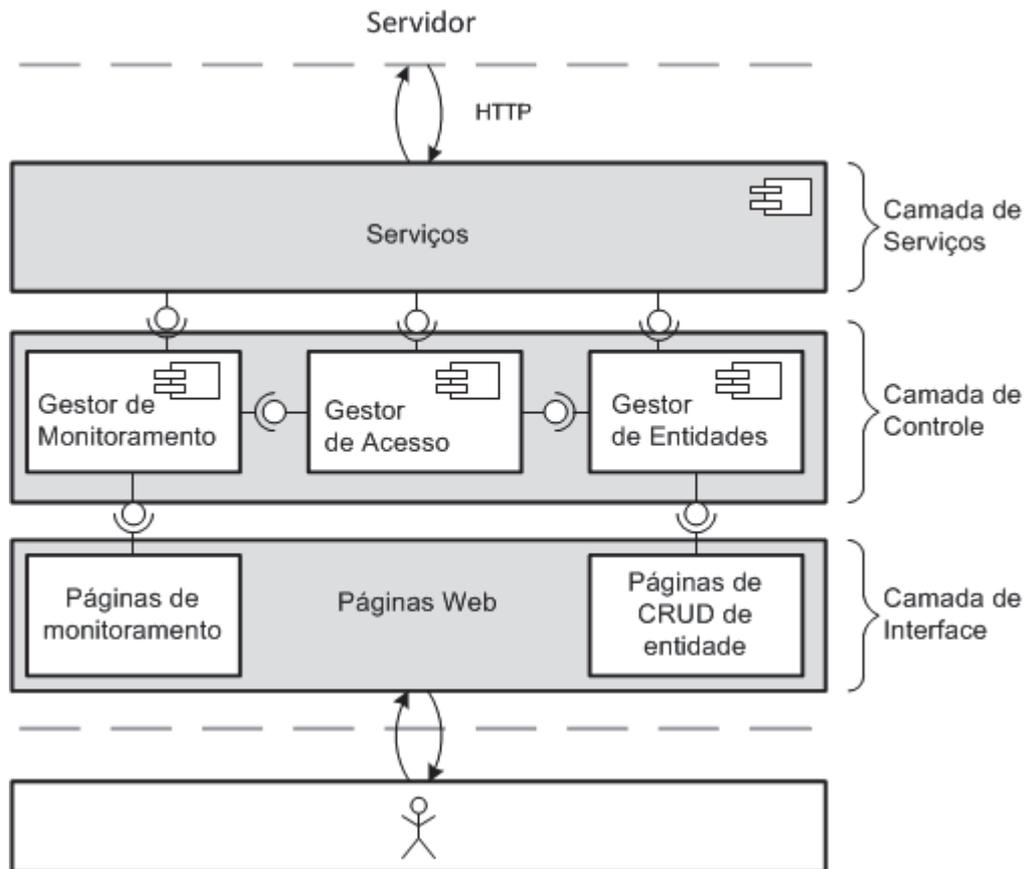
Fonte: Elaborada pelo autor.

4.4.4 Cliente Móvel

A decisão de fornecer um aplicativo para dispositivos móveis é devido à possibilidade de explorar recursos oferecidos pelos sistemas operacionais desses equipamentos. A grande variedade de dispositivos que executam os sistemas operacionais Android e iOS torna viável a sua utilização em grande escala. O uso do cliente móvel traz algumas vantagens significativas com relação ao uso do site web. Primeiro, é possível realizar armazenamento local, o que reduz a quantidade de dados que necessitam ser transmitidos entre o servidor e o cliente. Segundo, utilizando o processador dos dispositivos móveis permite que o processamento no lado do servidor seja diminuído. Terceiro, o armazenamento local, citado anteriormente, torna possível fazer uma análise das situações em que uma entidade foi submetida em situações passadas, sem que haja a necessidade de estar conectado ao servidor.

O cliente móvel foi planejado utilizando uma arquitetura em quatro camadas (Figura 25), segundo o modelo TAM da SAP (SAP, 2007). A camada superior é a *Camada de Serviços*, ela realiza as interações com o servidor, requisitando ou enviando dados conforme a necessidade encontrada pela *Camada de Controle*, a segunda camada. A *Camada de Controle* possui três módulos: O módulo *Gestor de Acesso*, que valida o usuário da aplicação e suas operações antes de tratar com o servidor; o módulo *Gestor de Entidades*, que realiza as operações de CRUD de entidades junto ao servidor; e o módulo *Gestor de Monitoramento*, que controla a sessão de monitoramento junto ao servidor, solicitando as informações das entidades envolvidas e apresentando os resultados a cada atualização recebida.

Figura 24: Arquitetura do Cliente Web



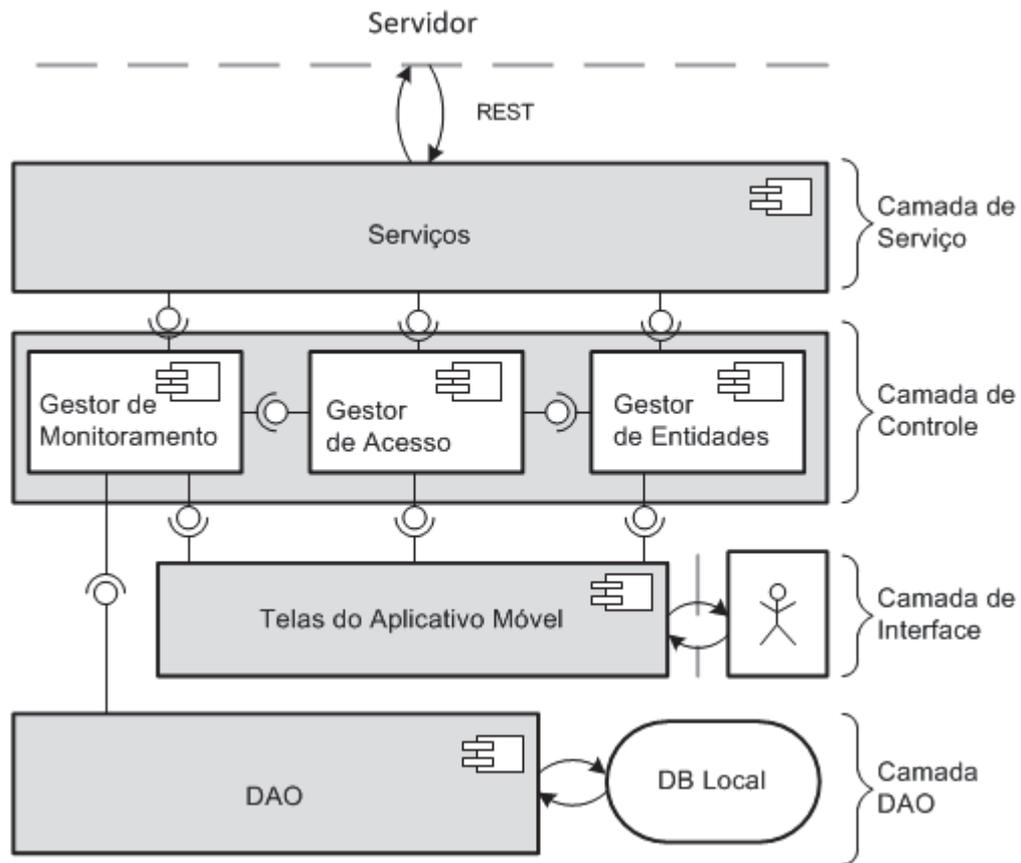
Fonte: Elaborada pelo autor.

A terceira camada é a DAO, ela é utilizada pelo *Gestor de Monitoramento* para armazenar as informações das sessões que foram executadas, para que seja possível analisar a situação de entidades no passado quando não for possível obter as informações de um servidor. A última camada é a *Camada de Interface*, ela contém as telas que permitem o usuário realizar as funcionalidades disponíveis.

4.4.5 Diagramas de atividade do cliente

Tanto o cliente web quanto o cliente para dispositivos móveis possuem a mesma base em sua concepção, por isso, utilizam as mesmas funcionalidades, a exceção fica com relação ao caso de uso Analisar Passado de Entidade que se estiver sendo executada por um cliente móvel, utiliza sua base local antes de consultar um servidor.

A Figura 26 e a Figura 27 apresentam diagramas de atividade Cadastrar Entidade e Editar entidade respectivamente. O processo de cadastramento de entidade é iniciado com a apresentação do formulário de edição para o usuário que após mandar salvar a nova entidade, tem os dados dos atributos avaliados pelo *Gestor de Entidade*, que pode ou não rejeitá-los. No caso de rejeição, o procedimento retorna para o formulário onde o usuário deverá corrigir os dados in-

Figura 25: Arquitetura do Cliente Móvel

Fonte: Elaborada pelo autor.

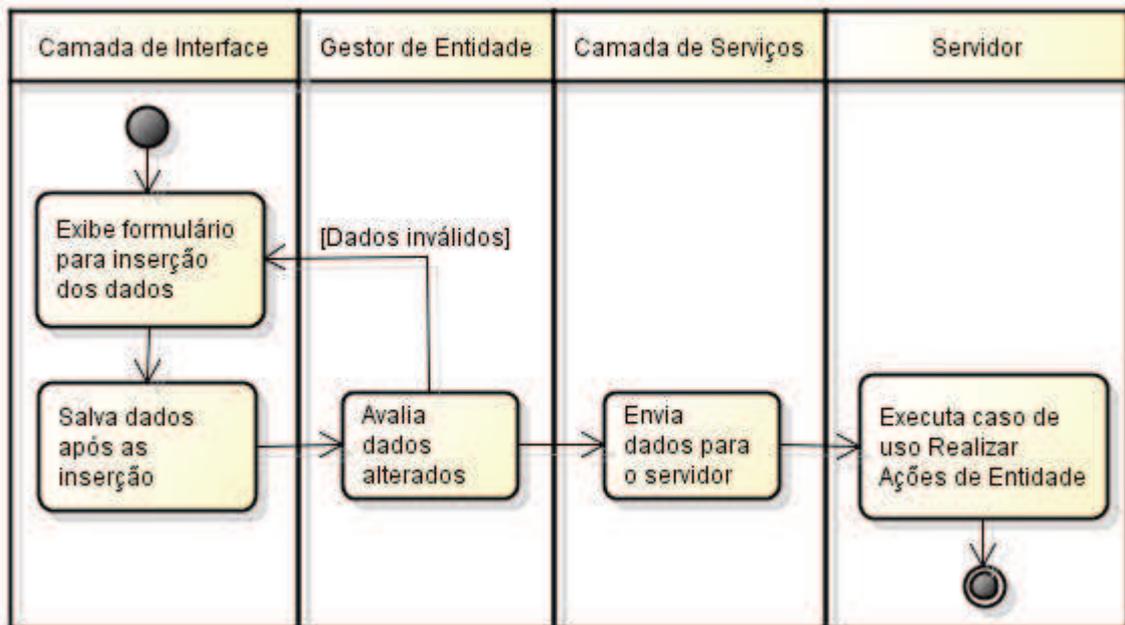
válidos. Se não houver problemas durante o procedimento de avaliação, os dados são enviados para o servidor por meio da *Camada de Serviços* para serem persistidos no banco de dados.

O processo de edição de uma entidade é similar ao cadastro, a diferença é que antes de apresentar o formulário de edição ao usuário, há uma requisição ao servidor para adquirir os dados da entidade que se deseja alterar. O restante do procedimento após receber os dados é igual ao do cadastro de novas entidades.

O processo de monitoramento de uma entidade é apresentado pelo caso de uso Monitorar Entidade, ilustrado pelo Diagrama de Atividade da Figura 28. Todo processo de monitoramento inicia com o usuário selecionando uma opção de monitoramento no cliente, esta opção requisita junto ao servidor uma lista de entidades que estejam disponíveis para monitoramento, que são apresentadas ao usuário. O usuário seleciona as entidades que deseja e então é iniciado um processo cíclico onde as informações contextuais dessas entidades são requisitadas ao servidor e apresentadas ao usuário. Este processo ocorre infinitamente enquanto houver coleta de informações das entidades ou até o usuário encerrar a sessão. Durante o procedimento cíclico, caso ocorra uma ocorrência de alarme, uma mensagem será informada ao usuário para que este tenha conhecimento da situação.

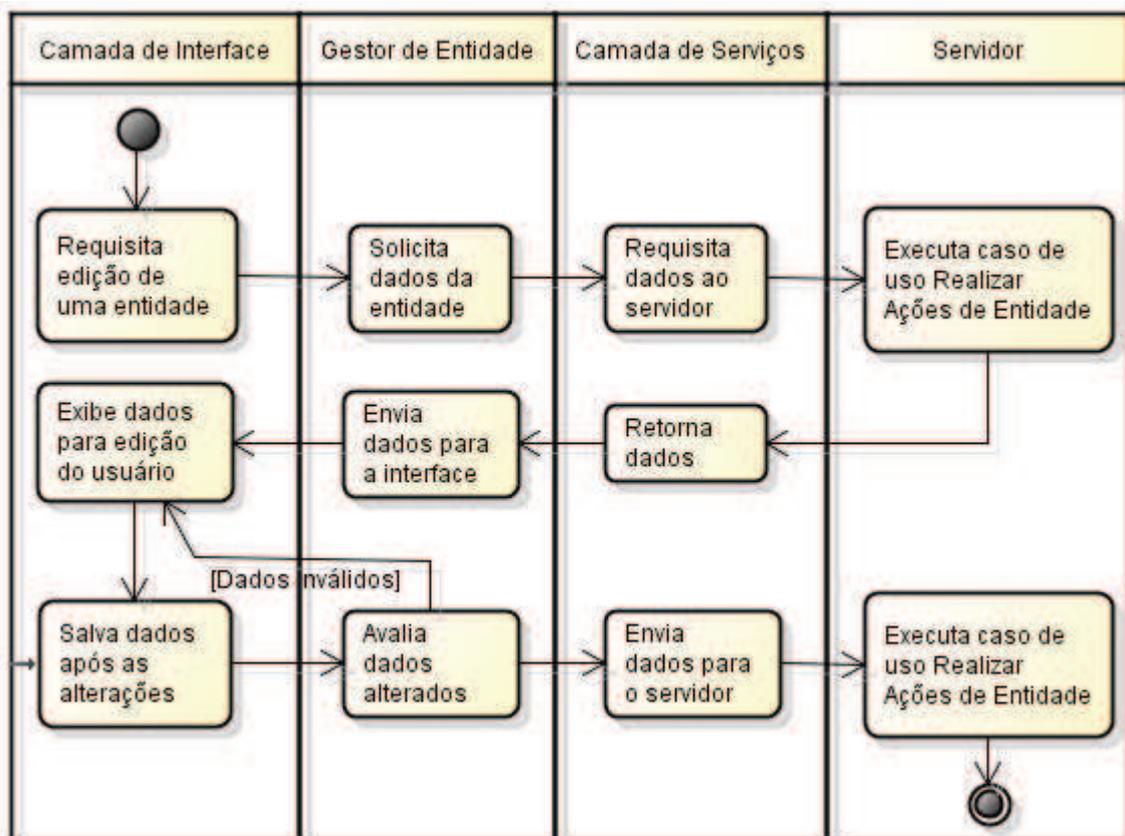
O monitoramento de entidades é dependente de outros dois casos de uso, o Obter Informa-

Figura 26: Diagrama de Atividades - Cadastrar Entidade



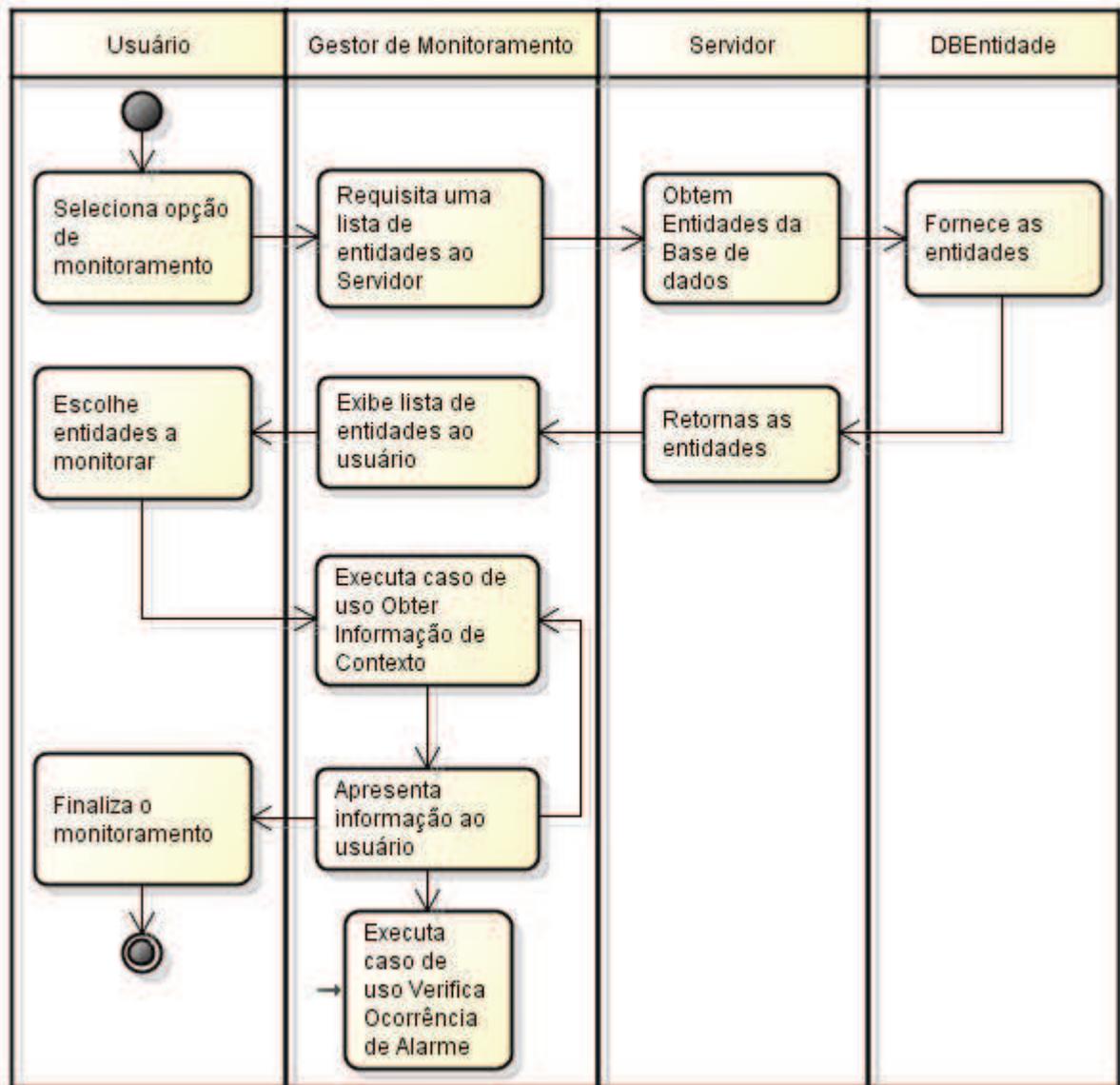
Fonte: Elaborada pelo autor.

Figura 27: Diagrama de Atividades - Editar Entidade



Fonte: Elaborada pelo autor.

Figura 28: Diagrama de Atividades - Monitorar Entidade



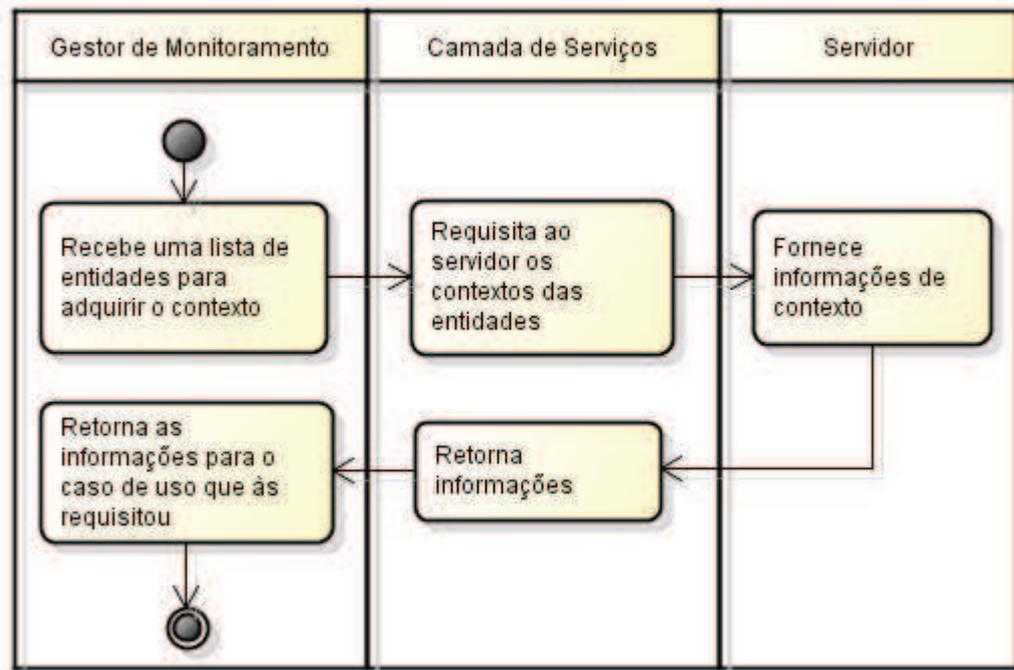
Fonte: Elaborada pelo autor.

ção de Contexto, representado pelo Diagrama de Atividade da Figura 29, e Verificar Ocorrência de Alarme, demonstrado pelo Diagrama de Atividades da Figura 30. Ambos os casos são executados pelo processo cíclico descrito anteriormente. A obtenção de informação de contexto nada mais é do que um procedimento de verificação de novas informações relacionadas às entidades a cada iteração do ciclo junto ao servidor. O resultado dessa consulta é enviado diretamente ao procedimento que invocou a consulta.

A verificação de alarmes é similar ao procedimento de consulta de situação de contextos, mas com o diferencial de que este realiza uma verificação no conteúdo recebido pelo servidor e, havendo necessidade, apresenta uma mensagem de alerta para o usuário.

