



UNIVERSIDADE FEDERAL DO PAMPA  
CIÊNCIA DA COMPUTAÇÃO

IURI ZACARIAS LONDERO

**TÉCNICAS DE COMPUTAÇÃO SENSÍVEL AO CONTEXTO**

Trabalho de Conclusão de Curso

Alegrete  
2012

**IURI ZACARIAS LONDERO**

**TÉCNICAS DE COMPUTAÇÃO SENSÍVEL AO CONTEXTO**

Trabalho de Conclusão de Curso apresentado como parte das atividades para obtenção do título de bacharel em Ciência da Computação na Universidade Federal do Pampa.

Orientador: Prof. Dr. Cléo Zanella Billa

**Alegrete**

**2012**

**IURI ZACARIAS LONDERO**

**TÉCNICAS DE COMPUTAÇÃO SENSÍVEL AO CONTEXTO**

Trabalho de Conclusão de Curso apresentado  
como parte das atividades para obtenção do  
título de bacharel em Ciência da Computação  
na Universidade Federal do Pampa.

Trabalho apresentado e aprovado em: 12 de Julho de 2012.

Banca Examinadora:

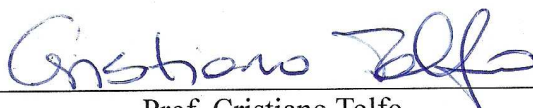


---

Prof. Cleo Zanella Billa

Orientador

Ciência da Computação - UNIPAMPA



---

Prof. Cristiano Tolfo

Ciência da Computação - UNIPAMPA



---

Prof. Sam da Silva Devincenzi

Ciência da Computação - UNIPAMPA

## RESUMO

Este trabalho tem por objetivo aplicar os conceitos de Computação Sensível ao Contexto na prática, tendo como foco principal o estudo das Técnicas de Modelagem de Contexto. Os trabalhos já desenvolvidos nesta área auxiliam novas pesquisas relacionadas ao tema. O desenvolvimento de aplicações que utilizam informações de contexto para produzir resultados de forma automática requer uma técnica de modelagem de contexto. Este trabalho abordou algumas das principais técnicas utilizadas na literatura, comparando-as através de requisitos desejáveis em aplicações sensíveis ao contexto. Foi desenvolvido um protótipo de aplicação utilizando a técnica baseada em ontologias. Esta aplicação levou em consideração um cenário de turismo, onde há muitas informações e que estão em constantes alterações. Para desenvolvimento da ontologia foram necessárias as ferramentas *Protégé* e *SWRL*, sendo utilizadas para modelar as informações de contexto e realizar consultas na Ontologia, sucessivamente. Com os resultados obtidos foi possível notar o comportamento de uma Aplicação Sensível ao Contexto. Por fim, foram apresentadas propostas de melhorias voltadas à aplicação desenvolvida neste trabalho.

Palavras-chave: Computação Sensível ao Contexto, Técnicas de Modelagem de Contexto, Informações de Contexto, Contexto, Ontologia.

## **ABSTRACT**

This work aims to apply the concepts of Context Sensitive Computing in practice, with the main focus of the study Context Modeling Techniques. The work already undertaken in this area helps further related research. Developing applications that use context information to produce results automatically requires a modeling technique of context. This study addressed some of the main techniques used in the literature, comparing them across desirable requirements in context-aware applications. We developed a prototype application using a technique based on ontologies. This application took into account a scenario of tourism, where there are many information and are in constant change. For ontology development tools were necessary Protégé and SWRL, being used to model the context information and consult on Ontology, successively. With the results we note the behavior of an Application Context Sensitive. Finally, there were also proposals for improvements aimed at the application developed in this work.

**Keywords:** Context Aware Computing, Modeling Techniques Context, Context Information, Context, Ontology.

## LISTA DE ILUSTRAÇÕES

Figura 1: Tela do horário parcial exibindo a localização e o nível de interesse dos colegas. Os Símbolos indicam o nível de interesse (DEY et al., 1999).....	17
Figura 2: (a) Usuário consultando a aplicação <i>Friend Locator</i> para encontrar um amigo. (b) Interface da aplicação que mostra a posição corrente do usuário ( <i>You</i> no mapa) e a que distância seu amigo (David) se encontra (250 metros) na direção de um local chamado Pampas (OLOFSSON et al., 2006).....	18
Figura 3: Tela da aplicação de simulador de localização (esquerda), aplicação da sala sensível ao contexto (centro), e a interface do serviço de notificação (direita) (HEER et al., 2003).....	19
Figura 4: Representação de informações de Localização.....	23
Figura 5: Perfil de localização usando CC/PP.....	24
Figura 6: Representação de informações de contexto com ORM.....	26
Figura 7: Contexto representado por um Modelo Orientado a Objetos.....	27
Figura 8: Exemplo de representação de ontologia.....	29
Figura 9: Exemplo de representação da ontologia em OWL.....	32
Figura 11: Interface gráfica da Ferramenta <i>Protégé</i> .....	33
Figura 10: Exemplo de regra SWRL.....	32
Figura 12: Visão geral da Ontologia.....	35
Figura 13: Representação da relação de classes entre si e a relação de classes com dados.....	36
Figura 14: Resultados de quais usuários tem interesse em determinada área.....	39
Figura 15: Resultados das atrações exibidas para o usuário João.....	41

## LISTA DE TABELAS

Tabela 1: Comparativo entre técnicas de modelagem .....	31
Tabela 2: Instâncias de usuários. ....	38
Tabela 3: Instâncias de Atrações turísticas.....	40

## LISTA DE ABREVIATURAS E SIGLAS

ASC – *Aspect-Scale-ContextInformation*

AT&T – *American Telephone and Telegraph*

CAPEUS – *Context-Aware Packets Enabling Ubiquitous Services*

CC/PP – *Composite Capability/Preference Profiles*

CoOL – *Context Ontology Language*

CSCP – *Comprehensive Structured Context Profiles*

GPS – *Sistema de Posicionamento Global*

ORM – *Modelling Object-Role*

OWL – *Ontology Web Language*

PDA – *Assistente Pessoal Digital*

SGML – *Standard Generic Markup Language*

XML – *Extensible Markup Language*

UAProf – *User Agent Profile*

UML – *Unified Modeling Language*

UNIPAMPA – *Universidade Federal do Pampa*



## SUMÁRIO

<b>RESUMO.....</b>	<b>4</b>
<b>ABSTRACT .....</b>	<b>5</b>
<b>LISTA DE ILUSTRAÇÕES.....</b>	<b>6</b>
<b>LISTA DE TABELAS .....</b>	<b>7</b>
<b>LISTA DE ABREVIATURAS E SIGLAS .....</b>	<b>8</b>
<b>SUMÁRIO.....</b>	<b>9</b>
<b>1 INTRODUÇÃO .....</b>	<b>11</b>
<b>1.1 ORGANIZAÇÃO DO TRABALHO .....</b>	<b>11</b>
<b>2 FUNDAMENTOS TEÓRICOS.....</b>	<b>13</b>
<b>2.1 CONTEXTO .....</b>	<b>13</b>
<b>2.2 SISTEMAS SENSÍVEIS AO CONTEXTO .....</b>	<b>14</b>
<b>2.3 APLICAÇÕES SENSÍVEIS AO CONTEXTO X APLICAÇÕES TRADICIONAIS</b> <b>16</b>	
<b>2.4 EXEMPLOS DE APLICAÇÕES SENSÍVEIS AO CONTEXTO .....</b>	<b>16</b>
<b>2.4.1 CONFERENCE ASSISTANT .....</b>	<b>16</b>
<b>2.4.2 FRIEND LOCATOR.....</b>	<b>17</b>
<b>2.4.3 CO-OCCUPANT AWARENESS.....</b>	<b>18</b>
<b>2.4.4 SHOPPING ASSISTANT .....</b>	<b>19</b>
<b>3 TRABALHOS RELACIONADOS .....</b>	<b>20</b>
<b>3.1 MODELOS DE REPRESENTAÇÃO DE CONTEXTO.....</b>	<b>20</b>
<b>3.1.1 MODELO CHAVE-VALOR.....</b>	<b>22</b>
<b>3.1.2 MODELO DE MARCAÇÃO .....</b>	<b>23</b>
<b>3.1.3 MODELO GRÁFICO .....</b>	<b>25</b>
<b>3.1.4 MODELO ORIENTADO A OBJETOS.....</b>	<b>26</b>
<b>3.1.5 MODELO BASEADO EM LÓGICA .....</b>	<b>27</b>
<b>3.1.6 MODELO BASEADO EM ONTOLOGIAS.....</b>	<b>28</b>
<b>4 METODOLOGIA.....</b>	<b>30</b>
<b>4.1 VISÃO GERAL DA APLICAÇÃO .....</b>	<b>30</b>
<b>4.2 COMPARATIVO ENTRE AS TÉCNICAS .....</b>	<b>30</b>
<b>4.2.1 TÉCNICA SELECIONADA .....</b>	<b>31</b>
<b>4.2.2 OWL .....</b>	<b>32</b>
<b>4.2.3 SWRL .....</b>	<b>32</b>

<b>4.2.4 PROTÉGÉ.....</b>	<b>33</b>
<b>4.3 MODELAGEM DA APLICAÇÃO.....</b>	<b>34</b>
<b>5 RESULTADOS .....</b>	<b>38</b>
<b>5.1 CASO 1.....</b>	<b>38</b>
<b>5.2 CASO 2.....</b>	<b>40</b>
<b>6 CONCLUSÃO.....</b>	<b>43</b>
<b>7 REFERÊNCIAS .....</b>	<b>44</b>

## **1 INTRODUÇÃO**

Existe uma demanda e tendência em se desenvolver aplicações capazes de processar resultados sem a interação direta do usuário. Esse tipo de aplicação em que não há a necessidade de o usuário interagir de forma direta com o computador denomina-se aplicação sensível ao contexto.