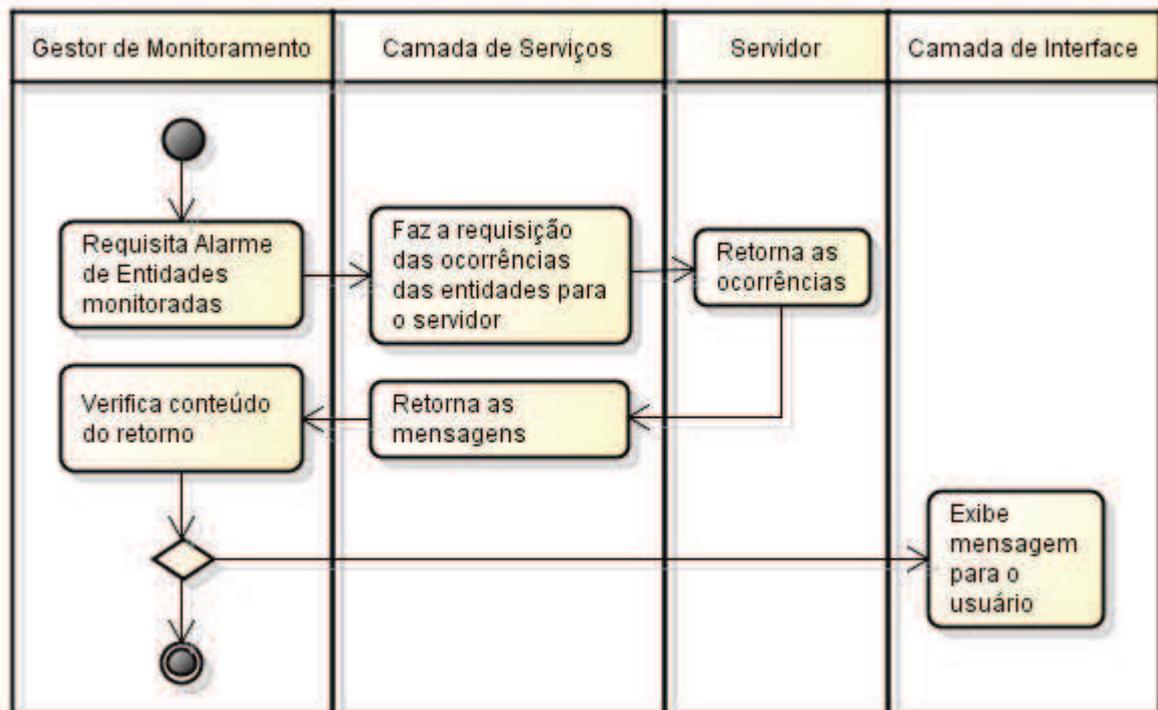
O procedimento que permite analisar as condições passadas de uma entidade apresenta uma diferença entre a versão do cliente web (Figura 31) e a versão do cliente móvel (Figura 32).

Figura 29: Diagrama de Atividades - Obter Informação de Contexto



Fonte: Elaborada pelo autor.

Figura 30: Diagrama de Atividades - Verificar Alarme



Fonte: Elaborada pelo autor.

Ambos os procedimentos são iniciados com o usuário selecionando algumas entidades para o monitoramento e um período de tempo relacionado. No cliente web, estas entidades são

obtidas diretamente com o servidor, já no cliente móvel, uma consulta no banco de dados local do dispositivo é realizada antes, requisitando as informações ao servidor apenas se não houver registros das entidades na base local. Isto é feito dessa maneira para evitar o consumo de bateria elevado do dispositivo com operações que envolvem tráfego de dados.

Figura 31: Diagrama de Atividades - Analisar Passado de Entidade - Website

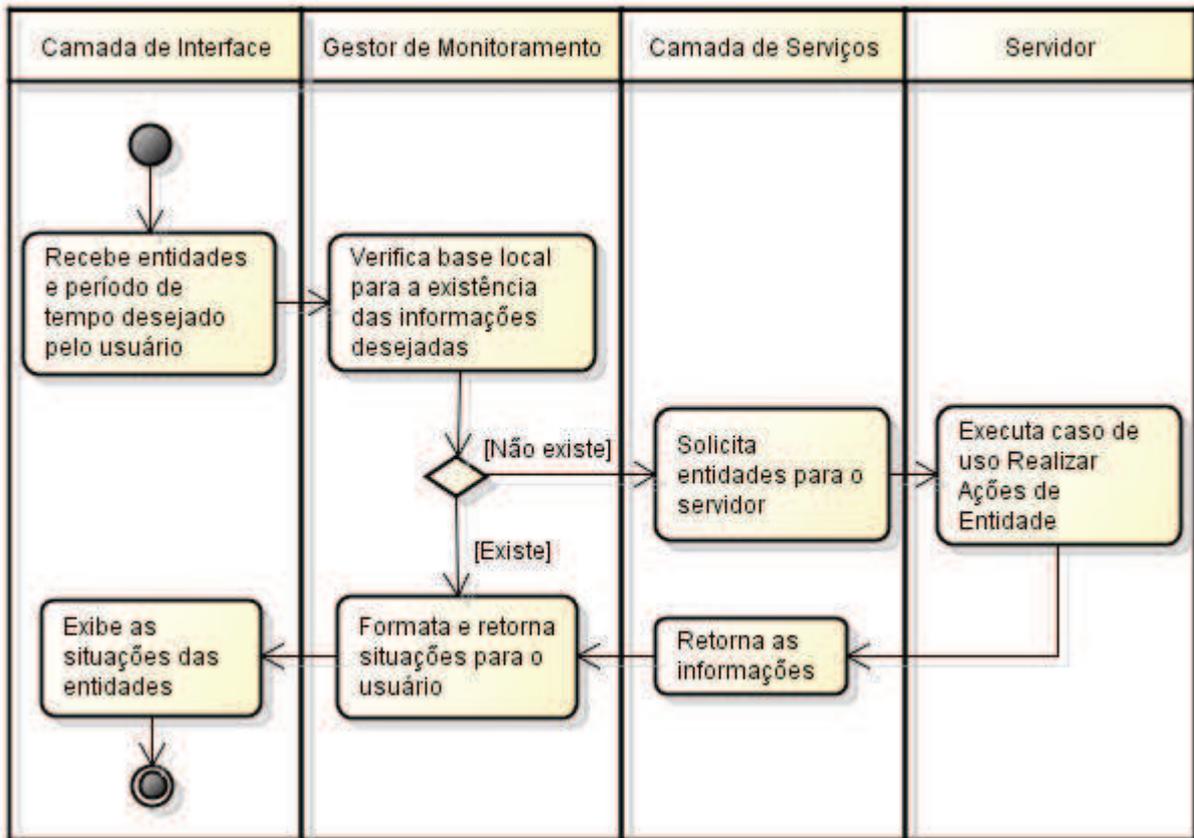


Fonte: Elaborada pelo autor.

4.5 Modelo de representação da entidade

O modelo descrito não envolve uma área específica de aplicação e para permitir uma ampla abrangência de possibilidades, um modelo genérico de representação de entidade deve ser proposto. Visando permitir flexibilidade na definição de entidades, optou-se por modelar a entidade por meio de uma classe denominada Entidade composta por: um atributo *ID_entidade* para a inserção de um código único que deverá ser preenchido pelo servidor no momento do cadastro da entidade; um atributo *Nome_entidade* que conterà a identificação da entidade para o usuário, uma estrutura de dados do tipo map com o nome de *Lista_de_Atributos* que fará o armazenamento dos atributos da entidade; uma lista de mensagens que servirá para armazenar registros de alarmes e uma lista de contextos que servirá como um histórico do monitoramento realizado. A estrutura *map* é composta por uma tupla de dois elementos, uma chave e um valor, que não permite duplicatas, sendo assim, ideal para o controle dos atributos das entidades.

Figura 32: Diagrama de Atividades - Analisar Passado de Entidade - Dispositivo Móvel



Fonte: Elaborada pelo autor.

4.6 Modelo de Contexto

No Onisciente, um contexto contém a descrição da situação de uma entidade, que permite a um usuário ter o entendimento das suas condições conforme ocorre o monitoramento. Nele deve constar além da descrição das condições da entidade que ele representa, um registro da data e hora em que foi gerado, e uma indicação da sua localização. Esta localização pode ser variável, ela pode indicar uma posição global baseada em latitude e longitude, uma posição relativa a uma leitora RFID, ou uma informação alto nível, por exemplo, um cômodo específico dentro de uma construção.

Para poder gerar informações de contexto, é desejado um método que possa trabalhar com diversos tipos de dados e tipos de entidades. Este método deve permitir que as entidades possam ser inseridas e também avaliadas a cada nova informação vinda de sensores. O processo de avaliação das entidades deve levar em conta as suas características e o resultado deve constituir contextos simples ou complexos, conforme a quantidade de dados envolvidos no processo.

Seguindo estes requisitos, se optou por utilizar ontologias como um motor de inferência para definir as situações de contextos das entidades do sistema. Uma ontologia permite extrair valor semântico de dados que estejam dentro do seu modelo de domínio a partir das relações entre os

seus elementos. Ontologias podem realizar o processamento bruto de análise para criação de contextos de uma forma mais simples do que seria fazer o processo por meio de programação tradicional graças ao seu poder de inferência.

A metodologia escolhida para a elaboração da ontologia foi proposta por Noy e McGuinness (2001). O domínio da ontologia é representado por *Contexto* e este possui inúmeras subclasses que representam os diversos tipos de contextos que são possibilitados para cada entidade envolvida. Os principais termos da ontologia são: contexto, pessoas, objetos, umidade, temperatura e lugar.

A Figura 33 apresenta a estrutura de classes elaborada para atender ao objetivo proposto para a ontologia. Entidades estão subdivididas em três tipos, cada uma com características próprias, que a diferem uma das outras. *Lugares* possuem atributos que informam valores de umidade e temperatura de um local, *Pessoas* possuem informações de temperatura de um indivíduo, mas não uma informação de umidade, e *Objetos* possuem limitações de temperatura e umidade, máximas e mínimas, que são utilizadas para determinar a situação em que se encontram. Algumas dessas classes de entidades possuem algumas subdivisões, *Lugares* podem ser do tipo *Indoor* e *Outdoor*, onde a diferença está em o lugar ser ou não ao ar livre, *Objetos* podem ser *Perecíveis*, onde possuem um tempo de vida útil ou um bem durável *Fixo*, como uma casa que não pode ser deslocada, ou ainda um bem durável *Móvel*, como um celular que pode ser transportado para qualquer lugar. Sensores estão subdivididos por tipos e podem exercer influência sobre qualquer tipo de entidade.

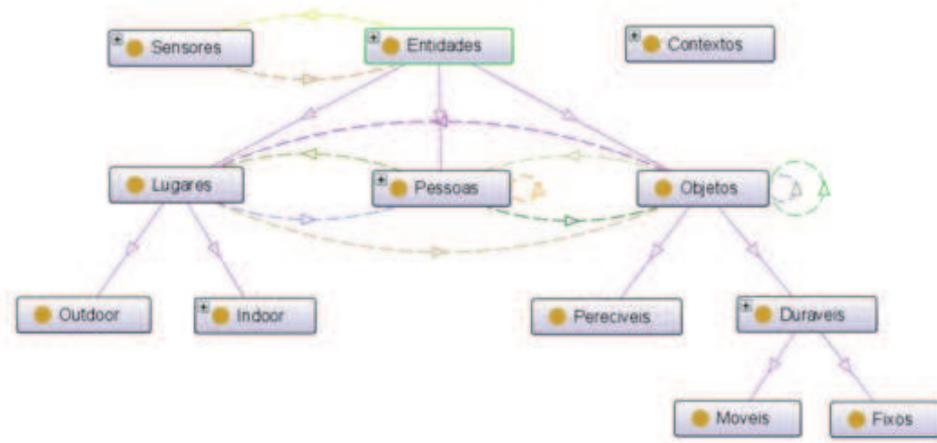
Ainda na Figura 33, é possível perceber a existência ligações entre as diversas classes definidas na ontologia, elas representam as propriedades das classes e definem as relações entre as classes. Ao ser inserida uma pessoa na ontologia, a informação deve informar o lugar que ela está inserida e, se for o caso, também informar a relação de objetos que esta pessoa possui. *Objetos* podem estar inseridos ou fazer parte de outros objetos maiores, assim como ser o resultado final da união de diversos objetos menores, além disso, devem estar incluídos em algum lugar e podem pertencer a uma pessoa. Os *Lugares* são uma referência para a definição da localização de objetos e pessoas dentro da ontologia.

A classe *Contextos* aparenta estar desconexa das demais classes na Figura 33, isto é porque ela não está expandida e não podemos ver a sua estrutura hierárquica. Na Figura 41 podemos visualizar as suas subclasses e também a relação com algumas entidades específicas. Cada entidade possui um grupo de contextos relacionados, estes contextos indicam quais são as entidades que estão de alguma maneira, fora das condições ideais. *Objetos* são avaliados segundo os seus atributos, *Pessoas* possuem regras predefinidas que determinam algum fator de risco. Por exemplo, é conhecido que a temperatura de uma pessoa deve variar entre 36°C e 37°C para ser considerada normal e qualquer valor de temperatura fora desse intervalo é considerado febre. Também é conhecido que alguns intervalos de umidade são maléficos para a saúde de uma pessoa. *Lugares* não possuem contextos definidos porque são os meios como os contextos são avaliados. Um *Objeto* só pode ser avaliado sabendo onde ele está inserido, o mesmo vale para

uma pessoa.

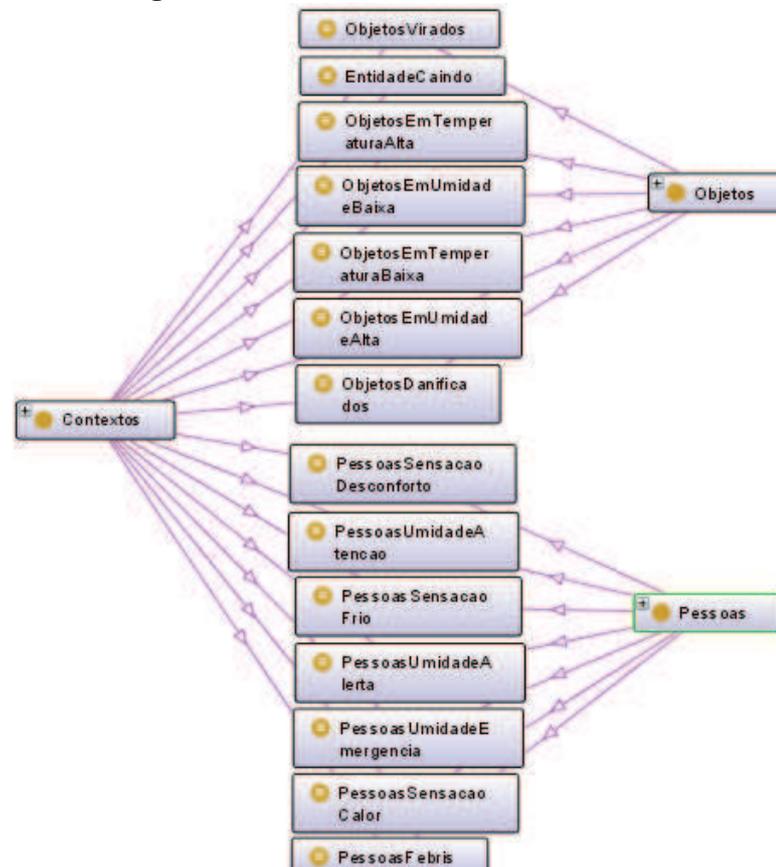
Um detalhe que justifica a aparente desconexão da classe *Contexto* com as demais classes da ontologia, conforme visto na Figura 33, é o fato de que as subclasses de *Contexto* são classes derivadas das classes de entidades.

Figura 33: Estrutura Geral de Classes da Ontologia



Fonte: Elaborada pelo autor.

Figura 34: Estrutura das Classes de Contexto



Fonte: Elaborada pelo autor.

A ontologia foi projetada para trabalhar com diversos tipos de entidades e é perfeitamente possível estender o seu uso agregando novos tipos de entidades e novas classificações de contextos, assim como atribuir novas propriedades que possam agregar valor à ontologia. Em função desta característica, uma estrutura foi elaborada para a representação do contexto dentro do modelo. Trata-se de uma tupla de 5 elementos (Figura 35), similar a apresentada no trabalho relacionado de Sheng et. al (2011), constituída pelo ID da entidade, data, hora, localização da entidade e descrição da situação em que ela se encontra.

Figura 35: Estrutura da informação de contexto após ser aplicada à ontologia

ID	Data	Hora	Localização	Descrição
----	------	------	-------------	-----------

Fonte: Elaborada pelo autor.

Até este ponto foi abordado o planejamento e definição da estrutura e funcionalidades do Onisciente. No capítulo subsequente será tratado como ocorreu a implementação do modelo apresentado.

5 IMPLEMENTAÇÃO

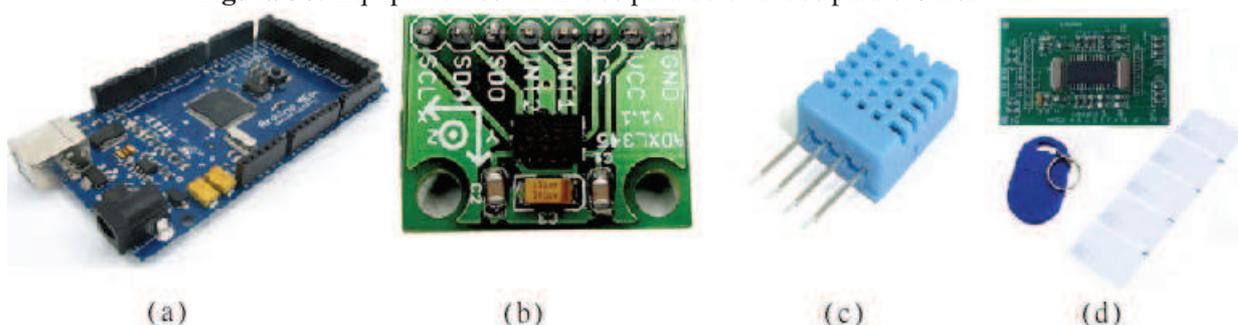
Neste capítulo é feito um relato de como foi realizada a implementação do Onisciente segundo o modelo especificado no capítulo 4. O capítulo inicia descrevendo as tecnologias utilizadas, seguido do desenvolvimento do servidor, descrevendo, camada a camada, suas funcionalidades mais relevantes. Por fim, são descritos como foram implementados os clientes web e Android nos mesmos moldes utilizados na apresentação do servidor.

5.1 Tecnologias Utilizadas

Para o desenvolvimento do servidor e site Web do Onisciente foi utilizado a linguagem C# com a ferramenta Visual Studio 2012, o banco de dados foi elaborado em SQL Server 2008 R2 e para gerenciar a conexão entre o SQL Server e o Visual Studio foi utilizado o ADO.NET Entity Data Model. Este recurso cria uma camada de objetos no projeto do Visual Studio que reflete a estrutura modelada no banco de dados e fornece as classes que serão utilizadas na aplicação. Para interagir com esta camada de objetos foi utilizado LINQ⁵. A biblioteca Newtonsoft Json⁶ foi utilizada para serializar dados para a troca de mensagens com os clientes Android e Web.

O desenvolvimento do cliente Android foi realizado em Java, utilizando Eclipse com suporte ao ADT (Android Development Tool). A biblioteca Gson⁷ (Google Json) foi utilizada para auxiliar na troca de mensagens com o servidor. Esta biblioteca é equivalente ao JSON utilizado no Visual Studio.

Figura 36: Equipamentos utilizados para obter dados para o Onisciente

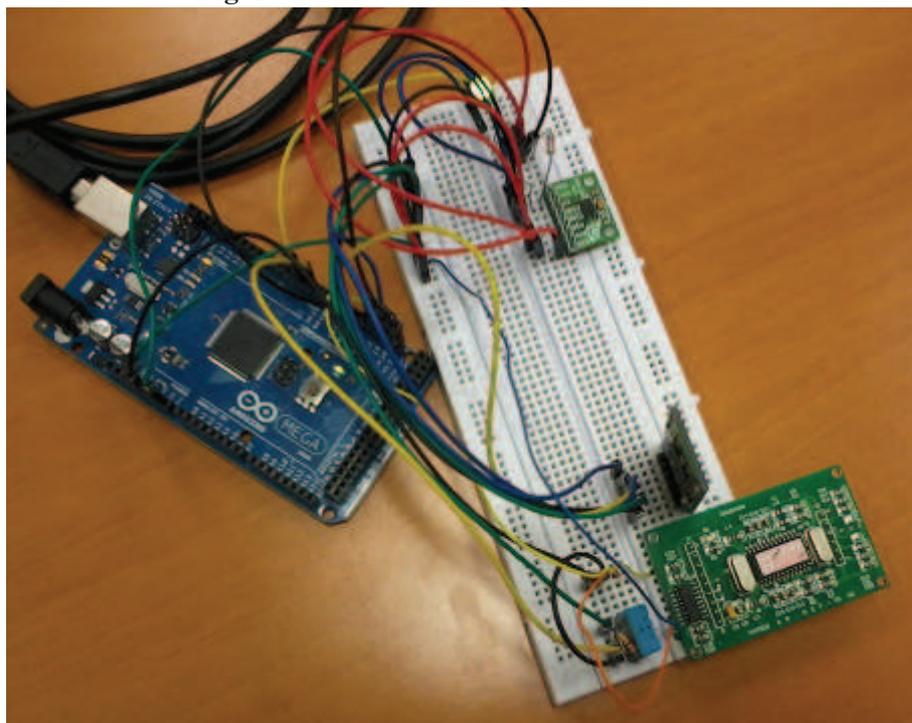


Fonte: Elaborada pelo autor.

Para a coleta de informações foram utilizados equipamentos disponíveis no Mobilab⁸ (Laboratório de Computação Móveis e Ubíqua da Unisinos). O principal dispositivo empregado consistiu de um hardware composto por um Arduino Mega 2560 (Figura 36.a) acoplado com uma leitora RFID de 13,56 MHz (Figura 36.d). Essa leitora pode realizar captura de informações a uma distância de até seis centímetros de uma tag, suporta ISO14443A/MIFARE e identifica automaticamente a presença da tags sem a necessidade de comandos. Esse equipamento também é composto de um sensor DHT-11 (Figura 36.c) para obter informações de temperatura e umidade, e um acelerômetro de 3 eixos ADXL345 (Figura 36.b), que é capaz de sentir

movimento ou inatividade de entidades, assim como medir movimento ou detectar colisões. O equipamento resultante da combinação destes elementos pode ser visualizada na Figura 37. Um celular rodando o sistema operacional Android foi utilizado como intermediário para realizar o envio dos dados obtidos para o Servidor. Ele se comunica com o arduino por meio de uma conexão Bluetooth e sockets com o servidor. Este procedimento foi necessário, pois o arduino não possuía meios de enviar os dados por falta de um módulo GPRS.

Figura 37: Hardware utilizado no Onisciente



Fonte: Elaborada pelo autor.

5.2 Servidor

O servidor foi desenvolvido conforme a especificação apresentada no modelo. Foram implementadas as quatro camadas propostas, sendo uma composta por elementos de hardware e o restante por software. Foi desenvolvido utilizando orientação a objetos em uma arquitetura orientada a serviços. Seu foco está em gerar contextos baseado na análise de dados de sensores explicitamente envolvendo temperatura, umidade e gravidade - o resultado do uso de um acelerômetro.

A seguir são detalhadas as classes empregadas no desenvolvimento do servidor.