Enquanto, em um sistema computacional tradicional, o programador precisa projetar um software pensando em todas as possibilidades de entradas, num sistema sensível ao contexto, o programador preocupa-se em modelar o contexto, com informações que sejam relevantes dentro do domínio da aplicação, tornando o sistema mais confiável, pois são tratados apenas dados que sejam úteis dentro do cenário do sistema.

No desenvolvimento de uma aplicação sensível ao contexto leva-se em conta todo o cenário ao redor, como o tempo, hora, localização, entre outros. A definição das informações relevantes dentro do contexto de uma aplicação é a primeira etapa para que um sistema sensível ao contexto seja desenvolvido, porém, para implementação desse sistema é preciso definir uma técnica para modelagem dessas informações.

Existem várias técnicas de modelagem que podem ser utilizadas, e neste trabalho foram abordadas algumas das principais, objetivando o estudo dessas técnicas e definir qual a mais adequada a determinado contexto, visando analisar os resultados práticos dessa abordagem. Para atingir este objetivo definiu-se um contexto de aplicação (cenário de turismo) onde por meio de simulações foi possível realizar as análises pretendidas. Outro fator importante na conclusão do trabalho foi o apoio a novas aplicações, sendo produzido material relevante para subsidiar novos trabalhos relacionados ao tema.

### **1.1 ORGANIZAÇÃO DO TRABALHO**

Este trabalho está dividido em sete seções:

A seção 1 compreende a introdução do trabalho, trazendo a motivação e os objetivos que levaram ao desenvolvimento do mesmo.

A seção 2 trata dos fundamentos teóricos, que serviram como base aos conceitos de Computação Sensível ao Contexto.

A seção 3 trata dos trabalhos relacionados, possuindo um detalhamento das técnicas de modelagem de contexto.

A seção 4 trata da metodologia, apresentando as fases de desenvolvimento do trabalho.

A seção 5 trata dos resultados obtidos.

A seção 6 trata da conclusão e trabalhos futuros.

A seção 7 apresenta as referências bibliográficas que proporcionaram o desenvolvimento.

## 2 FUNDAMENTOS TEÓRICOS

A computação sensível ao contexto tem sido um tema muito debatido dentro da ciência da computação. Os trabalhos já desenvolvidos contribuem cada vez mais para novas pesquisas relacionadas ao assunto.

Para compreender melhor esse novo paradigma de Computação, algumas definições devem ser compreendidas antes de começar a desenvolver aplicações que se baseiam em contexto.

### 2.1 CONTEXTO

Denotativamente contexto é o conjunto de circunstâncias que acompanham um acontecimento. Dentro da ciência da computação os significados de contexto se diferenciam um pouco, já que a maioria dos pesquisadores preferiu delimitar o tema, pois acreditam que contexto é algo muito amplo.

Schilit, Adams e Want (1994) acreditam que contexto pode ser dividido em três categorias:

- *Contexto de computação*, como conectividade de rede, os custos de comunicação e largura de banda de comunicação, e recursos próximos, tais como impressoras, monitores e estações de trabalho.
- *Contexto do usuário*, como o perfil do usuário, a localização, as pessoas próximas, até a situação social atual.
- *Contexto físico*, tais como iluminação, ruído, condições de tráfego e da temperatura.

Chen e Kotz (2000) também seguem a mesma linha anterior, porém propuseram uma quarta categoria:

- *Contexto do tempo*, como à hora do dia, semana, mês e estação do ano.

As categorias mencionadas, analisando a computação, usuário, meio físico e tempo são os meios utilizados para que as informações se desenvolvam e sejam definidas as ações adequadas a cada situação. Através dessas categorias pode ser caracterizada uma situação e com as informações de contexto prover serviços capazes de agir e reagir conforme as necessidades do usuário.

A primeira definição de informação de contexto relatada na literatura foi dada por Schilit e Theimer (1994, p. 1) na qual informação de contexto “seriam informações de

localização, de identidade de pessoas e de objetos próximos entre si e das mudanças nesses objetos”.

Para Dey (2001), informação de contexto é qualquer informação que possa ser utilizada para caracterizar uma entidade. Uma entidade é uma pessoa, lugar ou objeto considerado relevante para uma interação entre um usuário e uma aplicação, incluindo o usuário e aplicação em questão.

Partindo dos conceitos acima, podem-se definir informações de contexto como toda a informação que será importante dentro do cenário da aplicação, ou seja, aquelas informações que terão influência no momento em que está sendo executada a aplicação.

Bulcão (2006) acredita que as informações de contexto podem ser obtidas de duas formas: *explícitas*, quando a informação obtida é expressa intencionalmente pelo usuário; ou *implícita* quando a informação é obtida sem a comunicação intencional do usuário;

Esse processo de busca de informações irá servir como apoio ao desenvolvimento de aplicações adaptadas ao usuário. Um exemplo de aplicação na qual o usuário utilizaria a informação explícita seria um programa iniciado por um comando do usuário e informação implícita, as pessoas que estão ao redor em um ambiente físico.

## 2.2 SISTEMAS SENSÍVEIS AO CONTEXTO

Analisando o que foi falado sobre contexto anteriormente, uma aplicação é denominada sensível ao contexto quando usa informações de contexto para prover serviços capazes de adaptarem-se ao usuário.