⁵<http://msdn.microsoft.com/pt-br/library/vstudio/bb397926.aspx>

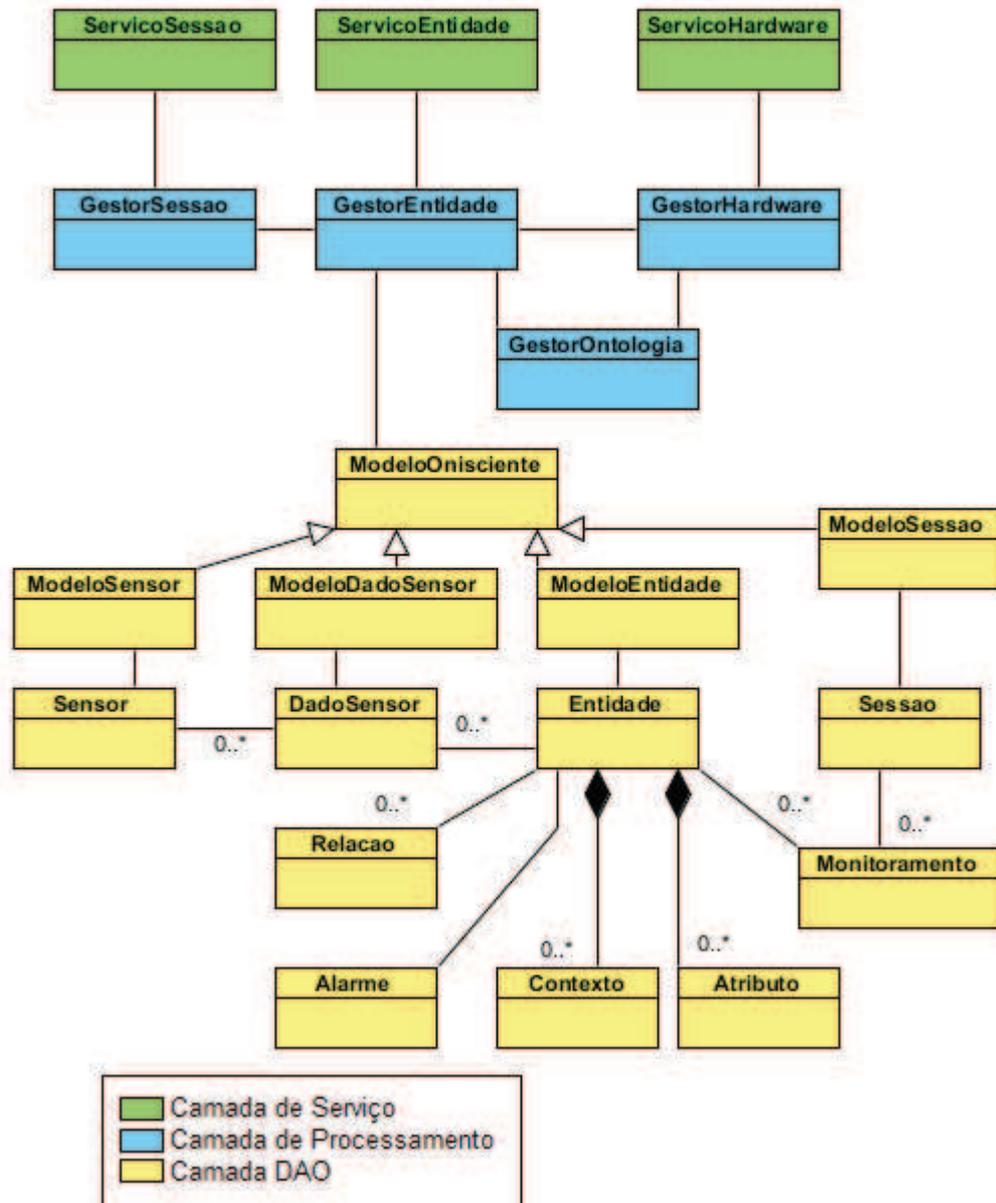
⁶<http://james.newtonking.com/projects/json-net.aspx>

⁷<https://code.google.com/p/google-gson/>

⁸<http://www.unisinos.br/lab/mobilab/>

5.2.1 Classes do Servidor

Figura 38: Diagrama de classes do servidor



Fonte: Elaborada pelo autor.

Como resultado da implementação, um total de 23 classes foram criadas para atender o objetivo proposto para o Onisciente. A Figura 38 apresenta o diagrama de classes, no qual elas estão distribuídas da seguinte forma: 3 na Camada de Serviços para interação com hardware, web sites e dispositivos móveis; 4 na Camada de Processamento para realizar a gestão de dados conforme o tipo; 15 na Camada DAO, sendo 5 para gerir as entidades mapeadas pelo banco de dados e as 10 restantes para as entidades mapeadas propriamente ditas.

Uma breve descrição de cada classe, separada por cada tipo de camada, pode ser vista a seguir.

Na Camada de Serviços estão presentes as seguintes classes:

- **ServicoHardware:** fornece um conjunto de funcionalidades que permite com que um equipamento equipado com algum tipo de sensor possa interagir com o Onisciente por meio de uma conexão Socket;
- **ServicoEntidade:** fornece um conjunto de serviços REST para dispositivos móveis e sites web que permitem a inclusão, exclusão ou atualização de entidades junto ao servidor. Também fornecem diversas funções de consulta para esses dispositivos;
- **ServicoSessao:** fornece um conjunto de funcionalidades por meio de serviços REST para a criação de sessões de monitoramento além de outras funcionalidades para consulta de contextos de entidades.

A Camada de Processamento contém as classes:

- **GestorEntidade:** esta classe é a responsável pelos procedimentos que realizam alguma operação sobre entidades, sejam elas inclusões, edições ou consultas;
- **GestorOntologia:** realiza as operações relacionadas a ontologia, inserindo novos elementos, atualizando informações de propriedades e criando os contextos de entidades;
- **GestorHardware:** é responsável por validar os dados recebidos dos sensores, classificando e enviando estes dados para os demais gestores do sistema persistirem a informação no Onisciente;
- **GestorSessao:** esta classe fornece os recursos necessários para controlar as sessões de usuários.

Na Camada DAO estão inseridas as classes:

- **ModeloOnisciente:** classe abstrata que contém os atributos e métodos básicos necessários para a manipulação de entidades;
- **ModeloSensor:** classe herdada da ModeloOnisciente que é utilizada para realizar a manipulação de entidades do tipo Sensor;
- **ModeloDadoSensor:** classe herdada da ModeloOnisciente que é utilizada para realizar a manipulação dados de sensores;
- **ModeloEntidade:** classe herdada da ModeloOnisciente que é utilizada para realizar a manipulação de entidades de forma geral;
- **ModeloSessao:** classe herdada da ModeloOnisciente que é utilizada para realizar a manipulação de sessões de monitoramento;

- **Alarme:** indica alarmes das entidades;
- **Atributo:** representa os atributos das entidades;
- **Contexto:** contém as informações de contexto de uma entidade;
- **Entidade:** representa uma entidade no sistema;
- **Monitoramento:** relaciona as entidades que uma sessão está monitorando;
- **Relacao:** indica uma relação entre duas entidades;
- **Sensor:** representa um sensor cadastrado no sistema;
- **DadoSensor:** contém dados de informação enviados por um sensor;
- **Sessao:** representa uma sessão existente no sistema;
- **Usuario:** representa um usuário do sistema;

A seguir, é feita uma descrição das funcionalidades implementadas no servidor, começando pelo processo de aquisição de dados, interação com a camada física, controle de entidades identificadas, ontologia, gestão da ontologia, geração de contextos descrição de contextos e o banco de dados.

5.2.2 Aquisição de Dados

A aquisição dos dados é realizada requisitando aos sensores, em intervalos regulares de tempo, que façam uma captação de informação. Conforme o tipo de sensor, os dados são formatados para em seguida serem enviados ao servidor do Onisciente. Dados de temperatura e umidade são enviados em conjunto em uma única mensagem de 11 bytes. Informações captadas pela leitora RFID são enviadas em uma mensagem com 23 bytes. As informações do acelerômetro que remetem as forças aplicadas nos 3 eixos são formatada e enviadas em mensagens que possuem entre 14 e 20 bytes e por fim, informações de velocidade de deslocamento são transmitidas em 5 bytes. A informação de velocidade não é obtida diretamente de um sensor, ela é adquirida processando os dados do acelerômetro para, após este procedimento, ser enviada ao servidor.

O processo de aquisição de dados é realizado com a utilização de um dispositivo móvel como elemento intermediário. O equipamento que realiza a coleta de dados utiliza uma conexão Bluetooth para realizar comunicação com outros dispositivos. Logo, um dispositivo móvel executando uma aplicação Android é utilizado para fazer a comunicação entre o servidor e este equipamento. Esta comunicação é feita por meio de uma conexão socket através do protocolo HTTPS.

A aplicação desenvolvida fica aguardando dados do equipamento, prepara a informação recebida no formato que o servidor do Onisciente está esperando e envia os dados por meio da conexão socket. O formato esperado pelo serviço de hardware da camada de serviço é: identificação do sensor#valor do dado do sensor para informações de temperatura, umidade e velocidade e identificação do sensor#valor do eixo x#valor do eixo y#valor do eixo z para dados do acelerômetro.

5.2.3 Interação com a camada física

Para fornecer um meio de adquirir os dados provenientes dos sensores sem atrapalhar o desempenho do servidor, uma *thread* é inicializada junto com a execução do servidor especialmente para coordenar as conexões Sockets com os dispositivos que estiverem enviando dados. Esta *thread* fica indefinidamente aguardando novas informações e quando as recebe, executa uma rápida avaliação da consistência da informação, para em seguida repassá-la para a camada de processamento para que seja finalmente processada e persistida no sistema.

5.2.4 Controle de entidades monitoradas

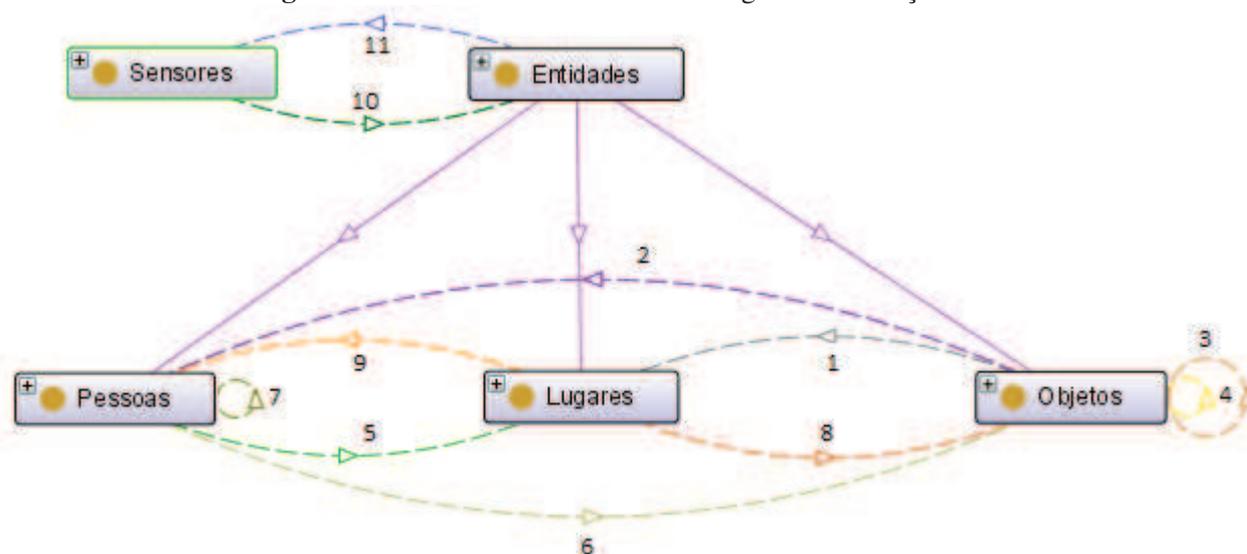
Uma das responsabilidades da camada de processamento é manter o controle das entidades que foram identificadas em algum momento da execução do servidor, e também definir quando uma entidade deve deixar de receber as informações que estão sendo recebidas através da camada física. Para isso, a partir do momento em que uma leitora RFID realiza a identificação de uma entidade, ela é inserida em uma lista que também contém o momento em que a entidade foi identificada. Esse momento serve como controle de tempo de vida da entidade. Se em um determinado período de tempo não houver uma nova identificação da entidade a partir da leitora RFID, a entidade será removida dessa lista. Durante o tempo que uma entidade estiver presente em uma lista, toda informação que for adquirida pelos sensores do mesmo equipamento que a identificou lhe serão aplicados.

Os dados de sensores de um hardware poderão exercer algum tipo de efeito apenas para as entidades que estiverem na lista e a aplicação desses dados sobre as entidades é realizada através da ontologia.

5.2.5 Ontologia

No Onisciente, a ontologia é utilizada no processo de avaliação e inferência de informações que geram os contextos das entidades nela existentes. No caso do Onisciente, *Pessoas*, *Objetos* e *Lugares*. Ela foi desenvolvida com a ferramenta Protegé⁹ em OWL num padrão conhecido como RDF/XML e implementada em C# com o auxílio de uma biblioteca open source, que

Figura 39: Classes de entidade da ontologia e suas relações



Fonte: Elaborada pelo autor.

interpreta esse tipo de estrutura, denominada dotNetRDF¹⁰.

A ontologia foi elaborada para gerar contextos baseado nos tipos de informações definidas na especificação do modelo (temperatura, umidade e gravidade). Para isso, as entidades definidas na ontologia necessitam obrigatoriamente ter algumas propriedades que possuam algum valor relacionado com os sensores existentes para comparação.

A Figura 39 exhibe as classes de entidades e as relações existentes entre as subclasses destas classes. Uma entidade da classe Objeto não possui umidade ou temperatura próprias, mas pode especificar limites, através de atributos, para essas características. Uma Localidade por outro lado, pode possuir valores de temperatura e umidade e os elementos que estiverem dentro dela podem sofrer algum efeito. Esse tipo de relação tem de ser especificada na ontologia para que o processo de inferência seja feito corretamente. As relações existentes entre as subclasses da entidade são descritas a seguir:

1. **ESTA_INSERTIDO:** Esta relação indica que um objeto está inserido em um lugar;
2. **POSSUI_PROPRIETARIO:** Relação que indica que um objeto possui um proprietário;
3. **COMPOE_OBJETO:** Indica que um objeto faz parte de outro objeto;
4. **E_COMPOSTO_POR:** Esta relação indica que um objeto é composto por outro objeto;
5. **ESTA_NO_LOCAL:** Indica que uma pessoa está em um lugar;
6. **E_PROPRIETARIO:** Relação que indica que uma pessoa é dona de um objeto;

⁹<http://protege.stanford.edu/>

¹⁰<http://www.dotnetrdf.org/>

7. **CONHECE**: Relação que indica que uma pessoa conhece uma determinada pessoa;
8. **POSSUI_OBJETO**: Esta relação indica que um lugar possui um objeto inserido em si;
9. **POSSUI_PESSOA**: Indica que uma pessoa está dentro de um local;
10. **ESTA_SENSORIANDO**: Esta relação indica que um sensor está monitorando uma entidade;
11. **POSSUI_SENSOR**: Indica que uma entidade tem um sensor lhe monitorando.

Como citado anteriormente, existem algumas propriedades obrigatórias para garantir a inferência dos contextos conforme o tipo de entidade em questão. A subclasse de *Objetos* é a que mais apresenta propriedades para avaliação, são elas:

- **possuiValorTemperaturaMínima**: Determina o valor mínimo de temperatura que o objeto pode ficar exposto;
- **possuiValorTemperaturaMáxima**: Determina o valor máximo de temperatura que um objeto pode ficar exposto;
- **possuiValorUmidadeMínima**: Determina o valor mínimo de umidade que o objeto pode ficar exposto;
- **possuiValorUmidadeMáxima**: Determina o valor máximo de umidade que o objeto pode ficar exposto;
- **possuiResistência**: Determina o valor máximo de força que pode ser aplicada sobre o objeto antes que ele sofra danos materiais.
- **possuiPeso**: Especifica o peso do objeto. É necessário para poder determinar contextos que envolvem vibração e impacto.

Todas essas propriedades são utilizadas como parâmetros de comparação com as propriedades de um local para a geração de contextos. A única exceção é *possuiResistencia* que pode ser utilizada de forma direta quando há informação de acelerômetro envolvido.

A subclasse *Lugares* contém apenas duas propriedades, porém são as que apresentam maior importância entre as existentes, elas são os pontos de referências para comparações entre as entidades na hora de gerar os contextos:

- **possuiTemperatura**: Esta propriedade indica qual é a temperatura do local, este valor será a base de comparação para os delimitadores de temperatura dos objetos;
- **possuiUmidade**: Esta é outra propriedade do local que é utilizada como base para comparativos dos limites dos objetos e também para as pessoas que estiverem no local;

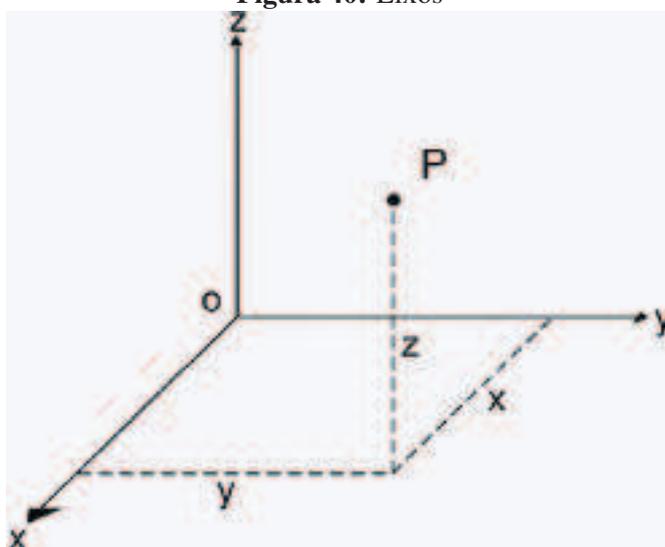
Por fim, a subclasse *Pessoas* possui apenas as propriedades **possuiTemperatura**, que indica a temperatura do próprio indivíduo e é alimentada por dados de sensores, e a propriedade **sensacaoTermica**, que é obtida através da combinação de dados como velocidade de deslocamento (capturada pelo acelerômetro) e temperatura. Ao contrário das entidades de *Lugares*, o valor da propriedade **possuiTemperatura** não é comparado com a de demais entidades, mas sim com valores predefinidos que servem para garantir as condições de saúde de uma pessoa.

Classes de *Objetos* e *Pessoas* possuem ainda propriedades comuns a ambas como **possuiVelocidade** e **sofreuImpacto**, onde a primeira é adquirida através do acelerômetro e a segunda é utilizada para registrar a força do impacto sofrido pela entidade.

As três subclasses de entidade também possuem atributos herdados da classe Entidade para a utilização com o acelerômetro, elas indicam um valor de força G que tenha sido aplicado sobre a entidade no eixo ao qual elas nomeiam. A Figura 40 ilustra estes eixos. Os atributos são:

- **possuiValorEixoX**: Indica um valor de força G aplicado sobre a entidade no eixo X;
- **possuiValorEixoY**: Indica um valor de força G aplicado sobre a entidade no eixo Y;
- **possuiValorEixoZ**: Indica um valor de força G aplicada sobre a entidade sobre o eixo Z.

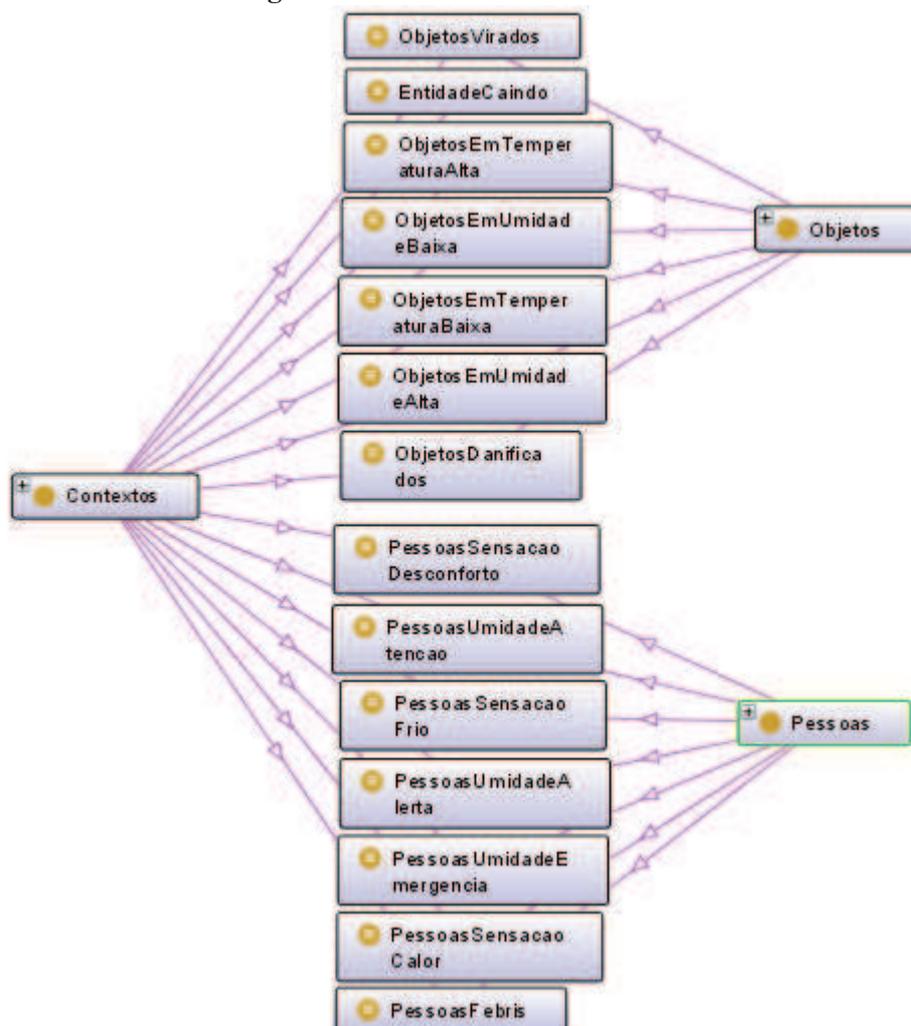
Figura 40: Eixos



Fonte: Elaborada pelo autor.

A classe *Entidade* também possui um atributo que é utilizado de forma dinâmica durante a execução do Onisciente, o atributo **estaSendoMonitorado** é inserido e removido das entidades conforme elas são identificadas por uma leitora RFID ou seu tempo de vida expira. Este atributo serve para que a ontologia saiba que uma entidade deve, ou não, ser avaliada para a geração de contextos. Ele é aplicável a qualquer tipo de entidade porque um acelerômetro pode ser atribuído para qualquer tipo.

Figura 41: Contextos do Onisciente



Fonte: Elaborada pelo autor.

Da mesma maneira que foram definidas subclasses de Entidade, existem também algumas subclasses de Contexto (Figura 41). Cada subclasse de contexto representa uma situação não favorável para um determinado tipo de entidade. Portanto, se o processo de geração de contexto não classificar uma entidade em nenhuma subclasse de contexto, esta entidade estará em condições consideradas satisfatórias. Os contextos definidos estão basicamente separados em dois tipos, os que atuam sobre *Objetos* e os que atuam sobre *Pessoas*. Ambos os tipos levam em consideração os atributos de cada entidade para definir a sua população, seja comparando esses atributos com algum valor predeterminado, como é o caso da maioria dos contextos relacionados a pessoas, ou com atributos do local onde estão inseridos, que é o caso dos contextos de objetos.

O Onisciente foi concebido para trabalhar com dois níveis de contexto, a nomenclatura destes níveis segue a denominação de Adelstein F.; Golden (2005). O nível primeiro é composto por contextos de baixo-nível que são gerados a partir de um único parâmetro, por exemplo, temperatura ou umidade. O segundo nível é denominado de alto-nível e utiliza a combinação de

contextos de baixo-nível ou um contexto de baixo-nível acrescido de uma nova informação para gerar contextos mais complexos. Por exemplo, a combinação de temperatura e umidade que separadamente poderiam influenciar negativamente uma entidade, mas combinadas não apresentam nenhum tipo de risco a entidade. Ou ainda, a combinação de um contexto de temperatura com deslocamento, onde a temperatura causava uma condição desfavorável para a entidade, mas o acréscimo da informação de que ela estava em movimento anulava essa condição.

Os contextos de objetos definidos para o Onisciente são o resultado da comparação de dados de temperatura ou umidade de locais com os limites impostos por suas propriedades. Em alguns casos, informações sobre movimentação, vibração ou impacto também são utilizadas. São eles:

- **ObjetosDanificados:** esse contexto leva em consideração a propriedade possuiResistência e lista apenas os objetos que sofreram uma força maior do que poderiam suportar;
- **ObjetosEmTemperaturaAlta:** esse contexto analisa a propriedade possuiValorTemperaturaMáxima do objeto e lista aqueles que estão expostos a uma temperatura muito elevada;
- **ObjetosEmTemperaturaBaixa:** esse contexto compara a propriedade possuiValorTemperaturaMinima do objeto com a possuiTemperatura do local onde ele está e lista aqueles que estão abaixo do indicado;
- **ObjetosEmUmidadeAlta:** esse contexto lista os *Objetos* que estão expostos a uma umidade superior ao especificado na propriedade possuiValorUmidadeMaxima;
- **ObjetosEmUmidadeBaixa:** esse contexto avalia o valor da propriedade possuiValorUmidadeMinima e lista os objetos que estão abaixo da umidade ideal;
- **ObjetosVirados:** contem todas as entidades que estejam situadas a um ângulo de 90° da posição ideal;
- **EntidadeCaindo:** contém as entidades que estejam sobre ação da gravidade próxima a 0.

Os contextos de pessoas são definidos comparando as propriedades das pessoas ou do local onde elas estão com valores predeterminados, nenhum contexto utilizando acelerômetro foi definido para o Onisciente. Os contextos definidos são:

- **PessoasFebris:** Esse contexto lista todas as pessoas cuja temperatura esteja fora do intervalo que varia entre 36 e 37°C.
- **PessoasUmidadeAlerta:** Neste contexto as pessoas que estiverem em um ambiente onde a umidade está avaliada entre 12 e 20% são listadas;
- **PessoasUmidadeAtencao:** Este contexto mostras as pessoas que estão em um ambiente com umidade entre 20 e 30%;

- **PessoasUmidadeEmergencia:** Todas as pessoas em um local com umidade abaixo de 12% são listadas aqui;
- **PessoasSensaçãoFrio:** Todas as pessoas que estão com sensação de frio em uma determinada localidade. Baseado em temperatura ou sensação térmica e umidade;
- **PessoasSensaçãoCalor:** Todas as pessoas que, dentro de um local estão com sensação de calor. Baseado em temperatura ou sensação térmica e umidade;
- **PessoasSensaçãoDesconforto:** Todas as pessoas que estão dentro de uma determinada localidade, que não estão se sentindo confortáveis, mas nem com sensação de frio ou calor;

Após ter-se adquirido conhecimento sobre a estrutura da ontologia e suas peculiaridades, descrevemos na subseção seguinte como é realizado a gestão da ontologia.

5.2.6 Gestão da Ontologia

A gestão da ontologia envolve aplicar novas informações obtidas às entidades existentes que estejam relacionadas com esta informação. Por exemplo, ao obtermos um valor de temperatura de um local, o atributo de temperatura deste local deve ser atualizada logo que a informação esteja presente no sistema, com isso garante-se a consistência da informação ao realizar o processo de geração de contexto.

Entre as funções da gestão de ontologia estão:

- **Adicionar uma nova entidade:** quando uma nova entidade é cadastrada ela é inserida na ontologia conforme o seu tipo. Por exemplo, ao cadastrarmos um livro no sistema, ele será inserido como um tipo da sub-classe *Movel*, pois é um subtipo de *Bem Durável* que por sua vez é um subtipo de *Objeto*;
- **Inserir/Atualizar dados de atributos:** quando a informação de um sensor está para ser inserida na ontologia, o Onisciente verifica qual entidade está relacionada com este sensor para que ela possa receber a informação. No caso de uma atualização de informação, o Onisciente verifica no início do procedimento se o atributo já não está inserido na entidade. Caso ele já exista, este é removido para a inclusão da nova informação. Isto é feito desta forma porque a biblioteca utilizada para manipular a ontologia não permite que seja realizada a atualização do dado diretamente, logo, o novo dado é inserido como uma nova informação e a antiga é removida para não haver atributos duplicados;
- **Remover atributos:** em algumas situações há a necessidade de incluir ou remover completamente um atributo de uma entidade da ontologia, este é o caso do atributo “*estaSendoMonitorado*” que só é inserido nas entidades que foram identificadas por uma leitora RFID e que deve ser removido quando deixam de ser encontradas.

De posse das informações de como é realizada a gestão da ontologia, é descrita na próxima subseção como é realizado o processo de geração de contexto.

5.2.7 Geração de Contexto

A geração de contexto é executada toda vez que ocorre uma alteração de estado na ontologia. Esta alteração de estado pode ser a inclusão de uma nova entidade ou a alteração de algum atributo. O procedimento executado para a criação de contextos segue a seguinte ordem:

1. Todos os registros de contextos existentes são removidos da ontologia para garantir que entidades que deixaram de atender algum critério de contexto não fiquem registradas como “sujeira”;
2. Para cada tipo de contexto definido na ontologia é realizada uma consulta SPARQL para obter as entidades que atendem o critério especificado;
3. A ontologia é atualizada com os resultados obtidos.

Durante a elaboração da ontologia, descobriu-se que não é possível realizar comparações entre propriedades de classes diferentes. A solução encontrada foi a utilização de consultas SPARQL, que nos permitem fazer esse tipo de comparação. Uma segunda limitação que estimulou o uso das consultas SPARQL foi que a biblioteca utilizada para integrar a ontologia com o C# não interpretava a definição de algumas classes por não conseguir trabalhar com classes derivadas, que na ontologia são classes definidas a partir de critérios específicos, por exemplo, o Contexto *PessoasFebris* poderia ser definido com uma classe equivalente a pessoas que possuem temperatura abaixo de 36 e acima de 37 graus Celsius. Uma terceira limitação é a de que o *reasoner* que acompanha a biblioteca de ontologia se demonstrou muito simples e com restrições de funcionalidade, conseguindo apenas resolver a hierarquia de classes das entidades e não conseguindo trabalhar sobre as classes de equivalência nem as definições das propriedades de objetos definidas na ontologia. Para compensar esta última limitação, a compensação das propriedades foi implementada manualmente para que a ontologia ficasse consistente durante a sua execução. Esta implementação envolveu, nos procedimentos de inclusão e atualização de um atributo ou entidade na ontologia, verificar se as relações entre os elementos envolvidos não possuíam relações inversas. Por exemplo, é a relação entre *Objetos* e *Pessoas* onde um *Objeto* pertence a uma *Pessoa* ao mesmo tempo em que uma *Pessoa* possui um *Objeto*.

O resultado das consultas SPARQL é utilizado para popular os contextos de objeto ou pessoa. Um exemplo de consulta SPARQL pode ser vista na Figura 42.

Em alguns contextos de complexidade mais elevada como *PessoasSensacaoCalor*, *PessoasSensacaoFrio*, *PessoasSensacaoDesconforto* e *ObjetosDanificados*, é necessário que os dados que serão aplicados à ontologia sejam trabalhados antes de realizar a consulta SPARQL. No caso do contexto *ObjetosDanificados* o dado recebido informa a força g aplicada sobre a entidade.

Figura 42: Consulta SPARQL que cria o contexto de pessoa febril.

```
"CONSTRUCT { ?pessoas a ont:PessoasFebris } WHERE { "
+ "?pessoas ont:possuiTemperatura ?temperaturaPessoa. "
+ "?pessoas ont:estaNoLocal ?local "
+ "FILTER (?temperaturaPessoa < 36.0 || ?temperaturaPessoa > 37.0)}"
```

Fonte: Elaborada pelo autor.

A força g representa a aceleração gravitacional que a Terra exerce sobre um objeto e possui um valor de $9,8m/s^2$. Para o contexto em questão, este valor é utilizado para definir se uma entidade está sofrendo, no momento da medição, uma força superior à que foi definida como a sua resistência, portanto, apenas entidades que tiveram o atributo *possuiResistencia* definido podem ser consideradas para este contexto. Outra exigência para a definição deste contexto é a presença do atributo *possuiPeso*, que é necessário para calcular a massa da entidade. Esta massa é utilizada para calcular a força total que a entidade está sofrendo e utilizada para definir se a entidade em questão sofreu algum dano pela aplicação dessa força. O cálculo da força aplicada sobre uma entidade pode ser obtido por:

$$Força\ Total = Peso * força\ do\ eixo * 9,8 \quad (5.1)$$

E a definição se a entidade sofreu algum dano é definida por:

$$Valor\ de\ possuiResistencia \leq Força\ total \quad (5.2)$$

Onde se o resultado da comparação for verdadeiro, significa que o objeto sofreu dano pela aplicação da força.

Quanto aos contextos *PessoasSensacaoCalor*, *PessoasSensacaoFrio* e *PessoasSensacaoDesconforto*, ele é o resultado da combinação de dois tipos de informação, temperatura e umidade. Este contexto é obtido utilizando uma fórmula de índice de desconforto humano (ONO H.; KAWAMURA, 1991), também conhecida como índice de temperatura-umidade. O resultado dessa fórmula indica o impacto que a umidade efetua sobre a temperatura e qual o efeito disso sobre uma pessoa. A fórmula é dada por:

$$ITH = T - 0,55 * (1 - UR) * (T - 14) \quad (5.3)$$

Onde:

$$T = (Temperaturaem^{\circ}C * 1,8) + 32 \text{ e}$$

$$UR = UmidadeRelativa/100$$

O resultado obtido por esta fórmula pode ser classificado conforme a tabela 5 elaborada por Ono H.; Kawamura (1991) onde há uma classificação para cada intervalo de resultados.

Tabela 5: Classificação de desconforto humano

Intervalo	Classificação
ITH > 80	Calor
75 > ITH <= 80	Desconforto pelo calor
60 > ITH <= 75	Confortavel
55 > ITH <= 60	Desconforto pelo frio
ITH <= 55	Frio

Fonte: Ono H.; Kawamura (1991).

Em certas situações, para definir mais precisamente o índice de desconforto de uma pessoa, pode ser necessário utilizar, ao invés da temperatura adquirida por um sensor, a sensação térmica sentida por esta pessoa. Para isto, há necessidade de obter além da temperatura ambiente, a velocidade pela qual a pessoa está se deslocando.

Estes valores são utilizados para fazer o cálculo da sensação térmica da pessoa representada pela entidade. A sensação térmica^{11,12} é nada mais do que a temperatura sentida pela pele quando exposta ao vento. A fórmula para definir a sensação térmica (ou temperatura relativa) é dada por:

$$ST = 33 + (10\sqrt{V} + 10,45 - V) * \frac{T - 33}{22} \quad (5.4)$$

Onde:

V = Velocidade do vento em m/s (*Velocidade em Km/h/3,6*)

T = Temperatura

O resultado obtido com esta fórmula indica verdadeira temperatura sentida por uma pessoa, quanto maior a velocidade do vento e menor a temperatura, menor será o valor obtido. A Figura 43 apresenta um quadro com os valores de sensação térmica para algumas temperaturas sobre o efeito de variadas velocidades de vento.

5.2.8 Descrição de Contextos

Toda vez que as informações de contextos são requisitadas por uma sessão, o gestor de sessão requisita junto à ontologia o local, ao qual ela está relacionada, e todas as classificações de contexto em que a entidade está inserida. A partir disso, a descrição da situação da entidade é elaborada. Por exemplo, se uma entidade está situada em uma sala registrada como nome *Armazém* e também está classificada nos contextos *ObjetoTemperaturaAlta* e *ObjetoUmidadeBaixa*, sua descrição será dada como: “A entidade está inserida no local *Armazém* e está em um local de temperatura acima do ideal e está em um local com umidade abaixo do ideal”. Quanto mais classificações de contexto ela estiver inserida, maior e mais detalhada será a descrição obtida.

¹¹<http://www.sofisica.com.br/conteudos/curiosidades/sensacaotermica.php>

¹²<http://www.hsw.uol.com.br/questao70.htm>

Figura 43: Quadro com valores de sensação térmica para valores de temperatura em graus °C

		VELOCIDADE DO VENTO																							
		7	11	14	18	22	25	29	32	36	40	43	47	50	54	58	61	65	68	72	76	79	83	86	90
Temp (°C)	TEMPERATURA CORRESPONDENTE (sensação térmica)																								
	-6	-7	-11	-14	-16	-18	-20	-21	-23	-24	-25	-26	-26	-27	-28	-28	-28	-29	-29	-29	-30	-30	-30	-30	-30
-5	-6	-10	-13	-15	-17	-19	-20	-21	-22	-23	-24	-25	-25	-26	-27	-27	-27	-28	-28	-28	-28	-28	-28	-28	-28
-4	-5	-9	-11	-14	-16	-17	-19	-20	-21	-22	-23	-23	-24	-24	-25	-25	-26	-26	-26	-26	-26	-27	-27	-27	-27
-3	-4	-8	-10	-13	-14	-16	-17	-18	-19	-20	-21	-22	-22	-23	-23	-24	-24	-24	-25	-25	-25	-25	-25	-25	-25
-2	-3	-6	-9	-11	-13	-15	-16	-17	-18	-19	-20	-20	-21	-22	-22	-22	-23	-23	-23	-23	-23	-23	-23	-23	-23
-1	-2	-5	-8	-10	-12	-13	-14	-16	-17	-17	-18	-19	-19	-20	-20	-21	-21	-21	-21	-22	-22	-22	-22	-22	-22
0	-1	-4	-7	-9	-10	-12	-13	-14	-15	-16	-17	-17	-18	-18	-19	-19	-19	-20	-20	-20	-20	-20	-20	-20	-20
1	0	-3	-5	-7	-9	-11	-12	-13	-14	-14	-15	-16	-16	-17	-17	-17	-18	-18	-18	-18	-18	-18	-18	-19	-19
2	1	-2	-4	-6	-8	-9	-10	-11	-12	-13	-14	-14	-15	-15	-16	-16	-16	-16	-17	-17	-17	-17	-17	-17	-17
3	2	-1	-3	-5	-6	-8	-9	-10	-11	-11	-12	-13	-13	-14	-14	-14	-15	-15	-15	-15	-15	-15	-15	-15	-15
4	3	0	-2	-4	-5	-6	-8	-8	-9	-10	-11	-11	-12	-12	-12	-12	-13	-13	-13	-13	-13	-14	-14	-14	-14
5	4	1	-1	-2	-4	-5	-6	-7	-8	-9	-9	-10	-10	-11	-11	-11	-11	-12	-12	-12	-12	-12	-12	-12	-12
6	5	3	1	-1	-3	-4	-5	-6	-6	-7	-8	-8	-9	-9	-9	-10	-10	-10	-10	-10	-10	-10	-10	-10	-11
7	6	4	2	0	-1	-2	-3	-4	-5	-6	-6	-7	-7	-7	-8	-8	-8	-8	-9	-9	-9	-9	-9	-9	-9
8	7	5	3	1	0	-1	-2	-3	-3	-4	-5	-5	-5	-6	-6	-6	-7	-7	-7	-7	-7	-7	-7	-7	-7
9	8	6	4	3	1	0	-1	-1	-2	-3	-3	-4	-4	-4	-5	-5	-5	-5	-5	-5	-6	-6	-6	-6	-6
10	9	7	5	4	3	2	1	0	-1	-1	-2	-2	-2	-3	-3	-3	-3	-4	-4	-4	-4	-4	-4	-4	-4
11	10	8	7	5	4	6	2	2	1	0	0	-1	-1	-1	-1	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
12	11	9	8	6	5	4	4	3	2	2	1	1	1	0	0	0	0	-1	-1	-1	-1	-1	-1	-1	-1
13	12	10	9	8	7	6	5	4	4	3	3	3	2	2	2	1	1	1	1	1	1	1	1	1	1
14	13	12	10	9	8	7	6	6	5	5	4	4	4	3	3	3	3	3	3	3	3	2	2	2	2
15	15	13	12	11	10	9	9	8	7	7	6	6	6	5	5	5	5	4	4	4	4	4	4	4	4
16	16	14	13	12	11	10	9	9	8	8	7	7	7	7	6	6	6	6	6	6	6	6	6	6	6
17	17	15	14	13	12	11	11	10	10	9	9	9	8	8	8	8	8	8	8	7	7	7	7	7	7
18	18	16	15	14	13	13	12	12	11	11	10	10	10	10	10	10	9	9	9	9	9	9	9	9	9
19	19	17	16	15	15	14	13	13	13	12	12	12	11	11	11	11	11	11	11	11	11	11	10	10	10
20	20	18	17	17	16	15	15	14	14	14	13	13	13	13	13	13	12	12	12	12	12	12	12	12	12

Fonte: INMET - Instituto Nacional de Meteorologia

Fonte: INMET, 2013.

5.2.9 Banco de Dados

O banco de dados é composto por dez tabelas, conforme diagrama E-R apresentado na Figura 44. A principal tabela é a *Entidade*. Ela contém o registro de todas as entidades cadastradas no sistema e está relacionada com a tabela de *Atributos*, que contém todas as características de cada entidade. Outra tabela relacionada é a de *Relação*, que indica com quais entidades uma entidade se relaciona além de descrever qual o tipo existente entre elas. Ainda relacionada com a tabela de entidades, estão a tabela *Contexto* que contém as descrições das situações em que uma entidade esteve, e a tabela *Alarme*, que possui as indicações de quando uma entidade esteve em uma situação considerada crítica. A tabela *Monitoramento* serve para manter uma relação entre as sessões criadas pelos usuários e as entidades. Os usuários existentes estão cadastrados na tabela *Usuario*, enquanto as sessões existentes estão registradas na tabela *Sessao*.

As últimas tabelas do banco são a *Sensor* que possui o cadastro dos sensores do sistema, e a *SensorData* que mantém um histórico dos dados adquiridos pelos sensores que ocasionaram

Figura 44: Modelo entidade-relacionamento do banco de dados do onisciente



Fonte: Elaborada pelo autor.

em um efeito sobre uma entidade.

O banco de dados é utilizado para manter um histórico de dados obtidos pelos sensores, além de armazenar os registros de contextos gerados durante o funcionamento do Onisciente. O armazenamento destas informações abre um precedente para o estudo de trilhas (SILVA et al.,

2010), um tópico que não é abordado neste trabalho. A ontologia, por sua vez, mantém apenas os dados mais recentes, o suficiente para poder realizar a geração de novos contextos.

Até este ponto este capítulo abordou como foi realizada a implementação do servidor do Onisciente. A partir de agora o foco estará concentrado nos clientes que interagem com o servidor.

5.3 Clientes

As duas opções de clientes, web e Android, são basicamente terminais de uso, onde o usuário irá solicitar uma ação, esta ação irá gerar uma requisição para o servidor e este a processará gerando um retorno que será apresentado ao usuário. Não há a realização de processamento no lado dos clientes, a não ser por pequenas validações nos dados que forem preenchidos para garantir que não se consuma largura de banda enviando mensagens desnecessárias entre o cliente e o servidor. Ambos os clientes utilizam os mesmos serviços disponíveis pelo servidor e ficam responsáveis por tratar a informação que vem serializada em JSON conforme a sua necessidade.

Ambos os clientes desenvolvidos apresentam o mesmo comportamento para a requisição das informações de contexto. No cliente Android, um timer realiza a chamada do serviço do servidor em um período de tempo pré-determinado. Já no cliente web, uma função AJAX é responsável por realizar esta chamada. Este ciclo é quebrado apenas se houver uma mudança na lista de entidades a serem monitoradas, tanto ao adicionar, tanto ao remover entidades da lista. Quando isto ocorre, a requisição é feita de forma automática e o timer é reiniciado.

No decorrer desta seção será realizada uma descrição de cada tipo de cliente, apresentando as classes que foram originadas pelo desenvolvimento dos modelos propostos e as interfaces que são visualizadas pelos usuários que também é por onde eles interagem com o servidor.

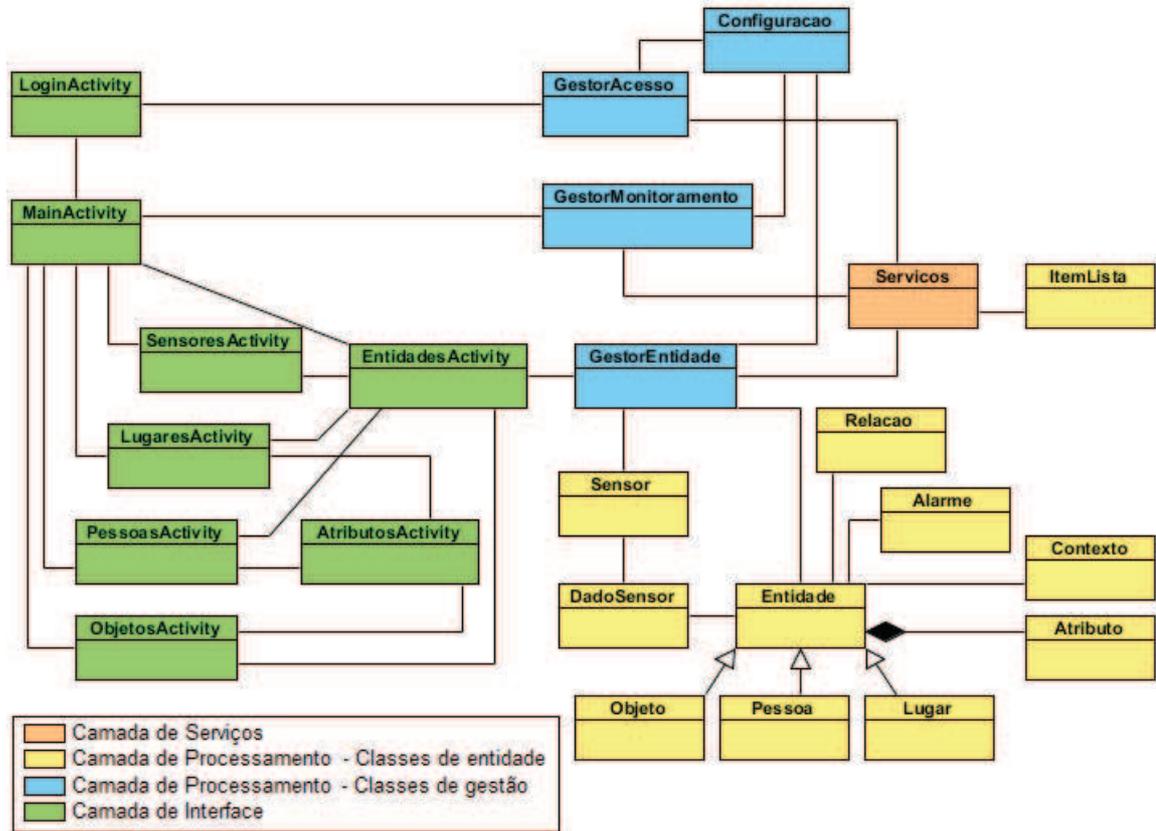
5.3.1 Cliente Android

O cliente Android foi desenvolvido seguindo as especificações definidas pelo modelo apresentado no Capítulo 4. A única exceção foi o processo de armazenamento local que permitiria com que um usuário pudesse consultar os contextos passados de entidades que tivessem sido monitoradas no passado. Com isso, arquitetura da aplicação reduziu o número de camadas de quatro para três, deixando o modelo muito similar ao especificado para o cliente web.