Uma das definições bem aceitas na comunidade de Computação Sensível a Contexto é dada por Dey, Salber e Abowd (2001), onde uma aplicação é classificada como sensível ao contexto quando utiliza informações de contexto para fornecer serviços e/ou outras informações relevantes a um usuário, onde a relevância está diretamente relacionada à tarefa que o usuário desempenha em um dado momento.

Esses serviços sensíveis a contexto surgem como alternativas ao uso de aplicações tradicionais, embora sejam mais complexas na fase de projeto e desenvolvimento, um dos principais objetivos é fornecer métodos que sejam “inteligentes” no momento de agir, ou seja, conhecer o ambiente em que se encontra e partindo desse conhecimento tomar decisões de forma autônoma.

Schilit e Theimer (1994) conceituam aplicações sensíveis ao contexto organizando-as da seguinte forma: as aplicações que adaptam, de forma manual ou automática, a apresentação de informações a usuários; e as aplicações que executam comandos de forma manual ou automática, associados a informações de contexto.

As aplicações sensíveis ao contexto podem agir manual ou automaticamente dependendo das mudanças e as necessidades do usuário, ou seja, a apresentação da informação pode ser executada através da intervenção do usuário ou pelo comando da própria aplicação.

Segundo Chen e Kotz (2000), o contexto pode ser definido em função de seu efeito sobre uma aplicação: contexto ativo e passivo. Contexto ativo é quando a aplicação adapta-se automaticamente para descobrir contexto, mudando o comportamento da aplicação, já no contexto passivo uma aplicação apresenta um contexto novo ou atualizado para o interesse do usuário, ou guarda um histórico do contexto para reutilizá-lo posteriormente.

Neste caso a aplicação de contexto ativo adapta o seu comportamento de forma automática ao contexto percebido, e na aplicação de contexto passivo apresenta ao usuário a informação de acordo com o contexto.

Para Pascoe (1998) as aplicações sensíveis a contexto podem ter outras características que não estão necessariamente presente em todas as aplicações, sendo elas: percepção contextual, adaptação contextual, descoberta de recursos contextuais e expansão contextual.

Dey (2000) se baseou nas características levantadas por Pascoe (1998) e propôs uma categorização que uma aplicação sensível a contexto pode atender e explicita que através dessa categorização é possível saber que tipos de características devem ser considerados no projeto e no desenvolvimento de aplicações sensíveis a contexto. A categorização que Dey (2000) defende baseia-se na apresentação de informações e serviços para o usuário, execução automática de um serviço e a união de informações de contexto.

Os pensamentos de Pascoe (1998) sobre as características de uma aplicação e posteriormente a categorização citada por Dey (2000), são passos relevantes para uma aplicação, pois como já foi falado, através dessas três categorias podem-se descobrir as características que possivelmente serão consideradas no desenvolvimento de uma aplicação.

## **2.3 APLICAÇÕES SENSÍVEIS AO CONTEXTO X APLICAÇÕES TRADICIONAIS**

Partindo das definições de aplicações sensíveis a contexto, é possível notar algumas diferenças em relação às aplicações computacionais tradicionais. Uma das principais diferenças é que aplicações tradicionais têm características mais estáticas e que se baseiam em ambientes desktops, enquanto aplicações sensíveis a contexto exploram o contexto dos diversos ambientes para produzir serviços mais adaptativos e voltados ao usuário (NEVES et al., 2010).

Outro ponto que difere essas aplicações é a fase de modelagem do sistema. As aplicações tradicionais requerem o tratamento de todos dados possíveis de entrada, enquanto aplicações sensíveis a contexto necessitam apenas das entradas que terão utilidade dentro do contexto da aplicação. Se uma aplicação irá usar a hora, a hora deverá ser levada em conta na modelagem, caso seja colocado outro dado diferente a aplicação apenas ignora.

Embora aplicações sensíveis a contexto sejam mais complexas para desenvolver, elas garantem confiabilidade ao sistema, como citado anteriormente, elas tratam apenas informações que são úteis dentro do contexto.

## **2.4 EXEMPLOS DE APLICAÇÕES SENSÍVEIS AO CONTEXTO**

A seguir são apresentadas algumas aplicações que utilizam informações de contexto para fornecerem serviços aos usuários, destacando o objetivo de cada aplicação e as informações que cada uma utiliza.

### **2.4.1 CONFERENCE ASSISTANT**

Alguns pesquisadores do Instituto de Tecnologia da Georgia nos Estados Unidos da América (DEY et al., 1999) desenvolveram essa aplicação, que consiste em auxiliar os participantes em conferências. A aplicação funciona da seguinte forma: quando um participante da conferência chega ao local, registram seus dados pessoais, os temas de pesquisa do seu interesse e colegas que participarão. Durante a conferência os participantes terão em mãos um Assistente Pessoal Digital (PDA) que o auxiliará apresentando os temas de seu interesse que estão sendo abordados, como também a localização dos outros participantes, pois a medida que se deslocam são monitorados, isso pode ser útil também no sentido que,



quando um tema se torna interessante para um dos participantes todos os outros ficarão sabendo e poderão ter também em seu PDA o conteúdo apresentado da conferência que seu colega participou. Na Figura 1 é apresentada uma tela deste trabalho, onde é possível identificar o nível de interesse dos usuários.