O desenvolvimento do cliente Android resultou na criação de 24 classes sendo que 8 estão relacionadas à camada de interface, representando as telas do sistema, 15 estão na camada de processamento e 1 na camada de serviços. Esta divisão pode ser acompanhada por meio da Figura 45 onde a camada de serviços está disposta pela cor vermelha, a camada de interface pela cor verde e a camada de processamento pelas cores amarelo e azul. As classes em azul representam as classes de gestão enquanto as em amarelo representam classes de estrutura. Cada classe da camada de interface representa a codificação de uma tela da aplicação, a especificação

de cada tela fica armazenada em um arquivo de extensão XML e para as telas que tiverem um menu, mais um arquivo contendo as suas informações.

Figura 45: Diagrama de classes do cliente Android



Fonte: Elaborada pelo autor.

A *Camada de Serviço* possui a classe **Serviços**, nesta classe estão concentrados os métodos que interagem por meio de *web services* REST com o servidor. As operações nela definidas realizam as operações de CRUD e buscam dados para a execução de monitoramento de entidades; Na *Camada de Processamento* estão presentes as classes:

- **GestorAcesso:** apresenta os métodos que fazem a validação do usuário com o servidor;
- **GestorEntidade:** fornece os métodos necessários para que um usuário possa adicionar, editar ou excluir entidades no Onisciente, essas entidades podem ser do tipo *Sensor*, *Local*, *Objeto* ou *Pessoa*;
- **GestorMonitoramento:** contém os métodos que permitem um usuário criar uma nova sessão de monitoramento, além de incluir ou remover elementos nessa seção;
- **Configuração:** nela estão inseridos alguns dados que podem ser utilizados pelas classes de gestão, que são essenciais para o funcionamento do cliente com o Onisciente;
- **ItemLista:** esta classe representa uma informação que será exibida para o usuário através de uma lista;

- **Sensor:** Classe que contém as informações de sensores recém-criados ou para edição pelo usuário;
- **Entidade:** Classe abstrata que define a base de todas as entidades existentes na aplicação;
- **Objeto:** classe herdada de Entidade que representa um objeto;
- **Pessoa:** classe herdada de Entidade que representa uma pessoa;
- **Local:** classe herdada de Entidade que representa uma localização;
- **Atributo:** representa uma característica de uma entidade;
- **Relação:** contém a informação de relação entre duas entidades, especificando o tipo de relação entre elas;
- **Alarme:** descreve uma situação crítica que tenha ocorrido com uma entidade. É preenchida apenas quando adquirida junto ao servidor do Onisciente para edição;
- **Contexto:** descreve uma situação de contexto ocorrida com a entidade ao qual está relacionada. É preenchida apenas quando adquirida junto ao servidor do Onisciente para edição;
- **DadoSensor:** contém uma informação adquirida por meio de um sensor que tenha exercido uma influência direta sobre a entidade. É preenchida apenas quando adquirida junto ao servidor do Onisciente para edição;

Como foi citado nas descrições acima, as classes *Alarme*, *Contexto* e *DadoSensor* somente possuem dados quando são consultadas junto ao servidor do Onisciente. Na Camada de Interface estão presentes as classes:

- **LoginActivity:** classe que possui a codificação da tela de login da aplicação;
- **MainActivity:** classe da tela de monitoramento da aplicação;
- **SensoresActivity:** classe da tela de cadastro e edição de sensores;
- **LugaresActivity:** classe com a codificação do cadastro e edição de lugares;
- **PessoasActivity:** classe com o código de cadastro e edição de pessoas;
- **ObjetosActivity:** classe com a codificação de cadastro e edição de objetos;
- **EntidadesActivity:** classe com a codificação da tela que exhibe listagens de entidades;
- **AtributosActivity:** classe com a codificação para a inclusão e edição de atributos para os cadastros de entidades da aplicação.

Apresentada a estrutura de classes do cliente Android, é apresentado a seguir as interfaces disponíveis ao usuário para interação com o servidor.

5.3.1.1 Interface do cliente Android

A interface do aplicativo Android foi construída para que o usuário possa ter uma experiência prática, realizando qualquer tarefa desejada em apenas alguns cliques. As telas elaboradas tendem a ter um aspecto limpo, exibindo apenas os componentes necessários para atender ao objetivo ao qual foram criadas, deixando as opções disponíveis aos usuários por meio de menus.

Figura 46: Tela de login, monitoramento, seleção de entidade a monitorar e seleção de nova entidade.



Fonte: Elaborada pelo autor.

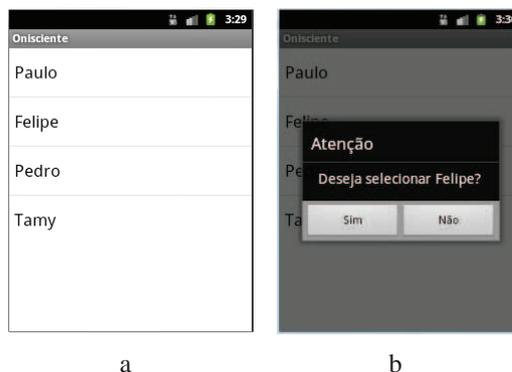
Na tela de login (Figura 46.a) estão presentes dois campos de texto e um botão de confirmação. O primeiro campo é para a especificação de um usuário válido e o segundo campo para a senha deste usuário. O botão dispara o evento para a realização de login do usuário no Onisciente.

A tela de monitoramento (Figura 46.b) é o ponto central de controle da aplicação Android, a partir dela pode-se chegar a qualquer outra tela da aplicação. Ela é composta por uma listagem de entidades selecionadas pelo usuário para que tenham seus contextos monitorados. A descrição destes contextos é inserida e atualizada em intervalos de tempo regulares, sempre com a descrição de sua situação mais recente. Um menu com as opções Monitorar, Remover e Adicionar está disponível onde a opção monitorar abre um menu secundário com uma listagem de opções (Figura 46.c) que representam os tipos de entidades presentes no Onisciente. Ao selecionar esta opção, o usuário será encaminhado para uma tela que conterá uma listagem com todas as entidades cadastradas para que realize a escolha de qual deseja monitorar. A opção remover retirará da lista de monitoramento a entidade que for selecionada pelo usuário antes de pressionar esta opção. A última opção disponível abre um menu secundário (Figura 46.d) com as opções de cadastro de entidades disponíveis.

Na tela de seleção de entidades (Figura 47.a) o usuário realiza a escolha de um elemento dentro de uma listagem de entidades, que deve ser confirmada através de uma caixa de mensagem (Figura 47.b).

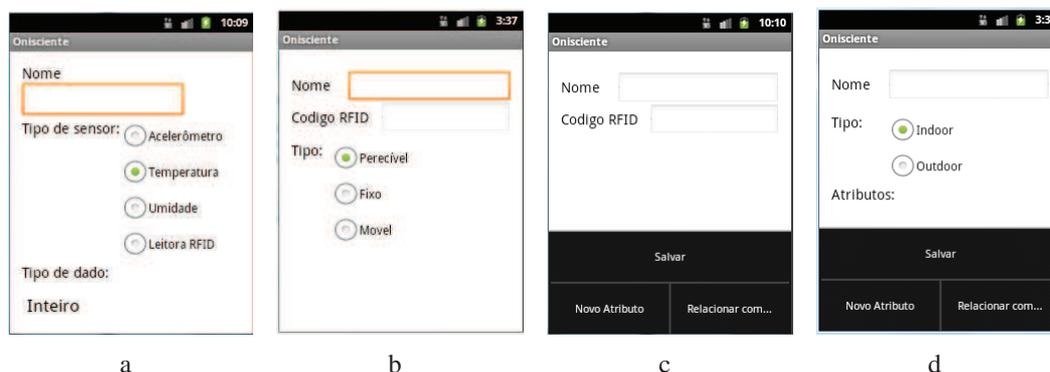
O cliente Android possui quatro telas de cadastro de entidades, a tela de cadastro de Sensores (Figura 48.a) é composta por um campo de texto para a definição de um nome para o sensor, um

Figura 47: Tela de seleção de entidades e confirmação de escolha.



Fonte: Elaborada pelo autor.

Figura 48: Telas de cadastro do cliente Android.



Fonte: Elaborada pelo autor.

conjunto de opções que definem o tipo do sensor e uma lista que contém o tipo de dado que o sensor irá fornecer para o Onisciente. A tela de cadastro de Objetos (Figura 48.b) é constituída por um campo texto para a inserção de um nome para o objeto, outro para a inserção do seu código RFID e um grupo de opções que determina o tipo de objeto que está sendo cadastrado. Na tela de cadastro de Pessoas (Figura 48.c) está presente um campo para a inserção do nome da pessoa que está sendo cadastrada e um para a inserção de um código RFID. Por fim, a tela de cadastro de Lugares (Figura 48.d) é composta por um campo de texto para a definição de uma identificação do local e um grupo de opções para a definição do tipo a qual ela pertence.

Com exceção da tela de cadastro de sensor, onde o usuário deve indicar a entidade que será monitorada (Figura 49.a), todas as outras telas possuem uma opção para o usuário realizar uma relação entre as entidades (Figura 49.b) e também uma opção de definir um novo atributo para a entidade que está sendo cadastrada (Figura 49.c).

Após a apresentação do cliente Android, o capítulo prossegue introduzindo o cliente web.

Figura 49: Telas de escolha de entidade a ser monitorada, definição de relação de entidades e adição de atributos.



Fonte: Elaborada pelo autor.

5.3.2 Cliente Web

O cliente web é composto por um website e é apresentado como um recurso para monitoramento de entidades para indivíduos que não possuam um dispositivo móvel que execute o sistema operacional Android. O cliente web utiliza os mesmos recursos de comunicação do cliente Android para interação com o servidor do Onisciente. A principal diferença é como as informações e os recursos são apresentados ao usuário, visto que, com mais espaço para apresentação de componentes, o site pode apresentar mais facilidade de uso para o usuário comparado ao cliente Android.

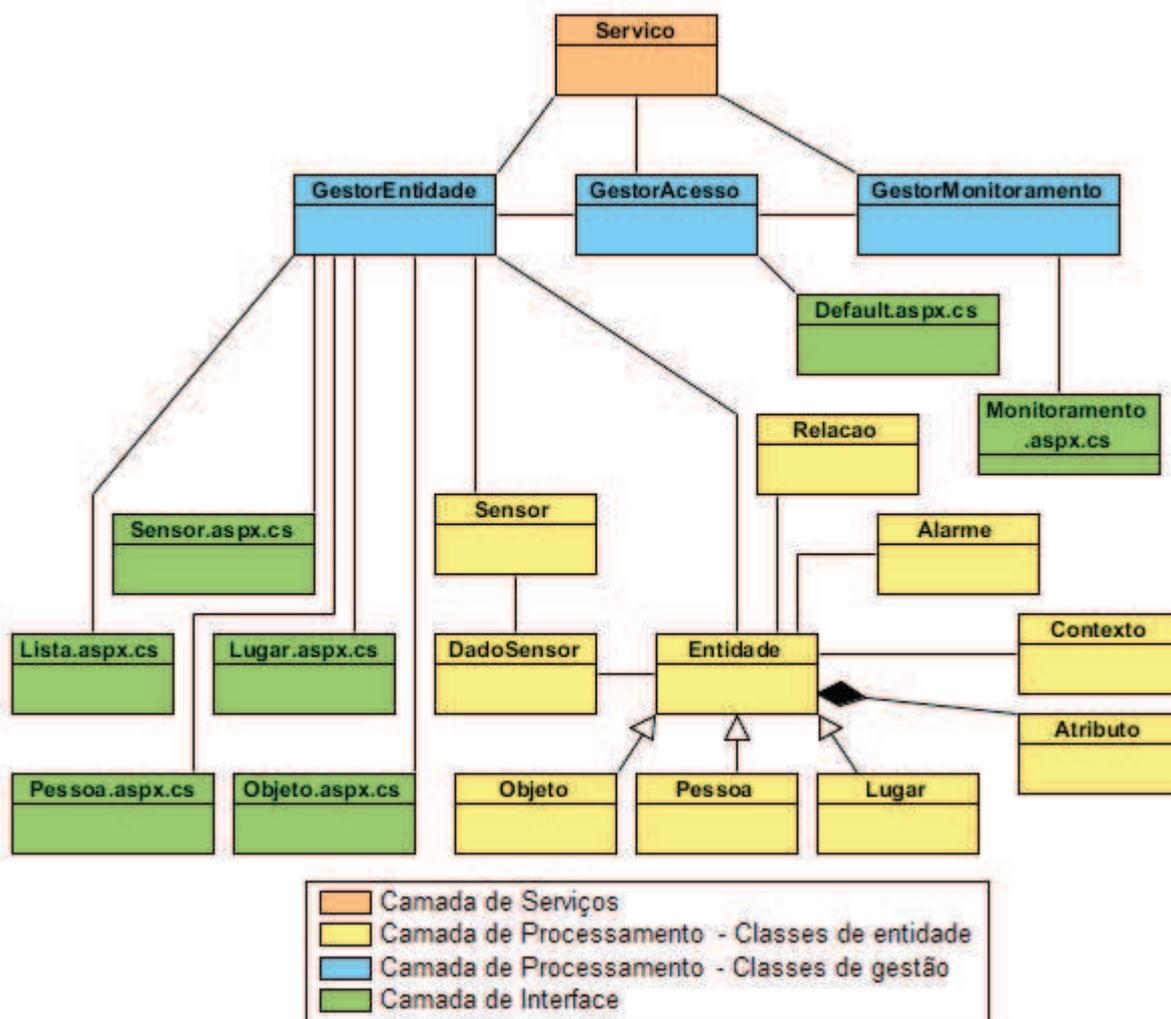
A implementação do cliente web seguiu o modelo especificado no capítulo 4 e gerou um conjunto de classes que pode ser visualizada na Figura 50. Em um total de 21 classes, onde uma está inserida na camada de serviços, sete na camada de interface e 13 na camada de controle.

Na *Camada de Serviços* está presente a classe de **Servico**, esta classe é responsável por realizar a comunicação entre o servidor e o cliente web, por meio dela são executadas requisições de consulta e envio de dados de novas entidades ao servidor do Onisciente;

A Camada de Processamento possui as seguintes classes:

- **GestorAcesso:** Classe da camada de controle que interage com a página de login do cliente e trata da validação do usuário junto ao servidor;
- **GestorMonitoramento:** Classe da camada de controle que realiza a gestão da sessão de monitoramento, é responsável por controlar as requisições de informações de contextos das entidades que estiverem inseridas na lista de monitoramento;
- **GestorEntidade:** Classe da camada de controle cuja finalidade é gerenciar todas as operações que envolvam de alguma forma a manipulação de entidades, sejam elas novas ou já existentes;
- **Sensor:** Classe que representa um sensor que atua sobre o sistema;

Figura 50: Diagrama de classes do cliente web



Fonte: Elaborada pelo autor.

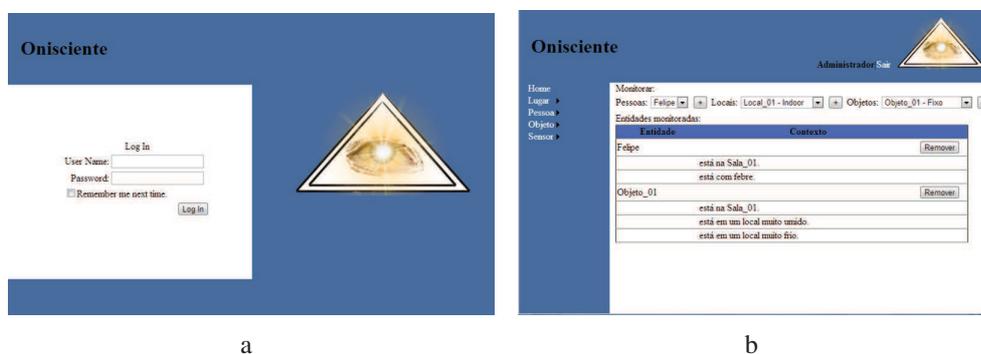
- **DadoSensor:** Classe que contém informação obtida de um sensor;
- **Relação:** Classe que define uma relação entre duas entidades;
- **Entidade:** Classe abstrata que funciona como base de todas as outras que representam algum tipo de entidade no sistema;
- **Objeto:** Classe herdada de entidade que representa uma entidade de tipo Objeto;
- **Pessoa:** Classe herdada que representa uma entidade do tipo pessoa;
- **Lugar:** Classe herdada que representa uma entidade do tipo lugar;
- **Alarme:** Classe que representa uma ocorrência de alarme de uma entidade;
- **Contexto:** Classe que representa um situação contexto de uma entidade;
- **Atributo:** Classe que define uma característica de uma entidade.

Na Camada de Interface, para cada classe existente na camada de interface existe um arquivo com o mesmo nome, porém sem a terminação “.cs”, este arquivo contém as especificações de layout da página, deixando a parte de controle para a classe propriamente dita. As classes pertencentes a essa camada são:

- **Default.aspx.cs:** Classe que representa a página inicial do cliente web que solicita o login do usuário;
- **Site.Master.aspx.cs:** Classe que contém
- **Monitoramento.aspx.cs:** Classe que representa a principal página do cliente web, onde é realizado a visualização do resultado do monitoramento;
- **Sensor.aspx.cs:** Classe que controla a página onde o usuário realiza a edição e o cadastro de entidades do tipo sensor;
- **Lista.aspx.cs:** Classe que é utilizada para exibir a listagem de entidades existentes de um tipo específico;
- **Lugar.aspx.cs:** Classe que permite que o usuário realize o cadastro e edição de entidades do tipo lugar;
- **Objeto.aspx.cs:** Classe que permite que o usuário faça o cadastro e edição de entidades do tipo objeto;
- **Pessoa.aspx.cs:** Classe que permite ao usuário cadastrar e editar entidades do tipo pessoa;

5.3.2.1 Páginas do Site

Figura 51: Telas de login e monitoramento.



Fonte: Elaborada pelo autor.

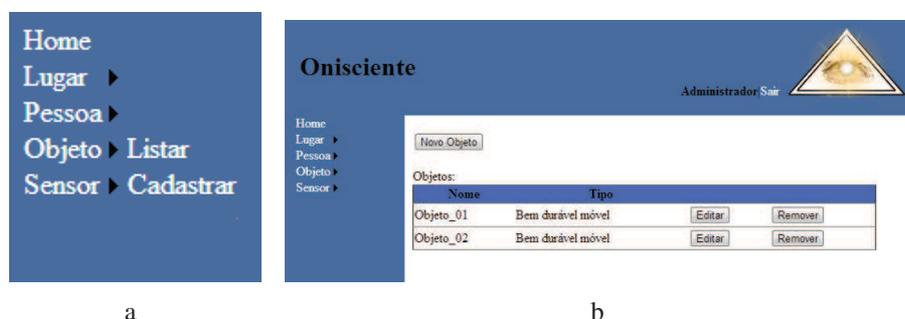
Ao entrar no website do Onisciente (Figura 51.a), ao utilizador será solicitado um nome de usuário e uma senha para conseguir acesso ao sistema. Esta página é composta apenas por dois campos texto para a entrada de dados e um botão para efetuar o login.

Realizado o login o usuário será enviado para a página principal do site (Figura 51.b) que exibe as entidades que estão sendo monitoradas e os recursos disponíveis para adicionar ou remover novas entidades no monitoramento. É possível adicionar facilmente novas entidades para serem monitoradas simplesmente selecionando-as em uma das caixas de seleção disponíveis. Cada uma das caixas apresentadas representa um agrupamento de tipos de entidades monitoradas pelo Onisciente. A caixa de seleção de Pessoas lista todas as pessoas cadastradas no sistema, a caixa de Locais contém todas as entidades que sejam do tipo Indoor ou Outdoor, e a caixa de objetos lista as entidades que sejam do tipo Percível, Fixo ou Móvel. Todas as entidades possuem na sua descrição a especificação do tipo ao qual elas pertencem.

Para adicionar um item no grupo de entidades monitoradas, basta ao usuário, após selecionar a entidade desejada, pressionar o botão “+” localizado ao lado da caixa de seleção, ele irá inserir o item escolhido na lista de entidades monitoradas. Para remover um item da lista, basta o usuário pressionar o botão “Remover” do item na lista. Todo item que possuir alguma informação de contexto possui abaixo de seu nome a listagem desses contextos.

Um menu (Figura 52.a) localizado ao lado esquerdo da página apresenta para cada item listado, a exceção de Home, um *submenu* onde é possível listar as entidades cadastradas para aquele tipo ou realizar um novo cadastro de entidade.

Figura 52: Menu de opções e lista de objetos cadastrados.



Fonte: Elaborada pelo autor.

As listagens de entidades cadastradas possuem como base o mesmo padrão de página (Figura 52.b), ela é composta por uma lista que apresenta o nome da entidade, seguido da descrição do tipo ao qual ela pertence. Para cada item exibido está presente um botão para a edição das informações desta entidade e outro para a remoção desta entidade da base do Onisciente. Se o usuário desejar adicionar uma nova entidade no sistema, ele pode utilizar o botão localizado acima da lista, que irá para a página de cadastro equivalente ao tipo de entidade que está sendo listada.

As páginas de cadastro de entidades são bastante similares, a de cadastro de lugares (Figura 53.a) é composta por um campo de texto para a entrada do nome que irá identificar o local, um grupo de opções que permite definir o tipo de local que a entidade representa e uma lista de atributos relacionados ao local. No final do formulário, logo abaixo da lista de atributos, fica

Figura 53: Páginas de cadastro de entidades e sensores.