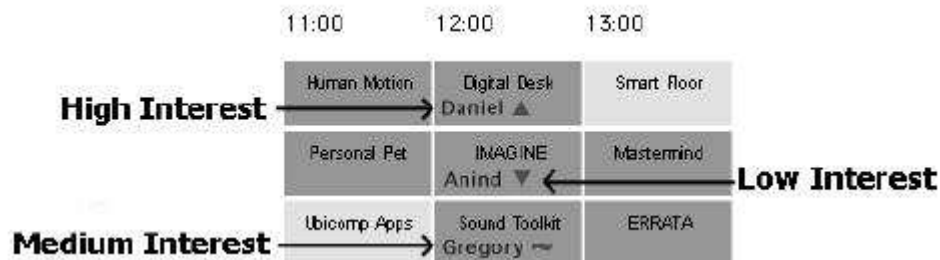
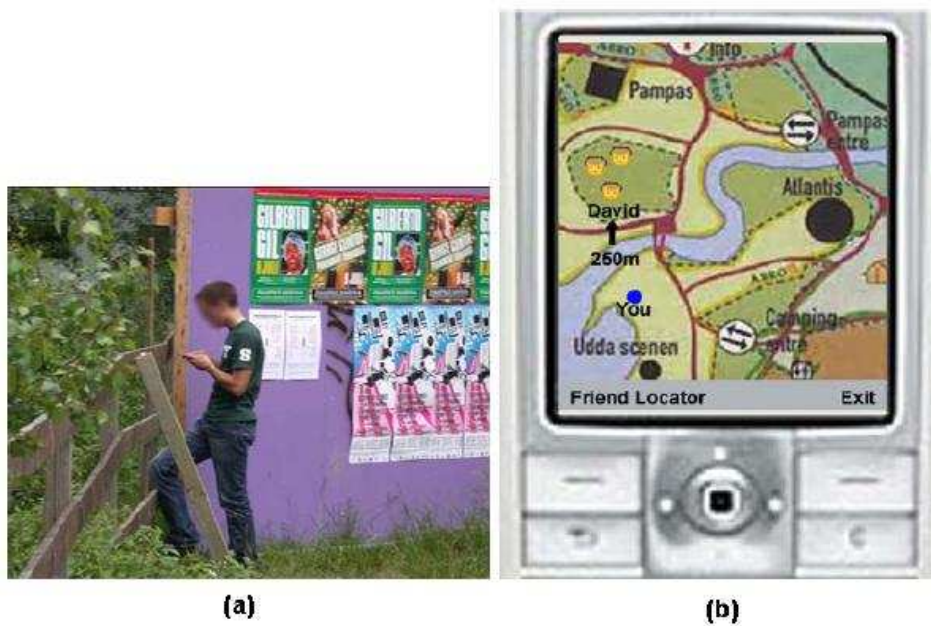


Figura 1: Tela do horário parcial exibindo a localização e o nível de interesse dos colegas. Os Símbolos indicam o nível de interesse (DEY et al., 1999).

#### 2.4.2 FRIEND LOCATOR

Essa aplicação foi desenvolvida por pesquisadores do Instituto de Tecnologia de Blekinge na Suécia (OLOFSSON et al., 2006). Sabendo que algumas pessoas gostam de ir a festas e frequentar espaços sociais acompanhadas de um grupo de amigos, essa aplicação vem para auxiliar, pois pode fornecer o local exato em que os amigos estão, bem como, não terão o constrangimento de falar ao celular em um local com ruídos.

Para localizar seus amigos o usuário irá cadastrá-los no *Friend Locator* quando houver a busca do nome procurado a aplicação irá utilizar mecanismos como um GPS e dará a distância exata do amigo perdido. Na Figura 2 é exibido um usuário procurando por seu amigo.



**Figura 2: (a) Usuário consultando a aplicação *Friend Locator* para encontrar um amigo. (b) Interface da aplicação que mostra a posição corrente do usuário (*You* no mapa) e a que distância seu amigo (*David*) se encontra (250 metros) na direção de um local chamado *Pampas* (OLOFSSON et al., 2006).**

### 2.4.3 CO-OCCUPANT AWARENESS

A aplicação seguinte foi desenvolvida por pesquisadores da Universidade de Berkeley nos Estados Unidos da América (HEER et al., 2003). A aplicação consiste em localizar pessoas em um mesmo ambiente fechado de grande extensão, ou seja, notificará quando pessoas entram ou saem na mesma sala disponibilizando informações a respeito dessas pessoas facilitando a comunicação interpessoal.