The figure consists of four screenshots of the Onisciente web application interface, labeled a, b, c, and d. Each screenshot shows a registration form with a sidebar menu on the left and a main content area on the right. The top of each page features the 'Onisciente' logo and the user's name 'Administrador Sair'.

Screenshot a: Registration form for an entity named 'Sala_03'. The form includes fields for 'Nome' (Sala_03), 'Tipo' (Indoor selected), and 'Adicionar Atributo'. Below is a table of attributes:

Nome	Valor	Tipo	Remove
temperatura	22.0	Double	Remove
umidade	44.0	Double	Remove
		Intero	Salvar

Screenshot b: Registration form for a person named 'Paulo'. It includes fields for 'Nome' (Paulo) and 'Código RFID' (12345678). Below is a table of attributes:

Nome	Valor	Tipo	Remove
temperatura	36.0	Double	Remove

Relationships are defined as: Relacionamento: proprietário de, Entidade: Sala_01 - Indoor, Relacionar.

Screenshot c: Registration form for an object named 'Objeto_04'. It includes fields for 'Nome' (Objeto_04) and 'Código RFID' (99998888). The 'Tipo' options are: Percível, Bem durável fixo, Bem durável móvel. Below is a table of attributes:

Nome	Valor	Tipo	Remove
temperaturaMaxima	36.0	Double	Remove
temperaturaMinima	-10.0	Double	Remove

Relationships are defined as: Relacionamento: propriedade de, Entidade: Sala_01 - Indoor, Relacionar.

Screenshot d: Registration form for a sensor named 'Sensor_01'. It includes fields for 'Nome' (Sensor_01) and 'Tipo' (Labora RFID selected). Below is a table of attributes:

Nome	Valor	Tipo	Remove
temperatura			
umidade			

Relationships are defined as: Relacionamento: propriedade de, Entidade: Felpo, Relacionar.

Fonte: Elaborada pelo autor.

localizado o botão “Salvar”, que persiste a entidade junto ao servidor.

O processo de cadastro de pessoa (Figura 53.b) possui dois campos de texto, um para identificar a entidade e outro para a inserção do código RFID que serve para a identificação desta entidade. Nesta página, além da listagem de atributos também há a possibilidade de especificar as relações entre as entidades já cadastradas, por exemplo, podemos definir que a pessoa é proprietária de um ou mais objetos. O procedimento de inclusão de relacionamentos funciona por meio da seleção da relação desejada em uma caixa de seleção e da entidade alvo desta relação, seguido do pressionamento do botão “Relacionar”. O resultado do evento será a inclusão da relação na lista de relacionamentos localizada logo abaixo dessas caixas de seleção. Se o usuário desejar remover a relação, basta clicar no botão “Remover” localizado no final da linha que descreve a relação na lista.

O procedimento para cadastro de objetos (Figura 53.c) é bastante similar ao realizado na página de cadastro de pessoas, as diferenças entre elas são as opções para a definição do tipo de entidade que está sendo cadastrada, e as opções de relacionamento entre entidades.

A lista de atributos existente nestas três páginas pode ser incrementada ou decrementada conforme o desejo do usuário. Para adicionar novos atributos à entidade, o usuário deve clicar no botão “Adicionar Atributo” localizado acima da lista. No final da lista, aparecerá uma nova linha, conforme pode ser visto na área em destaque da Figura 54, contendo campos de texto

Figura 54: Inclusão de novo atributo

Atributos:

Nome	Valor	Tipo	
temperatura	22.0	Double	<input type="button" value="Remover"/>
umidade	44.0	Double	<input type="button" value="Remover"/>
<input type="text"/>	<input type="text"/>	Inteiro <input type="button" value="v"/>	<input type="button" value="Salvar"/>

Fonte: Elaborada pelo autor.

para a inserção do nome e valor do atributo, e uma caixa de seleção para a definição do tipo do atributo. No final da linha há um botão “Salvar” para que o atributo seja persistido na entidade. Se o usuário desejar remover algum atributo existente, basta ele clicar sobre o botão “Remover” localizado ao final de cada registro.

A última página de cadastro disponível é a de cadastro de sensor (Figura 53.d), ela é bastante similar as telas de cadastro de pessoas e objetos, porém possui menos elementos e pode estar relacionada a apenas uma entidade. A página possui apenas um campo de texto para a entrada do nome do sensor, um grupo de opções para a definição do tipo ao qual ela pertence, uma caixa de seleção para especificar o tipo de dado que ela informará e uma caixa de seleção para cada tipo de entidade do Onisciente, onde deve ser selecionar uma para definir aquela que será utilizada pelo Onisciente como o alvo dos dados obtidos.

A partir do próximo capítulo será abordado o processo avaliativo realizado para determinar se o resultado da implementação realizada atende os requisitos traçados no capítulo 4, onde foi feita a especificação do modelo.

6 AVALIAÇÃO

Este capítulo aborda o processo de avaliação do Onisciente. Inicialmente é descrita a metodologia de avaliação escolhida para avaliar o modelo. Em seguida, são apresentadas, respectivamente, as duas avaliações realizadas. A primeira consiste em uma avaliação da ontologia baseada em métricas e também na execução de cenários. A segunda avaliação utiliza casos de usos para explorar a capacidade do Onisciente de gerar contextos em situações reais de uso.

6.1 Metodologia de avaliação

Com relação a ontologias, não existe um consenso de metodologia para a realização de avaliações (ALMEIDA, 2009). Para realizar a avaliação do Onisciente, foi decidido utilizar a metodologia baseada em métricas descrita em FOEval (BOUIADJRA; BENSLIMANE, 2011). Esta metodologia possui um conjunto de métricas que podem ser escolhidas pelo usuário para aplicação sobre a ontologia. São elas (BOUIADJRA; BENSLIMANE, 2011):

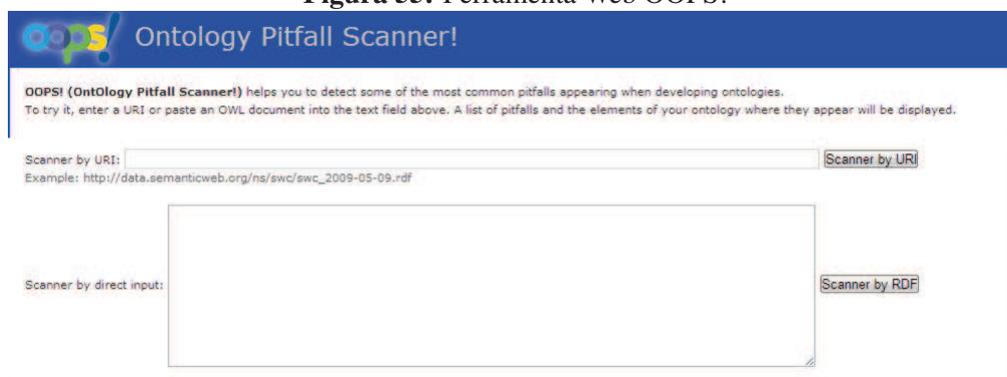
- **Cobertura:** consiste em cobertura de classe e cobertura de relação. A cobertura de classe representa, a partir de palavras chaves pesquisadas, quantos nomes de classes estão presentes na ontologia. A cobertura de relação representa quantos nomes de relação estão presentes na ontologia;
- **Riqueza:** a riqueza de uma ontologia pode ser medida de três maneiras diferentes:
 - (1) Riqueza de relação (RR) é a métrica que reflete a diversidade de relações e posicionamento das relações na ontologia. Uma ontologia que contém muitas relações é mais rica em taxonomia do que as que contém apenas relações hierárquicas;
 - (2) Riqueza de atributos (RA) é o número médio de atributos que são definidos para cada classe que indicam a quantidade de informações pertencentes a uma instância. Quanto mais atributos definidos, mais conhecimento a ontologia contém;
 - (3) Riqueza da ontologia (RO) é a soma da riqueza de relação e riqueza de atributos. A riqueza de relação é definida como a razão entre o número de relações não hierárquicas definidos na ontologia, dividido pelo número total de relações. A riqueza de atributos é definido como o número de atributos para todas as classes, dividido pelo número de classes;
- **Nível de detalhamento:** Esta métrica descreve:
 - (1) Nível de detalhamento global: indica o quão bom o conhecimento é agrupado em diferentes categorias e subcategorias na ontologia. Esta medição pode distinguir entre uma ontologia vertical e uma ontologia horizontal. Formalmente é definido como o número médio de subclasses por classe;

(2) Nível de detalhamento específico: é um indicador dos termos mais importantes em uma ontologia. É preferível que um termo pesquisado seja uma classe com várias subclasses do que uma classe sem nenhuma subclasse. Esta métrica é definida como uma soma de quatro parâmetros: (a) o número médio de subclasses e classes superiores à classe do termo pesquisado; (b) o número de relações para a classe pesquisada; (c) o número de relações para as subclasses da classe pesquisada; (d) o número de relações para as classes superiores à classe pesquisada;

- **Compreensão:** é o número médio de anotações de classe, número médio de anotações de relação e número médio de instâncias por classe;
- **Eficiência computacional:** este princípio dá o panorama do quão facilmente/corretamente uma ontologia pode ser processada, em particular a velocidade em que um *reasoner* necessita para resolver as tarefas requeridas. Formalmente, a eficiência computacional de uma ontologia é a soma do número médio de classes, o número médio de subclasses por classe, o número médio de relações por classe e o tamanho médio da ontologia.

Dadas as métricas disponíveis para avaliação da ontologia através da metodologia descrita em FOEval (BOUIADJRA; BENSLIMANE, 2011), foram escolhidas as métricas riqueza, nível de detalhamento global e eficiência computacional para realizar a avaliação da ontologia do Onisciente. Estas métricas foram selecionadas por serem as que melhor se enquadram às necessidades do Onisciente. Além do uso de métricas para avaliar a ontologia, optou-se por realizar uma avaliação utilizando o Ontology Pitfall Scanner - OOPS!¹³ (POVEDA-VILLALON; SUAREZ-FIGUEROA; GOMEZ-PEREZ, 2012). O OOPS! (Figura 55) é uma ferramenta web que analisa a estrutura da ontologia em busca de armadilhas que possam levar a erros de modelagem. A Figura 56 contém uma listagem parcial destas armadilhas, a lista completa pode ser encontrada no site¹⁴ da ferramenta.

Figura 55: Ferramenta Web OOPS!



Fonte: Elaborada pelo autor.

¹³<http://oeg-lia3.dia.fi.upm.es/oops>

¹⁴<http://www.oeg-upm.net/oops/catalogue.jsp>

Figura 56: Lista de problemas de estrutura de ontologias detectáveis pelo OOPS!

Human understanding	Modelling issues
<ul style="list-style-type: none"> • P1. Creating polysemous elements • P2. Creating synonyms as classes • P7. Merging different concepts in the same class • P8. Missing annotations • P11. Missing domain or range in properties • P12. Missing equivalent properties • P13. Missing inverse relationships • P19. Swapping intersection and union • P20. Misusing ontology annotations • P22. Using different naming criteria in the ontology 	<ul style="list-style-type: none"> • P2. Creating synonyms as classes • P3. Creating the relationship "is" instead of using "rdfs:subClassOf", "rdf:type" or "owl:sameAs" • P4. Creating unconnected ontology elements • P5. Defining wrong inverse relationships • P6. Including cycles in the hierarchy • P7. Merging different concepts in the same class • P10. Missing disjointness • P17. Specializing too much a hierarchy • P11. Missing domain or range in properties • P12. Missing equivalent properties • P13. Missing inverse relationships • P14. Misusing "owl:allValuesFrom" • P15. Misusing "not some" and "some not" • P18. Specifying too much the domain or the range • P19. Swapping intersection and union • P21. Using a miscellaneous class • P23. Using incorrectly ontology elements • P24. Using recursive definition • P25. Defining a relationship inverse to itself • P26. Defining inverse relationships for a symmetric one • P27. Defining wrong equivalent relationships • P28. Defining wrong symmetric relationships • P29. Defining wrong transitive relationships
Logical consistency	
<ul style="list-style-type: none"> • P5. Defining wrong inverse relationships • P6. Including cycles in the hierarchy • P14. Misusing "owl:allValuesFrom" • P15. Misusing "not some" and "some not" • P18. Specifying too much the domain or the range • P19. Swapping intersection and union • P27. Defining wrong equivalent relationships • P28. Defining wrong symmetric relationships • P29. Defining wrong transitive relationships 	
Real world representation	
<ul style="list-style-type: none"> • P9. Missing basic information • P10. Missing disjointness 	

Fonte: Poveda-Villalon, Suarez-Figueroa e Gomez-Perez (2012)

Além destas duas metodologias para a avaliação de ontologias, foi decidido realizar uma avaliação por meio de cenários com o objetivo de avaliar o comportamento da ontologia em situações reais de uso. A utilização de cenários tem sido utilizada pela comunidade acadêmica para validar trabalhos em ambientes sensíveis ao contexto e ubíquos (SATYANARAYANAN, 2011)(DEY, 2001).

Com relação ao modelo do Onisciente, as principais métricas de avaliação utilizadas na computação são: a avaliação funcional – que tem como objetivo provar que o modelo atende a todas as funções propostas; a avaliação de desempenho – que procura determinar a capacidade de resposta, confiabilidade ou escalabilidade de um sistema sob uma determinada carga de trabalho; e a avaliação de usabilidade – que procura determinar o nível de aceitação e usabilidade dos usuários (FERNANDEZ; ABRAHAO; INSFRAN, 2012). Para o Onisciente, são realizadas as avaliações funcionais e de desempenho, uma vez que nesse momento não foi considerado a usabilidade como um fator essencial de avaliação. Apesar de o modelo apresentar meios de interação com o usuário, o foco das avaliações está concentrado no processo de geração de contextos.

Para realizar a avaliação funcional e de desempenho, foram utilizados casos de uso baseados em cenários predefinidos, que possuem como objetivo explorar os limites das entidades, que foram informados durante o seu cadastro. Na sequência deste capítulo serão apresentados dois casos de uso, o primeiro explorando a capacidade de leitura do acelerômetro, juntamente com a capacidade do onisciente de gerar contextos baseado nas informações que estiverem sendo

adquiridas. O segundo caso de uso avaliará o impacto de variações de temperatura e umidade sobre entidades que possuam intervalos sensíveis a estes tipos de valores. Para cada caso de uso implementado, são apresentados os resultados obtidos, juntamente com uma análise destes resultados.

A seguir, são apresentados os resultados da aplicação das metodologias selecionados junto ao Onisciente.

6.2 Avaliação da Ontologia

A avaliação da ontologia envolveu primeiramente a verificação da estrutura elaborada pela ferramenta OOPS!. O resultado (Figura 57) indicou que apenas a falta de *annotations* (P08), recurso que não utilizado, e a citação de um elemento que não possui relação com o restante da ontologia (P04), no caso a classe Contexto. O motivo para esta desconexão é que esta classe é constituída por subclasses que derivam das classes de entidades, não possuindo relação direta com as classes de entidade. Esta desconexão foi apresentada e explicada no capítulo 4 na subseção de Modelagem de Contexto.

Figura 57: Resultado obtido ao utilizar o OOPS!

The screenshot shows the OOPS! (Ontology Pitfall Scanner) interface. At the top, it says "Ontology Pitfall Scanner!". Below that, a description states: "OOPS! (Ontology Pitfall Scanner) helps you to detect some of the most common pitfalls appearing when developing ontologies. To try it, enter a URI or paste an OWL document into the text field above. A list of pitfalls and the elements of your ontology where they appear will be displayed." There are two input fields: "Scanner by URI:" with an example URL and "Scanner by direct input:" containing an XML snippet. Below the input fields, there are two buttons: "Scanner by URI" and "Scanner by RDF". The "Evaluation results" section shows a table with two rows:

[Expand All] [Collapse All]	
Results for P04: Creating unconnected ontology elements.	1 case
Results for P08: Missing annotations.	57 cases

On the right side, there is a "Want to help?" section with links for "Suggest new pitfalls" and "Provide feedback", and a "Documentation:" section with a link for "Pitfall problems".

Fonte: Elaborada pelo autor.

Sabendo que a ontologia está bem constituída, passou-se então para a segunda etapa na avaliação da ontologia, a avaliação por métricas.

6.2.1 Avaliação por métricas

A primeira métrica avaliada foi a riqueza da ontologia. Seguindo as fórmulas de avaliação especificadas em FOEval (BOUIADJRA; BENSLIMANE, 2011), a riqueza de relação (RR) para a ontologia limpa, isto é sem indivíduos, ficou em 1, e populada de forma mínima, com uma entidade e um conjunto completo de sensores, atingiu o índice de 0,69. A riqueza de

atributos (RA) alcançou um índice de 0,44, indiferente da condição da ontologia. Por fim, a riqueza da ontologia (RO) ficou em 1,44 para a ontologia limpa e em 1,13 para a ontologia com um uso mínimo. A avaliação desta métrica indicou que a ontologia apresenta uma maior riqueza de relações entre as suas classes do que de atributos que as definem.

A segunda métrica avaliada foi o nível de detalhamento global, para as 28 classes dispostas no Onisciente um índice de 1,18 foi obtido. Isto significa que a ontologia proposta para o Onisciente se enquadra na taxonomia vertical, onde, para a maioria das classe existentes, subclasses estão presentes elevando a quantidade de categorias em que o conhecimento é classificado.

A terceira e última métrica escolhida para avaliação, foi a eficiência computacional. Esta métrica atingiu um valor de 1,42 ao realizarmos a comparação da ontologia do Onisciente com uma variação dela mesma 100 vezes superior. Este resultado indica que mesmo tendo um considerável aumento na quantidade de classes, instâncias, relações e outros elementos, a ontologia sofre pouca perda de eficiência mantendo seu desempenho satisfatório.

Como a ontologia foi elaborada sem a utilização de outras ontologias, e também não foram encontradas ontologias que possuíssem as mesmas características da do Onisciente, os resultados obtidos serão deixados como referência para implementações futuras que desejem avaliar as suas características as confrontando com as do Onisciente.

6.2.2 Avaliação por cenários

Os cenários elaborados procuram representar algumas das atividades durante a execução do Onisciente, verificando o comportamento da ontologia. Estes cenários procuram demonstrar a funcionalidade da ontologia durante a aplicação de dados sobre a mesma. Para isso, foram elaborados 3 cenários hipotéticos que procuram atender este requisito.

O primeiro cenário procura demonstrar o funcionamento da ontologia ao receber informações obtidas pelo acelerômetro, que são aplicadas diretamente sobre uma entidade específica. A descrição do cenário é realizada a seguir:

“Uma caixa contendo um material considerado frágil, de baixa resistência, que pesa em torno de 1,5Kg é transportada por um caminhão em uma via asfaltada quando este passa por um buraco que sacode a carga arremessando-a para cima a uma força de 2G. Ao cair, a caixa rola ficando fora de posição até o término da viagem. Segundo o acelerômetro, os valores dos eixos x, y e z registravam respectivamente 0.1,-0.6 e 0.7.”

A aplicação deste cenário envolve analisar os seguintes pontos:

1. Avaliação do impacto ocorrido quando há a queda da carga;
2. A capacidade de detecção da posição da carga.

Figura 58: Consulta SPARQL que procurará por objetos danificados.

```
"CONSTRUCT { ?objetos a ont:ObjetosDanificados } WHERE "
+ "{ ?objetos ont:hasResistencia ?r. "
+ "?objetos ont:sufferedImpact ?i. "
+ "?objetos ont:hasWeight ?m. "
+ "?objetos ont:isBeenMonitored ?z. "
+ "FILTER ( (?m * (?i*9.81)) > ?r )]"
```

Fonte: Elaborada pelo autor.

A execução deste cenário usa os dados obtidos pelo equipamento utilizado, os enviando para o servidor do Onisciente que executa consultas SPARQL na ontologia (Figuras 58 e 59).

A consulta da Figura 58 obtém os objetos que serão classificadas como objetos danificados. Todo objeto que sofrer algum impacto (i) que gerar uma força maior que a sua resistência (r) será classificado como danificado. A força é obtida ao converter o impacto (i), que está representado em força da gravidade, e multiplicando o resultado, em metros por segundo, pelo peso do objeto. Esta consulta deverá fornecer os elementos necessários para avaliar o primeiro item enumerado como objetivo de análise.

Figura 59: Consulta SPARQL que verificará objetos que estejam virados.

```
"CONSTRUCT { ?objetos a ont:ObjetosVirados } WHERE"
+ "{ ?objetos ont:hasGForceX ?gx. "
+ "?objetos ont:hasGForceY ?gy. "
+ "?objetos ont:hasGForceZ ?gz. "
+ "?objetos ont:isBeenMonitored ?z. "
+ "FILTER (?gx > 0.9 || ?gx < -0.9 || ?gy > 0.9 || ?gy < -0.9 || ?gz < -0.9)}"
```

Fonte: Elaborada pelo autor.

A Figura 59 apresenta uma consulta para verificar o posicionamento de objetos com base nos valores de um acelerômetro. Nesta consulta SPARQL são avaliados os valores registrados para os eixos X, Y e Z do objeto. Se houver alguma forte variação nos eixos do objeto ele é considerado como virado. Para esta consulta funcionar corretamente, considera-se que o objeto em avaliação está armazenado na posição considerada correta. O retorno desta consulta serve para realizar a avaliação do segundo objetivo de análise deste cenário.

Como resultado, a aplicação dos dados descritos no cenário mostrou que o impacto gerado no momento que a carga foi arremessada acarretou em avaria do seu conteúdo. As informações posteriores ao impacto informaram que a caixa passou o restante da viagem fortemente inclinada, o que, conforme as definições da ontologia, caracteriza que a entidade está virada.

O segundo cenário proposto tem como foco avaliar o comportamento da ontologia ao receber informações de temperatura e umidade e qual o efeito que ambas as informações combina-

das podem exercer sobre uma pessoa e um objeto.

“Em uma construção estão dispostas três salas que possuem temperatura e umidade controlada, dentro de cada uma destas salas está uma pessoa e um objeto que possui a necessidade de estar em um ambiente com certas condições específicas. A primeira sala é utilizada para resfriar alimentos e possui uma temperatura de aproximadamente -5°C com umidade de 65%, ela contém um objeto que suporta temperaturas entre -15 e 2°C e umidades variando entre 30 e 80%. A segunda sala está a uma temperatura de 25°C e umidade em 30% e o objeto dentro dela pode ficar a uma temperatura entre 0 e 60°C e uma umidade que varie entre 0 e 50%. A terceira sala representa um local que contém um forno que quando aceso mantém a sala a 40°C e umidade em 15%, o objeto dentro dela suporta temperaturas entre 0 e 100°C e umidades entre 10 e 70%. Em um determinado momento uma queda de luz desliga os equipamentos que mantinham a refrigeração do ambiente, causando mudanças na temperatura e umidade a medida que o tempo passa. Após uma hora a primeira sala alcança uma temperatura de -1°C e umidade de 58%, a segunda 28°C de temperatura, mantendo os 30% de umidade e a terceira sala 51°C de temperatura com uma umidade de 13%. Ao final da segunda hora, quando a luz retornou, a primeira sala indicava valores de 4°C e 52% de temperatura e umidade respectivamente, a segunda sala estava com temperatura de 31°C e umidade 31% enquanto a terceira sala alcançou 59°C e 9% de umidade. ”

O cenário descrito tem como finalidade avaliar:

1. O grau de conforto das pessoas a cada medição de temperatura e umidade realizada;
2. As condições dos objetos inseridos a cada medição;
3. O contexto de cada entidade envolvida no momento final do cenário.

Da mesma forma que o primeiro cenário, a resolução deste cenário é realizada utilizando consultas SPARQL com os dados adquiridos pelo equipamento. As Figuras 60, 61, 62, 63, 64, 65 e 66 exibem as consultas utilizadas para a avaliação deste cenário.

As três consultas apresentadas nas Figuras 60, 61 e 62 buscam classificar a sensação de desconforto das pessoas. A primeira irá informar as pessoas que estão com sensação de desconforto, são as pessoas que não estão nem com frio, nem com calor, mas também não estão se sentindo confortáveis. A segunda e terceira irão informar pessoas que estão com sensação de calor e frio respectivamente. Estas três consultas possuem como finalidade avaliar o primeiro item dos objetivos apresentados.

A resolução do segundo item objetivado é realizada pelo conjunto das quatro consultas SPARQL apresentadas pelas Figuras 63, 64, 65 e 66. As duas primeiras irão procurar por

Figura 60: Consulta SPARQL que procurará por pessoas que não estão confortáveis.

```

CONSTRUCT { ?pessoas a ont:PessoasSensacaoDesconforto } WHERE {
?pessoas ont:isInPlace ?l.
?l ont:hasTemperatura ?t.
?l ont:hasUmidade ?u
FILTER (
((?t*1.8+32) - 0.55*(1 - ?u/100)*((?t*1.8+32)-14) >= 75 &&
(?t*1.8+32) - 0.55*(1-?u/100)*((?t*1.8+32)-14) < 80 ) ||
((?t*1.8+32) - 0.55*(1 - ?u/100)*((?t*1.8+32)-14) >= 55 &&
(?t*1.8+32) - 0.55*(1-?u/100)*((?t*1.8+32)-14) < 60 ) ) }

```

Fonte: Elaborada pelo autor.

Figura 61: Consulta SPARQL que procurará por pessoas com sensação de calor.

```

CONSTRUCT { ?pessoas a ont:PessoasSensacaoCalor } WHERE {
?pessoas ont:isInPlace ?l.
?l ont:hasTemperatura ?t.
?l ont:hasUmidade ?u
FILTER ( (?t*1.8+32) - 0.55*(1-?u/100)*((?t*1.8+32)-14) > 80 ) }

```

Fonte: Elaborada pelo autor.

Figura 62: Consulta SPARQL que procurará por pessoas com sensação de frio.

```

CONSTRUCT { ?pessoas a ont:PessoasSensacaoFrio } WHERE {
?pessoas ont:isInPlace ?l.
?l ont:hasTemperatura ?t.
?l ont:hasUmidade ?u
FILTER ( (?t*1.8+32) - 0.55*(1-?u/100)*((?t*1.8+32)-14) < 55 ) }

```

Fonte: Elaborada pelo autor.

Figura 63: Consulta SPARQL que procurará por objetos em temperatura alta.

```

CONSTRUCT { ont:ObjetosEmUmidadeAlta a ?object } WHERE {
?object ont:isInsertedIn ?lugares.
?object ont:hasValorUmidadeMaxima ?x.
?lugares ont:hasUmidade ?y.
FILTER(?x < ?y)}

```

Fonte: Elaborada pelo autor.

objetos que estejam expostos a umidades altas e baixas e as duas seguintes por objetos em temperaturas baixas ou altas respectivamente.

Figura 64: Consulta SPARQL que procurará por objetos em umidade baixa.

```
CONSTRUCT { ont:ObjetosEmUmidadeBaixa a ?object } WHERE {
  ?object ont:isInsertedIn ?lugares.
  ?object ont:hasValorUmidadeMinima ?x.
  ?lugares ont:hasUmidade ?y.
  FILTER( ?x > ?y)}
```

Fonte: Elaborada pelo autor.

Figura 65: Consulta SPARQL que procurará por objetos em temperatura baixa.

```
CONSTRUCT { ont:ObjetosEmTemperaturaBaixa a ?object } WHERE {
  ?object ont:isInsertedIn ?lugares.
  ?object ont:hasValorTemperaturaMinima ?x.
  ?lugares ont:hasTemperatura ?y.
  FILTER( ?x > ?y)}
```

Fonte: Elaborada pelo autor.

Figura 66: Consulta SPARQL que procurará por objetos em temperatura alta.

```
CONSTRUCT { ont:ObjetosEmTemperaturaAlta a ?object } WHERE {
  ?object ont:isInsertedIn ?lugares.
  ?object ont:hasValorTemperaturaMaxima ?x.
  ?lugares ont:hasTemperatura ?y.
  FILTER( ?x < ?y)}
```

Fonte: Elaborada pelo autor.

O terceiro item dos objetivos propostos envolve a análise dos resultados obtidos durante a execução do cenário. Nas situações em que foram registradas as mudanças de condições dos ambientes, em alguns casos foram gerados contextos para cada entidade presente (pessoas e objetos) no cenário. Para as pessoas, foram gerados contextos representando a sensação de desconforto de cada indivíduo para cada situação descrita, o registro destes contextos podem ser visualizados na Tabela 6. Quanto aos objetos, a Tabela 7 informa a quantidade de contextos gerados para cada tipo em avaliação, nela é possível perceber que apenas na terceira situação informada é que contextos foram registrados, pois foi quando as condições das salas onde estavam os objetos extrapolaram os limites indicados pelos seus atributos.

O resultado obtido indicou que apenas na segunda hora, no momento em que a luz retornou, os objetos apresentaram sinais de possíveis avarias. O objeto da primeira sala acabou sendo exposto a uma temperatura acima do ideal enquanto o objeto da sala três acabou ficando a uma umidade abaixo do ideal, os demais objetos não tiveram nenhuma indicação que pudessem ter sofrido algum tipo de avaria. Vale destacar que mesmo que o Onisciente informe que os objetos

Tabela 6: Contextos de sensação de desconforto das pessoas em cada situação do cenário 2.

	Situação 1	Situação 2	Situação 3
Pessoa 1	frio	frio	frio
Pessoa 2	frio	desconforto	desconforto
Pessoa 3	ok	ok	desconforto

Tabela 7: Contextos de objetos registrados em cada situação do cenário 2.

	Situação 1	Situação 2	Situação 3
Objeto em temperatura alta	0	0	1
Objeto em temperatura baixa	0	0	0
Objeto em umidade alta	0	0	0
Objeto em umidade baixa	0	0	1

tenham sido expostos a condições desfavoráveis, não há meios de garantir que eles sofreram algum tipo de dano, o que o sistema informa é uma possibilidade segundo as características da entidade analisada. Quanto às pessoas, o indivíduo que esteve na primeira sala esteve com frio durante todo o tempo que se passou, a pessoa que esteve na segunda sala inicialmente estava com frio, mas devido o aumento de temperatura passou o restante do tempo com sensação de desconforto. Por fim, a pessoa que estava na sala mais quente passou a maior parte do tempo confortável tendo sensação de desconforto no final do período. A Tabela 8 informa os contextos existentes ao final da execução do cenário.

Tabela 8: Contextos gerados ao final do cenário 2.

Pessoa 1 está com sensação térmica desconfortável.
Objeto 1 está em local muito quente.
Objeto 3 está em local com umidade abaixo do ideal.

O terceiro cenário explora a combinação dos dados obtidos com o acelerômetro, temperatura e umidade sobre uma pessoa. Considere a seguinte situação:

“João está andando de moto a uma velocidade de 80 km/h em uma estrada asfaltada em ótimas condições durante uma chuva de inverno. No momento, a temperatura está em 13°C a uma umidade de 88%.”

Este cenário tem como objetivo avaliar:

1. A sensação térmica da pessoa;
2. O nível de conforto da pessoa.

O primeiro item é obtido através da consulta SPARQL explicitada pela Figura 67, ela calcula e salva na ontologia a sensação térmica que a pessoa deve estar sentindo.

Figura 67: Consulta SPARQL que calcula a sensação térmica de uma pessoa.

```

CONSTRUCT { ?pessoas ont:hasTermicSensation ?st } WHERE {
  ?pessoas ont:isInPlace ?l.
  ?pessoas ont:hasSpeed ?v.
  ?pessoas ont:hasSpeedSqrt ?vrq.
  ?l ont:hasTemperatura ?t.
  ?l ont:hasUmidade ?u.
  BIND((33 + ((10*?vrq)+10.45-?v)*((?t - 33)/22)) AS ?st) }

```

Fonte: Elaborada pelo autor.

Para determinar o nível de conforto são utilizadas consultas SPARQL que calculam o índice de desconforto das pessoas. Ao contrario das consultas utilizadas no segundo cenário, estas consultas utilizam a sensação termica da pessoa e não a temperatura do ambiente em que elas estão localizadas. As Figuras 68, 69, 70 demonstram estas pesquisas com o parâmetro de sensação termica em uso.

Figura 68: Consulta SPARQL que procurará por pessoas que não estão confortáveis baseado na sua sensação térmica.

```

CONSTRUCT { ?pessoas a ont:PessoasSensacaoDesconforto } WHERE {
  ?pessoas ont:isInPlace ?l.
  ?l ont:hasTermicSensation ?t.
  ?l ont:hasUmidade ?u
  FILTER (
    ((?t*1.8+32) - 0.55*(1 - ?u/100)*((?t*1.8+32)-14) >= 75 &&
    (?t*1.8+32) - 0.55*(1-?u/100)*((?t*1.8+32)-14) < 80 ) ||
    ((?t*1.8+32) - 0.55*(1 - ?u/100)*((?t*1.8+32)-14) >= 55 &&
    (?t*1.8+32) - 0.55*(1-?u/100)*((?t*1.8+32)-14) < 60 ) )

```

Fonte: Elaborada pelo autor.

Ao aplicar os dados na ontologia, foi descoberta que a sensação térmica a qual a pessoa estava exposta era de aproximadamente 1°C e utilizando esta informação para analisar o índice de conforto desta pessoa, foi possível que durante o tempo em que ela esteve na estrada, ela estava passando frio.

A ontologia possui grande importância para o funcionamento do Onisciente, os testes realizados até este momento tiveram como objetivo avaliar a sua capacidade de analisar informações apresentadas e gerar contextos que estivessem em conformidade às situações especificadas pelos cenários de teste. Após finalizada esta etapa é dada sequência nas avaliações dando ênfase ao modelo como um todo.

Figura 69: Consulta SPARQL que procurará por pessoas com sensação de calor baseado na sua sensação térmica.

```
CONSTRUCT { ?pessoas a ont:PessoasSensacaoCalor } WHERE {
  ?pessoas ont:isInPlace ?l.
  ?l ont:hasTermicSensation ?t.
  ?l ont:hasUmidade ?u
  FILTER ( (?t*1.8+32) - 0.55*(1-?u/100)*((?t*1.8+32)-14) > 80 ) }
```

Fonte: Elaborada pelo autor.

Figura 70: Consulta SPARQL que procurará por pessoas com sensação de frio baseado na sua sensação térmica.

```
CONSTRUCT { ?pessoas a ont:PessoasSensacaoFrio } WHERE {
  ?pessoas ont:isInPlace ?l.
  ?l ont:hasTermicSensation ?t.
  ?l ont:hasUmidade ?u
  FILTER ( (?t*1.8+32) - 0.55*(1-?u/100)*((?t*1.8+32)-14) < 55 ) }
```

Fonte: Elaborada pelo autor.

6.3 Avaliação do Onisciente

A avaliação do Onisciente tem como objetivo, primeiramente, testar a funcionalidade do protótipo desenvolvido em situações reais de uso. Também tem como finalidade obter métricas para analisar o desempenho do servidor ao gerar contextos conforme dados são obtidos em pequenos intervalos de tempo. O processo avaliativo envolve a aplicação de dois casos de uso que representam situações reais que podem ocorrer com entidades. O primeiro caso de uso representa uma carga sendo transportada por uma via que varia entre boas e más condições. O segundo caso de uso explora a exposição de entidades às condições climáticas durante um período controlado de tempo.

6.3.1 Primeiro caso de uso

O primeiro caso de uso foi utilizado para explorar a capacidade do Onisciente de determinar se objetos estão sofrendo vibrações e impactos. O teste consistiu em realizar uma rota predefinida, constituída por trechos bem distintos. O primeiro trecho (Figura 71 cor vermelha) é composto por uma estrada de terra batida (Figura 72) de aproximadamente 1,2km de extensão, bastante irregular, que possui muita oscilação e buracos. O segundo trecho (Figura 71 cor azul) consiste em uma via asfaltada (Figura 73) com extensão aproximada de 1,5km, bem construída, plana, com algumas lombadas.

Ao determinar este caso de uso, alguns pontos que necessitam uma atenção mais aprofun-

Figura 71: Rotas do primeiro caso de uso.



Fonte: Elaborada pelo autor.

dada foram enumerados:

1. Analisar os picos de força exercidos sobre o objeto monitorado e o impacto gerado sobre a entidade;
2. O tempo gasto pelo Onisciente para determinar situações de contexto da entidade após a ocorrência de um evento (se alguma ocorrência de eventos for registrada durante o experimento);
3. A precisão do contexto gerado pelo Onisciente.

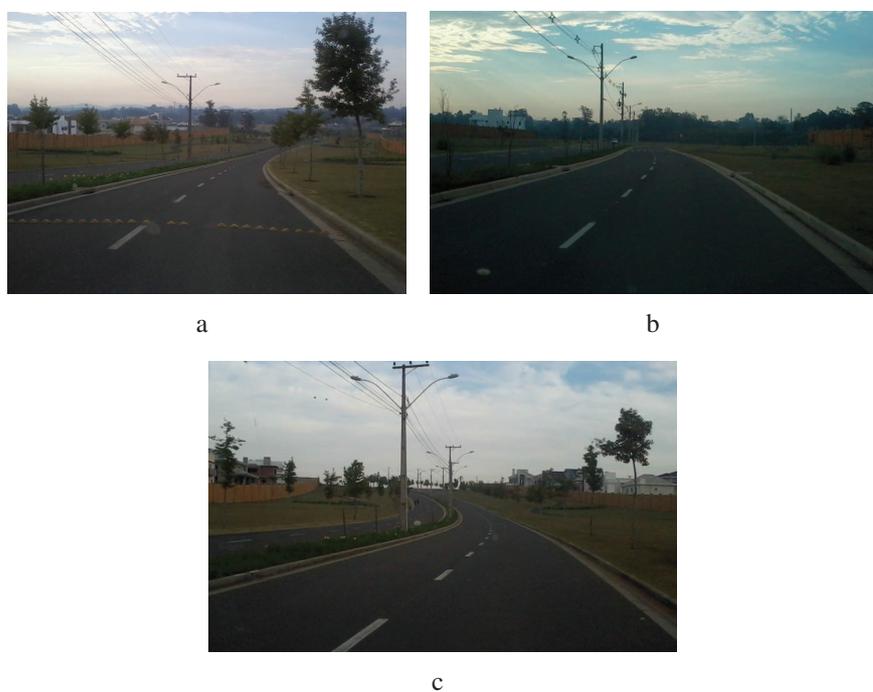
O primeiro item foi destacado porque ele tem como finalidade comprovar a capacidade do modelo proposto em adquirir informações da forma mais condizente com os acontecimentos

Figura 72: Imagens do trecho 1 do primeiro caso de uso.



Fonte: Elaborada pelo autor.

Figura 73: Imagens do trecho 2 do primeiro caso de uso.



Fonte: Elaborada pelo autor.

pelos quais está sendo exposto. O segundo ponto busca analisar o desempenho do Onisciente ao receber novas informações. Por fim, como última análise busca-se avaliar o resultado do

processamento do Onisciente.

6.3.2 Resultados do primeiro caso de uso

Cada trecho foi percorrido por três vezes a uma velocidade média de 40 km/h. O equipamento de monitoramento foi posto sobre o tampão do porta-malas do veículo utilizado. Para garantir a integridade do equipamento, este foi posto dentro de uma caixa para protegê-lo de possíveis impactos durante o trajeto.

Conforme a realização dos trajetos foi sendo realizada, alguns registros de contextos foram gerados. O primeiro trecho, composto por uma estrada de terra, registrou, nas três passagens, dois registros de impactos, com valores aproximados, a exceção da terceira passagem que apresentou um valor bem acima dos demais (desvio padrão de 0,32). Estes valores podem ser analisados ao verificar a Tabela 9, onde são informados para cada passagem do primeiro trecho definido a quantidade de impactos ocorridos e qual o valor do maior impacto registrado. A medição dos dois primeiros impactos foi adquirida no mesmo ponto dos trajetos realizados logo ao início da rota. A terceira medição foi realizada no final do trajeto em um local situado nos últimos metros da rota.

Tabela 9: Registro de contextos do trecho 1

Passagem	Impactos	Maior Impacto
1 ^a	2	0,64
2 ^a	2	0,58
3 ^a	2	1,16

No segundo trecho definido, composto por uma via asfaltada com algumas lombadas, os resultados obtidos divergiram a cada passada. Na primeira execução do trecho não foram registradas ocorrências, porém, na segunda passada houve um registro de que a caixa virou e na terceira a ocorrência de um impacto. Ambos os registros foram feitos no mesmo ponto do trajeto, em uma pequena lombada para redução de velocidade. A Tabela 10 apresenta os resultados adquiridos ao percorrer o segundo trecho do experimento. Nela estão presentes, para cada execução realizada, a quantidade e tipo de contextos gerados e, se necessário, uma descrição informando um valor relacionado ao contexto. A variação dos resultados obtidos está relacionada a velocidade com que o veículo trafegou pelo trecho no decorrer da execução do trajeto planejado. Durante o experimento, se tentou manter uma velocidade constante para que a coleta de dados pudesse ser mais homogênea entre as execuções, porém, devido pequenas variações nesta velocidade, no ponto onde era necessário realizar uma frenagem a intensidade do uso dos freios influenciou na variação da coleta dos dados resultando nos dados que geraram os contextos adquiridos.

Durante a realização do caso de uso, notou-se que o desempenho do Onisciente esteve abaixo do esperado para realizar a análise das informações recebidas e processar os contex-

Tabela 10: Registro de contextos do trecho 2

Passagem	Contexto Gerado	Quantidade	Informação
1 ^a	Nenhum	0	-
2 ^a	Carga virou	1	-
3 ^a	Impacto	1	Valor do impacto de 1,09.

tos. O equipamento utilizado para adquirir as informações foi programado para enviar os dados de maneira seletiva, mas devido a irregularidade do terreno percorrido, haviam muitas mudanças nos eixos do acelerômetro, gerando uma grande quantidade de dados que eram enviados ao servidor. O servidor por sua vez possui um gargalo no processo de geração de contextos devido a necessidade da biblioteca que gere a ontologia ter de recarregá-la antes que se possa realizar as consultas SPARQL. A combinação destas duas situações acarretou em um atraso no processamento que variou de 5 a 10 segundos conforme o trecho percorrido. Para otimizar este processo seria necessário ou implementar a gestão da ontologia no servidor utilizando uma biblioteca que realizasse um procedimento mais eficiente que a atual, ou explorar uma tecnologia diferente. Esta resolução não será abordada neste trabalho, destacando-a como um trabalho futuro.

Independente da margem de atraso no processamento dos contextos, o Onisciente se revelou bastante satisfatório ao que se trata da precisão do contexto gerado, não criando nenhum contexto falso positivo conforme os dados adquiridos.

6.3.3 Segundo caso de uso

O segundo caso de uso consistiu em um monitoramento das condições climáticas de um ambiente aberto (*Outdoor*) durante um período de tempo em um dia chuvoso de inverno na cidade de Gravataí, localizada no estado do Rio Grande do Sul, Brasil. Neste ambiente era possível encontrar um objeto que possuía algumas restrições de umidade e temperatura que determinavam os limites aos quais poderia ser exposto. Este objeto também foi monitorado para se ter conhecimento se as condições as quais ele estava sendo exposto não estavam lhe afetando negativamente.

Para este caso de uso, os tópicos que foram escolhidos para serem melhores analisados foram:

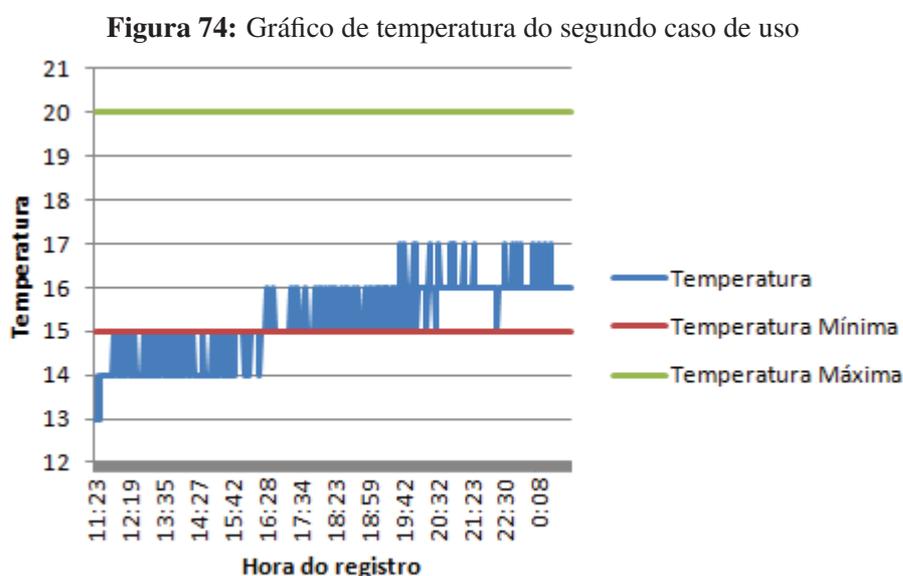
1. A quantidade de vezes que a entidade monitorada ultrapassou os limites especificados;
2. Os tipos de contextos gerados durante a execução do caso de uso;

O primeiro item listado equivale a quantidade de contextos gerados durante o experimento, já o segundo tópico procura quantificar os tipos gerados. O objetivo deste caso de uso é explorar o comportamento do Onisciente com a diversidade de situações que devem ser obtidas durante o

processo devido as configurações do objeto monitorado e as variações de temperatura e umidade do ambiente.