Esta aplicação simula a movimentação de pessoas e objetos através de uma interface *web*, sem a necessidade de sensores e rede de comunicação, assim, quando uma pessoa adentrar na sala, é notificado pelo simulador que recebe a identidade para saber qual é a pessoa e, em seguida, compartilha o endereço eletrônico e paginas *web* de todos ocupantes da sala. Nessa aplicação existe também uma política de privacidade, onde os dados do usuário serão usados somente se ele permitir. Na Figura 3 são exibidas as telas principais desta aplicação.

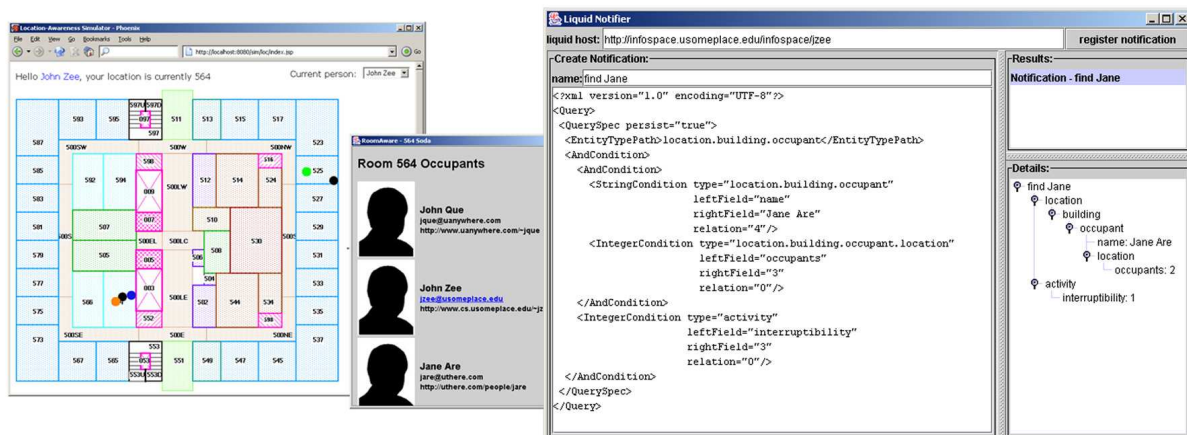


Figura 3: Tela da aplicação de simulador de localização (esquerda), aplicação da sala sensível ao contexto (centro), e a interface do serviço de notificação (direita) (HEER et al., 2003).

#### 2.4.4 SHOPPING ASSISTANT

Esta aplicação foi desenvolvida pelo grupo AT&T *Bell Laboratories* (ASTHANA et al., 1994). Consiste em um guia que auxilia compradores fornecendo detalhes de itens, localizando os itens, fazendo comparativos de preços, entre outros.

A loja mantém os perfis dos clientes, uma preocupação é quanto à privacidade, com isso a aplicação classifica os clientes em duas classes: clientes regulares que fazem compras de forma anônima, e os que têm perfil na loja e expõe sua identidade ganham descontos adicionais.

### 3 TRABALHOS RELACIONADOS

Para o desenvolvimento de aplicações sensíveis ao contexto é necessário entender as técnicas usadas para modelagem de contexto. Essas técnicas são modelos que servem como estrutura para apoiar o momento de modelagem de uma aplicação.

A partir desta seção serão apresentadas várias técnicas que são utilizadas para a modelagem de contexto.

#### 3.1 MODELOS DE REPRESENTAÇÃO DE CONTEXTO

Nesta unidade serão apresentadas algumas das principais técnicas utilizadas para modelagem de contexto, detalhando cada técnica e suas peculiaridades, com o objetivo de prover mecanismos que sirvam de base para o desenvolvimento de novas aplicações relacionadas ao paradigma de Computação Sensível ao Contexto.

As modelagens de contexto existentes diferem em vários aspectos, como: facilidade de captura dos conceitos do mundo real pelos engenheiros de software, no poder expressivo dos modelos de informações de contexto, nos mecanismos fornecidos para apoiar o raciocínio sobre informações de contexto, no desempenho computacional deste raciocínio e na escalabilidade do gerenciamento de informações de contexto.

Bettini et al., (2010), consideram inicialmente um conjunto de requisitos para os modelos de contexto, são eles:

- *Heterogeneidade e mobilidade*: como as fontes de informações para aplicações sensíveis a contexto são várias, diferindo na taxa de atualização e nível semântico, é interessante o modelo da aplicação, gerenciar esse tipo de requisito.
- *Relacionamentos e dependências*: é interessante o modelo gerenciar a relação e dependência entre as informações para garantir o comportamento correto das aplicações, por exemplo, a largura de banda aumentando pode afetar o desempenho da bateria em um dispositivo móvel, logo, são informações distintas, mas que possuem relação.
- *Histórico (timeliness)*: as aplicações podem precisar de acesso a estados históricos ou previsão de estados futuros, logo, essa característica deve ser gerenciada pelo modelo. Um dos pontos que preocupa nessa característica são a quantidade de informações e o nível com que são atualizadas, com isso, algumas técnicas de sumarização podem ser necessárias, por exemplo.