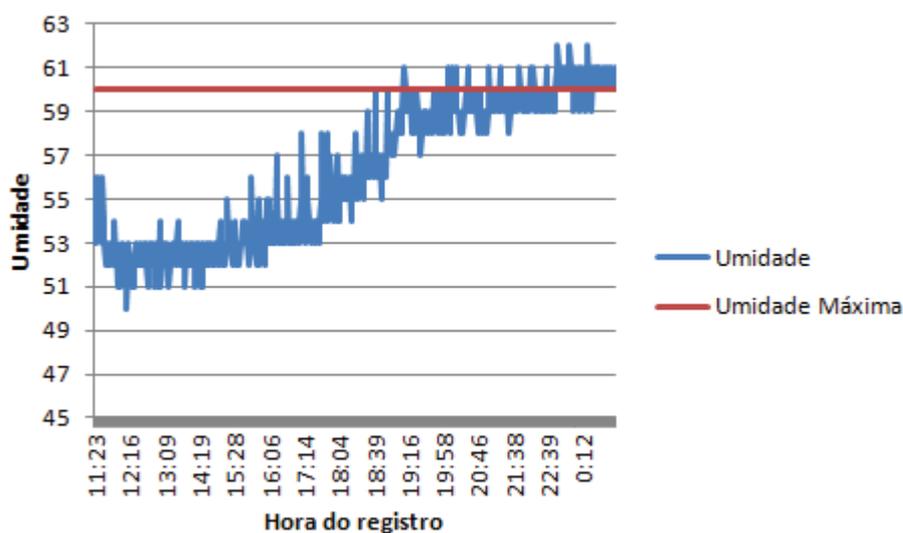
6.3.4 Resultados do segundo caso de uso

A aplicação do caso de uso foi realizada durante um período de 14 horas iniciando às 11:23 horas da manhã até às 1:08 horas da manhã do dia seguinte. Durante este intervalo de tempo foi monitorando um objeto inerte que possuía limites de umidade mínima e máxima de 30 e 60% respectivamente e deveria estar em uma temperatura que variasse entre 15 e 20°C. Durante a coleta de dados a temperatura variou entre 13°C no início dos testes, chegando a valores de 17°C no final do período, estes dados podem ser visualizados pelo gráfico ilustrado pela Figura 74. A umidade por sua vez obteve variações entre 50 e 62% e está demonstrada pelo gráfico da Figura 75. Ambos os gráficos exibem os valores dos dados coletados, e indicam o limite inferior e superior de temperatura, para o gráfico de temperatura, e qual o limite superior de umidade do objeto para o gráfico de umidade. Através deles é possível verificar que o objeto esteve exposto a temperaturas abaixo do ideal no início do experimento e que a partir de um determinado momento a umidade passou a ser bastante elevada, ficando acima do ideal para o objeto.



Fonte: Elaborada pelo autor.

Durante a ocorrência do experimento, foram gerados diversos casos de contexto, totalizando 74 ocorrências de umidade acima do ideal e 225 casos de temperatura abaixo do limite mínimo. Curiosamente, houve quatro ocorrências inválidas de objeto virado durante o período de testes, estas informações foram resultados de uma falha do acelerômetro ao efetuar a leitura do sensor já que o objeto monitorado estava completamente inerte e corretamente posicionado sobre uma mesa.

Figura 75: Gráfico de umidade do segundo caso de uso

Fonte: Elaborada pelo autor.

6.4 Discussão geral sobre as avaliações

De maneira geral, o resultado da avaliação do Onisciente e de sua ontologia atenderam às expectativas criadas quanto a sua funcionalidade, mas em se tratando do seu desempenho, ficou um pouco abaixo do esperado devido ao gargalo de processamento do servidor. Fora este problema, o servidor gerou contextos de forma correta, tendo os resultados com conformidade às características de cada entidade analisada.

O equipamento utilizado apresentou estabilidade durante a sua utilização, à exceção quando aplicado no segundo caso de uso que, quando em estado de completa inércia, apresentou valores captados pelo acelerômetro. Estes falsos positivos do acelerômetro foram suficientes para gerar contextos inválidos nos testes realizados, mas não foram percebidos durante a execução do primeiro caso de uso.

A ontologia por sua vez, pareceu estar bem constituída e as consultas SPARQLS postas à prova demonstraram atender às expectativas quanto a sua funcionalidade e desempenho, gerando resultados consistentes rápida e eficientemente, mesmo quando utilizadas de maneira encadeada.

7 CONSIDERAÇÕES FINAIS

Neste capítulo são apresentadas as considerações finais sobre o trabalho realizado. O capítulo aborda as conclusões obtidas durante a realização do trabalho, as principais contribuições ao comparar o Onisciente com os trabalhos que serviram como inspiração para a sua elaboração, além da apresentação de tópicos para eventuais trabalhos futuros que possam agregar algum valor por meio de adição ou aperfeiçoamento de funcionalidades.

7.1 Conclusões

Este trabalho apresentou o Onisciente, um modelo que possui como principal objetivo gerar contextos a partir de dados variados de diversas fontes de dados, que combinados podem prover informação sobre uma ou mais entidades que possam ou estejam sendo monitoradas. No decorrer da evolução deste trabalho, foram apresentadas as especificações do modelo, assim como a descrição de sua implementação e uma abordagem avaliativa que explorou diversos pontos chave de sua funcionalidade. No final, pôde-se concluir que o Onisciente atende aos objetivos que lhe foram traçados, ao trazer para o usuário final informações que lhe descrevem a situação do que ele deseja monitorar.

As avaliações revelaram que o Onisciente permite o reconhecimento de situações críticas de entidades monitoradas, quase que em tempo real e de maneira automatizada, não sendo necessária intervenção humana em nenhum momento do processo de geração de contextos. O processo de geração de contextos foi bastante satisfatório, tendo criados todos aqueles que eram esperados, tanto para os cenários especificados tanto para os casos de uso utilizados, e condizentes com a realidade. Levando em consideração apenas o segundo caso de uso onde houve a maior quantidade de contextos gerados e sendo este o único teste realizado onde não havia uma quantidade predeterminada de contextos esperados, 98, 68% dos resultados obtidos estiveram em conformidade com as definições para os contextos de temperatura e umidade impostas pelos atributos da entidade monitorada, totalizando 299 contextos corretos de um total de 303. A quantidade restante retrata contextos baseados em dados que geraram falsos positivos. Estes dados foram obtidos incorretamente pelo equipamento utilizado, o Onisciente apenas processou a informação.

Como ponto fraco, o desempenho do processo de geração de contextos ficou um pouco abaixo do esperado levando de 5 a 10 segundos entre a captura dos dados e a geração do contexto, isto foi ocasionado pelo uso de uma biblioteca de manipulação de ontologias que, quando exigida, não apresentou os recursos suficientes para garantir a eficiência do processo sendo necessário realizar algumas operações de E/S no disco rígido do servidor. Para solucionar este gargalo de desempenho uma alternativa seria utilizar uma biblioteca de manipulação de ontologias que oferecesse melhores recursos que a utilizada no Onisciente, mas até o início da implementação deste trabalho não havia uma biblioteca disponível com tais características. Ou-

tra possível solução seria retirar do atual modelo a gestão da ontologia, fornecendo este recurso por meio de uma API em uma linguagem que dispusesse de melhores recursos, que apresentem uma melhor eficiência do que os encontrados para C#. A implementação desta solução não poderia ser realizada em tempo hábil para a conclusão deste trabalho, além de que explorar esta alternativa também envolveria ter de analisar o impacto da comunicação entre as partes envolvidas, além de desenvolver tratamentos para eventuais falhas de comunicação. Esta possível solução é deixada como um trabalho futuro.

Outro ponto a destacar foi que, devido a restrições de equipamento, não foi possível explorar a total capacidade do modelo, como a possibilidade de utilizar os recursos de localização disponíveis pelas leitoras RFID com um raio de alcance para leitura de *tags*, que poderia incrementar o nível de detalhamento de contextos que tivessem como base o posicionamento de entidades. A exploração desta funcionalidade é deixada como um trabalho futuro.

7.2 Principais contribuições

A principal contribuição do Onisciente é a de ser um modelo capaz de gerar contextos através do tratamento e combinação de dados, a partir de inúmeras e variadas informações, adquiridas por diferentes sensores, o que o torna diferenciado pois os trabalhos encontrados na área possuem focos em aplicações específicas e tendem a abordar partes do processo realizado pelo Onisciente, não sendo capazes de fornecer os mesmos resultados. Ele foi constituído de forma que possa ser utilizado em variados tipos de estudos, projetos e diversas áreas.

Em comparação com os trabalhos relacionados, o Onisciente tem a vantagem de ter sido projetado de uma maneira genérica, que permite a expansão de suas funcionalidades, através do incremento de sua ontologia e a inclusão de novos tipos de sensores. O modelo proposto possui a capacidade de realizar as mesmas medições apresentadas pelos trabalhos relacionados, sendo credenciado a atuar como um substituto dos recursos neles utilizados. Demais diferenças entre o Onisciente e os trabalhos relacionados podem ser verificadas na Tabela 11, onde, se destacam a capacidade de trabalhar com diversos tipos de RFID e o oferecimento de *web services* REST, que permite uma fácil integração com qualquer tecnologia, aplicação ou ferramenta que venha a consumir o resultado do processo de geração de contextos oferecido pelo Onisciente. Outro destaque é a quantidade de recursos para a visualização de resultados oferecido pelo Onisciente. Enquanto os demais trabalhos, em sua maioria, não especificam os meios pelos quais os usuários podem visualizar os resultados gerados, o Onisciente provê este acesso através de dispositivos móveis e via acesso *web*. Com relação aos trabalhos relacionados, o Onisciente apresenta algumas limitações que podem ser melhoradas futuramente. A limitação imposta pelo alcance da leitora RFID utilizada impediu que medições como as realizadas em Delen, Sharda e Hardgrave (2011) fossem efetuadas, onde, analisando a distância entre a entidade e o equipamento de monitoramento, é possível estimar a variação do efeito do ambiente sobre esta entidade. Outra limitação é quanto a granularidade da informação obtida pelos sensores.

Os sensores de temperatura e umidade utilizados no protótipo não apresentavam as variações existentes dentro de um valor obtido, por exemplo, uma temperatura de 23,4 ou 23,7°C sempre é informada como 23°C, reduzindo a precisão do contexto a ser gerado com esta informação.

Tabela 11: Comparativo entre o Onisciente e os trabalhos relacionados

	Sheng et. al (2011)	Catarinucci et. al (2009)	Zhang, Zheng, Liu (2008)	Delen et. al (2011)	Kang et. al (2012)	Escribano et. al (2012)	Onisciente
Ontologia	OWL	OWL	OWL-S	-	-	-	OWL
RFID	-	Passivo	-	Semipassivo	Passivo, Ativo	Passivo, Ativo	Passivo, Ativo, Semipassivo
Sensores	Vários (Não específica)	Vários (Não específica)	-	Temperatura	Temperatura	Temperatura, Umidade	Temperatura, Umidade, Acelerômetro, Outros
Modelagem de Contexto	-	-	-	Ontologia, Baseada em regras	Ontologia, Baseada em regras, Multi Agente	Ontologia	Ontologia, Baseada em regras
Web serviços para integração com outras tecnologias	Não específica	Não específica	Não específica	Não específica	Não específica	Não específica	REST
Recursos de visualização de resultados	Sim, mas não específica	Web, Mobile	Não específica	Não específica	Sim, mas não específica	Não específica	Web, Mobile

7.3 Trabalhos Futuros

O Onisciente representa um modelo inicial que foi elaborado a fim de permitir a sua expansão para que seja adaptado ou complementado com o objetivo de adquirir contextos através do uso de outras fontes de dados que não as citadas neste trabalho, além das utilizadas na avaliação. Com isso, alguns trabalhos futuros são apresentados visando a adição de novos contextos.

Primeiramente, considera-se a utilização de leitoras RFID ativas para a detecção de entidades dentro de ambientes. Nos testes realizados foi utilizada uma leitora RFID de contato o que limitou a aplicação desta tecnologia no modelo. Com uma leitora com capacidade de identificar tags com um alcance de alguns metros seria possível usufruir dos recursos de localização desta tecnologia, permitindo determinar o posicionamento de entidades dentro do ambiente em que estão sendo monitoradas enriquecendo a informação de contexto resultante.

Uma possibilidade que está relacionada ao desempenho do Onisciente seria implementar a gestão da ontologia utilizando uma outra linguagem que possua melhores recursos para a sua gestão. A linguagem JAVA seria uma possibilidade, tendo em vista que ela possui alguns frameworks voltados a aplicações com foco em semântica como, por exemplo, o JENA¹⁵.

A incorporação de outros tipos de sensores é tida como um multiplicador da qualidade do resultado da geração de contextos. Ao elevar a variedade de tipos de informação disponível, tem-se a possibilidade de melhorar o nível de detalhamento do contexto a ser gerado, melhorando a precisão da informação a ser apresentada ao usuário final.

Um recurso interessante que acabou não sendo implementado é o sistema de geração de alarme, este recurso poderia liberar um usuário de estar acompanhando as entidades monitoradas, recebendo uma mensagem SMS em seu telefone no momento que um contexto fosse registrado.

¹⁵<http://jena.apache.org/>

REFERÊNCIAS

- ACIMS. **DEVJAVA Modeling and Simulation Tool**. Disponível em: <<http://www.acims.arizona.edu/>>.
- ADELSTEIN F.; GOLDEN, G. R. L. S. S. K. S. G. **Fundamentals of Mobile and Pervasive Computing**. 1. ed. [S.l.]: McGraw-Hill Professional, 2005.
- ALLIANCE, Z. **ZigBee Specification**. Disponível em: <<http://www.zigbee.org/Specifications.aspx/>>.
- ALMEIDA, M. B. A proposal to evaluate ontology content. **Applied Ontology**, Amsterdam, The Netherlands, The Netherlands, v. 4, n. 3-4, p. 245–265, Aug. 2009.
- BOUET, M.; SANTOS, A. dos. RFID tags: positioning principles and localization techniques. In: WIRELESS DAYS, 2008. WD '08. 1ST IFIP, 2008. **Anais...** [S.l.: s.n.], 2008. p. 1–5.
- BOUIADJRA, A. B.; BENSLIMANE, S. M. FOEval: full ontology evaluation. In: NLPKE, 2011. **Anais...** IEEE, 2011. p. 464–468.
- CATARINUCCI, L.; COLELLA, R.; ESPOSITO, A.; TARRICONE, L.; ZAPPATORE, M. A Context-aware Smart Infrastructure based on RFID Sensor-tags and its Application to the Health-care Domain. In: ETFA, 2009. **Anais...** IEEE, 2009. p. 1–8.
- CHEN, G.; KOTZ, D. **A Survey of Context-Aware Mobile Computing Research**. Hanover, NH, USA: Dartmouth College, 2000.
- DELEN, D.; SHARDA, R.; HARDGRAVE, B. The promise of RFID-based sensors in the perishables supply chain. **Wireless Communications, IEEE**, [S.l.], v. 18, n. 2, p. 82–88, april 2011.
- DEY, A. K. Understanding and Using Context. **Personal Ubiquitous Comput.**, London, UK, UK, v. 5, n. 1, p. 4–7, Jan. 2001.
- FERNANDEZ, A.; ABRAHAO, S.; INSFRAN, E. A systematic review on the effectiveness of web usability evaluation methods. In: EVALUATION ASSESSMENT IN SOFTWARE ENGINEERING (EASE 2012), 16TH INTERNATIONAL CONFERENCE ON, 2012. **Anais...** [S.l.: s.n.], 2012. p. 52–56.
- FINKENZELLER, K. **RFID Handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication**. 3. ed. [S.l.]: John Wiley and Sons, 2010.
- FRIEDMAN-HILL, E. **Jess in Action: rule-based systems in java**. [S.l.]: Manning Publications Co., 2003.
- GA-ESCRIBANO, J. J.; DIOS, J. J.; PASTOR, J. M.; GARCIA, A. Improvement in the tracking of special loads by using a three-level RFID system. **International Journal of RF Technologies: Research and Applications**, [S.l.], v. 3, n. 3, p. 181–199, january 2012.

- GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing? **International Journal of Human-Computer Studies**, Duluth, MN, USA, v. 43, n. 5–6, p. 907 – 928, Dec. 1995.
- GU, T.; PUNG, H. K.; ZHANG, D. Q. A middleware for building context-aware mobile services. In: **VEHICULAR TECHNOLOGY CONFERENCE, 2004. VTC 2004-SPRING. 2004 IEEE 59TH**, 2004, Milão, Itália. **Anais...** [S.l.: s.n.], 2004. v. 5, p. 2656 – 2660 Vol.5.
- GU, Y.; LO, A.; NIEMEGERES, I. A survey of indoor positioning systems for wireless personal networks. **Communications Surveys Tutorials, IEEE**, [S.l.], v. 11, n. 1, p. 13 –32, quarter 2009.
- HIGHTOWER J.; BORRIELLO, G. W. R. SpotON : an indoor 3d location sensing technology based on rf signal strength. **Communications**, [S.l.], p. 1–16, 2000.
- HOAREAU, C.; SATOH, I. Modeling and Processing Information for Context-Aware Computing: a survey. **New Generation Computing**, [S.l.], v. 27, p. 177–196, 2009. 10.1007/s00354-009-0060-5.
- KANG, Y.-S.; JIN, H.; RYOU, O.; LEE, Y.-H. A simulation approach for optimal design of RFID sensor tag-based cold chain systems. **Journal of Food Engineering**, [S.l.], v. 113, n. 1, p. 1 – 10, 2012.
- KIM, H.-M.; LEE, K.-H. Device-independent web browsing based on CC/PP and annotation. **Interacting with Computers**, [S.l.], v. 18, n. 2, p. 283 – 303, 2006.
- KOELLE, A.; DEPP, S.; FREYMAN, R. Short-range radio-telemetry for electronic identification, using modulated RF backscatter. **Proceedings of the IEEE**, [S.l.], v. 63, n. 8, p. 1260 – 1261, aug. 1975.
- LANDT, J. The history of RFID. **Potentials, IEEE**, [S.l.], v. 24, n. 4, p. 8 – 11, oct.-nov. 2005.
- LI, N.; BECERIK-GERBER, B. Performance-based evaluation of RFID-based indoor location sensing solutions for the built environment. **Advanced Engineering Informatics**, Amsterdam, The Netherlands, The Netherlands, v. 25, n. 3, p. 535 – 546, Aug. 2011. Special Section: Engineering informatics in port operations and logistics.
- LIU, H.; DARABI, H.; BANERJEE, P.; LIU, J. Survey of Wireless Indoor Positioning Techniques and Systems. **Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on**, Piscataway, NJ, USA, v. 37, n. 6, p. 1067 –1080, nov. 2007.
- MEHRA, P. Context-Aware Computing: beyond search and location-based services. **IEEE Internet Computing**, Piscataway, NJ, USA, v. 16, n. 2, p. 12–16, Mar. 2012.
- MOSTEFAOUI, G. K.; PASQUIER-ROCHA, J.; BREZILLON, P. Context-Aware Computing: a guide for the pervasive computing community. In: **THE IEEE/ACS INTERNATIONAL CONFERENCE ON PERSVASIVE SERVICES**, 2004, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2004. p. 39 – 48.
- NI, L. M.; LIU, Y.; LAU, Y. C.; PATIL, A. P. LANDMARC: indoor location sensing using active rfid. **Wireless Networks**, Hingham, MA, USA, v. 10, n. 6, p. 701–710, Nov. 2004.

NOGUEZ F; COSTA, C. A. B. J. L. V. A MODEL INTEGRATING VIRTUAL WORLDS AND TRAILS. **IADIS International Conference Applied Computing**, Madrid, Espanha, v. 1, p. 1–8, oct. 2012.

NOY, N. F.; MCGUINNESS, D. L. **Ontology Development 101: a guide to creating your first ontology**. Stanford, CA, USA: Stanford Knowledge Systems Laboratory, 2001. (KSL-01-05).

ONO H.; KAWAMURA, T. Sensible Climates in Monsoon Asia. **International Journal of Biometeorology**, [S.l.], v. 35, p. 39–47, 1991.

POVEDA-VILLALON, M.; SUAREZ-FIGUEROA, M.; GOMEZ-PEREZ, A. Validating Ontologies with OOPS! In: TEIJE, A.; VÖLKER, J.; HANDSCHUH, S.; STUCKENSCHMIDT, H.; D'ACQUIN, M.; NIKOLOV, A.; AUSSENAC-GILLES, N.; HERNANDEZ, N. (Ed.). **Knowledge Engineering and Knowledge Management**. [S.l.]: Springer Berlin Heidelberg, 2012. p. 267–281. (Lecture Notes in Computer Science, v. 7603).

ROUSSOS, G.; KOSTAKOS, V. RFID in pervasive computing: state-of-the-art and outlook. **Pervasive and Mobile Computing**, Amsterdam, The Netherlands, The Netherlands, v. 5, n. 1, p. 110–131, Feb. 2009.

SAP. **Standardized Technical Architecture Modeling**. 2007.

SATYANARAYANAN, M. Mobile computing: the next decade. **SIGMOBILE Mob. Comput. Commun. Rev.**, New York, NY, USA, v. 15, n. 2, p. 2–10, Aug. 2011.

SCHILIT, B.; ADAMS, N.; WANT, R. Context-aware computing applications. In: **FIRST WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS**, 1994., 1994, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 1994. p. 85 –90.

SCHILIT, B.; THEIMER, M. Disseminating active map information to mobile hosts. **Network, IEEE**, Piscataway, NJ, USA, v. 8, n. 5, p. 22 –32, sept.-oct. 1994.

SHAMSUZZOHA A.; HELO, P. T. Real-time Tracking and Tracing System: potentials for the logistics network. **INTERNATIONAL CONFERENCE ON INDUSTRIAL ENGINEERING AND OPERATIONS MANAGEMENT**, [S.l.], 2011.

SHENG, J.; ZOU, W.; YANG, L.; WANG, B. A RFID-Based Context-Aware Service Model. In: **IEEE 10TH INTERNATIONAL CONFERENCE ON TRUST, SECURITY AND PRIVACY IN COMPUTING AND COMMUNICATIONS**, 2011., 2011, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2011. p. 779 –784.

SILVA, J.; ROSA, J.; BARBOSA, J.; BARBOSA, D.; PALAZZO, L. Content distribution in trail-aware environments. **Journal of the Brazilian Computer Society**, [S.l.], v. 16, n. 3, p. 163–176, 2010.

STOCKMAN, H. Communication by Means of Reflected Power. **Proceedings of the IRE**, [S.l.], v. 36, n. 10, p. 1196 – 1204, oct. 1948.

WANG, X. H.; ZHANG, D. Q.; GU, T.; PUNG, H. K. Ontology Based Context Modeling and Reasoning using OWL. **Pervasive Computing and Communications Workshops, IEEE International Conference on**, Los Alamitos, CA, USA, v. 0, p. 18, 2004.

WANT, R. An introduction to RFID technology. **Pervasive Computing, IEEE**, [S.l.], v. 5, n. 1, p. 25 – 33, jan.-march 2006.

ZEIGLER B.P., P. H. K. T. **Theory of Modeling and Simulation**. 2. ed. [S.l.]: Academic Press, 2000.

ZHANG, Y.; ZHENG, Q.; LIU, F. An Extended Context Model in a RFID-Based Context-Aware Service System. In: INTERNATIONAL SYMPOSIUM ON INTELLIGENT INFORMATION TECHNOLOGY APPLICATION WORKSHOPS, 2008., 2008, Washington, DC, USA. **Proceedings...** IEEE Computer Society, 2008. p. 693 –697.