- *Imperfeição*: devido a sua natureza dinâmica e heterogênea, o modelo deve ser capaz de tratar as mudanças de qualidade das informações de contexto para produzir a modelagem correta. Por exemplo, a posição de um GPS em poucos metros pode ser imprecisa, e se não for tratada pode aumentar a imprecisão das informações com o tempo.
- *Raciocínio*: como as aplicações utilizam informações de contexto para avaliar se há mudanças para o usuário, decidir tomar alguma decisão pode requerer capacidades de raciocínio. Portanto, as técnicas de modelagem, além da verificação de consistência, devem ser capazes de prover raciocínio sobre o contexto. Essas técnicas precisam ser eficientes.
- *Usabilidade de formalismos de modelagem*: como as aplicações sensíveis ao contexto utilizam informações que são modeladas, as técnicas de modelagem devem prover mecanismos que sejam usuais e formais para facilitar a modelagem dos conceitos do mundo real e facilitar a utilização das informações de contexto em tempo de execução.
- *Eficiente acesso ao contexto*: embora seja uma tarefa difícil de cumprir em modelos grandes e com muitos dados de objetos, prover acesso eficiente às informações de contexto é necessário. Para isso, no momento da modelagem devem ser representados corretamente os atributos para os caminhos que acessam os objetos relevantes.

Segundo Barth (2004) há um conjunto de requisitos para avaliar as técnicas de modelagem de contexto:

- *Composição distribuída*: a composição e administração dos modelos de contexto são extremamente dinâmicas em termos do tempo, topologia da rede e recursos.
- *Validação Parcial*: capacidade para validar conhecimento de forma parcial. Devido à composição distribuída, nem sempre é possível validar todo conhecimento relacionado ao contexto.
- *Qualidade da informação*: a qualidade da informação pode variar de acordo com o sensor utilizado, entre outros fatores. Os métodos devem suportar informações com distintos níveis de qualidade.
- *Nível de formalidade*: modelos devem possuir sintaxe e semânticas bem definidas.
- *Aplicabilidade em ambientes já existentes*: utilizar tais modelos em aplicações já existentes. Por exemplo: utilizar com *Web Services*.

Estes requisitos são o passo inicial para analisar as técnicas de modelagem existentes, com eles é possível ver qual a melhor técnica para determinado tipo de contexto.

A seguir são detalhadas algumas das principais técnicas utilizadas para o

desenvolvimento de aplicações sensíveis a contexto.

### 3.1.1 MODELO CHAVE-VALOR

A modelagem de contexto baseada em Chave-Valor, juntamente com modelos de Marcação, é uma das primeiras técnicas utilizadas com o objetivo de apoiar os engenheiros de software a modelar informações de contexto.

Esse modelo trata de simples pares de chave-valor que definem uma lista de atributos e valores descritores de informações de contexto usadas pelas aplicações sensíveis a contexto. Embora seja um modelo fácil de gerir, esbarram na falta de recursos para suportar estruturas sofisticadas que permita trabalhar com algoritmos eficientes de recuperação de contexto.

Schilit, Adams e Want (1994) utilizaram pares chave-valor para modelar contexto, utilizando informações como o nome para referenciar um valor de localização. Neste trabalho o autor se utiliza de dispositivos móveis que usam uma rede de celular baseada em infravermelho para comunicação. O contexto do sistema abrange uma sala, onde cada usuário portando seu dispositivo é capaz de conectar a rede de celulares local e a partir, consegue-se obter a localização das pessoas que estão nesse espaço, pode acontecer com o passar do tempo à informação sobre localização se tornar imprecisa, para ter dados precisos o sistema pode ser programado para que de tempos em tempos envie pacotes para atualizar a localização dos usuários.

O modelo chave-valor é usado frequentemente em estruturas de serviços distribuídos, Samulowitz et al. (2001) desenvolveram uma arquitetura sensível a contexto para seleção e execução de serviços chamada *Context-Aware Packets Enabling Ubiquitous Services* (CAPEUS), onde os usuários através de um dispositivo móvel podem acessar uma variedade de serviços em seu ambiente, por exemplo, impressão de documentos. A arquitetura é capaz de descobrir qual a impressora está mais próxima do usuário quando ele enviar o arquivo para ser impresso, para isso ocorrer de forma correta o usuário deve estar conectado na rede local do ambiente, independente do tipo de rede. Os autores utilizam mecanismos de modelagem chave-valor para descrever as entidades do sistema, neste caso a impressora é uma entidade e seus atributos como, cor, tecnologia e tipo de papel, são associados a valores, por exemplo, branca, laser, A4, respectivamente.

Conforme a Figura 4 pode-se ver um exemplo de informação de contexto usada em uma aplicação que se baseie em localização. Embora seja uma representação ilustrada da informação, a forma como ela vai ser representada computacionalmente irá depender do

projetista de software.

NOME	SALA	DISTÂNCIA
João	201	30m
Maria	301	50m
Joana	303	55m
Marcos	102	20m

**Figura 4: Representação de informações de Localização**

A vantagem desse tipo de abordagem é que independentemente da informação que seja buscada, o sistema pode chegar a essa informação de várias formas, no exemplo dado, para descobrir a distância que o usuário da aplicação está do usuário João, pode-se buscar diretamente pelo nome do usuário João, ou procura-se os usuários de cada sala. Essa técnica não necessariamente irá restringir o uso dessas informações como determinado parâmetro, no exemplo, poderíamos chegar à distância do usuário buscando pelo nome, mas o contrário também é válido, poderíamos procurar pelo usuário que estivesse a 30 metros de distância.

Essa forma de modelar informações de contexto remete a informações estáticas e difíceis de relacionar com outros tipos de informações.

### 3.1.2 MODELO DE MARCAÇÃO

Essa técnica é uma das primeiras encontradas na literatura de Computação Sensível ao Contexto. Esse modelo é composto por uma estrutura de dados hierárquica formada por *tags* de marcação que contém atributos e conteúdos.

Geralmente, esse tipo de abordagem é usado para representação de perfis, que serão descritos em linguagens de marcação derivadas do *Standard Generic Markup Language* (SGML), um exemplo de linguagem derivada do SGML é o *Extensible Markup Language* (XML).

Embora seja uma abordagem simples e que use métodos de linguagens de marcação, que são bastante usuais dentro da Ciência da Computação, essa técnica não consegue abranger alguns requisitos desejáveis para aplicações baseadas em contexto, como se trata de *tags* que irão descrever o contexto, pode-se perder os relacionamentos, as dependências, o histórico e a qualidade das informações. No entanto, são usadas algumas extensões como *Composite Capability/Preference Profiles* (CC/PP) e *User Agent Profile* (UAProf) que fornecem

métodos avançados para descrever informações de contexto e de usuário, tornando a descrição dessas informações mais robustas e menos estáticas.

Held et al. (2002), analisou a ferramenta CC/PP e propôs o *Comprehensive Structured Context Profiles* (CSCP) afim de suprir as principais deficiências existentes no CC/PP. O CSCP também mantém estrutura semelhante ao XML, porém não define hierarquias fixas para as informações, o que era um ponto negativo na descrição de informações de contexto, este mecanismo se tornou flexível e abriu espaço para o desenvolvimento de aplicações em geral baseadas em contexto.

Indulska et al (2003) analisou o CC/PP e propôs uma abordagem semelhante ao CSCP, o CC/PP *Context Extension* estende o CC/PP básico e o UAProf, relacionando alguns aspectos importantes do contexto em forma de árvores, nessa abordagem cada componente irá ser representado como uma árvore que irá conter seus atributos, por exemplo, informação de localização pode conter localização física, localização geográfica, entre outros. Nessa extensão do CC/PP é possível relacionar os componentes, porém pode ser difícil relacionar informações contextuais complexas.

Na Figura 5 há um exemplo de modelagem de informações de localização usando CC/PP que abrange vários tipos de localização, onde cada tipo possui seus atributos e fazem parte do *PerfilLocalização*, ou seja, dentro do *PerfilLocalização* estão incluídas a Localização Física, Lógica, Geográfica, a Orientação e Modificações.

```
[PerfilLocalização
  [LocalizaçãoFísica [País, Estado, Cidade, Bairro]]
  [LocalizaçãoLógica [EndereçoIP]]
  [LocalizaçãoGeográfica [Longitude, Latitude, Altitude]]
  [Orientação [AngRumo, AngPicada]]
  [Modificações [ErroVertical, ErroHorizontal, ErroRumo, ErroPicada]]
]
```

**Figura 5: Perfil de localização usando CC/PP**

Esta forma de representar informações de contexto garante a consistência das informações, como já foi dito, em linguagens de marcação puras poderia haver perda de relacionamentos, histórico, dependências e qualidade da informação. No caso das relações entre os componentes, pode-se modelar um perfil que seja responsável por referenciar quais



perfis estão relacionados, por exemplo, o perfil do usuário deve manter uma relação com o perfil de localização, essa relação será representada em forma de outro perfil, se as informações sobre o usuário e localização fossem representadas no mesmo perfil, poderia gerar-se informação inconsistente, perdendo as relações entre as mesmas, essa perda de relacionamentos foge do objetivo de aplicação sensível ao contexto.

Esse método CC/PP pode remeter a perfis estáticos, depende da forma como será projetada a aplicação e qual o objetivo principal.

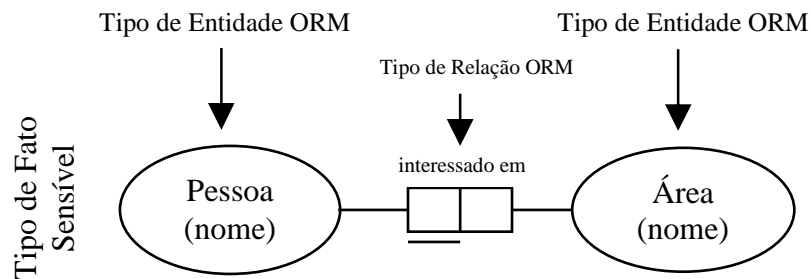
### 3.1.3 MODELO GRÁFICO

Como o próprio nome sugere, esse modelo usa mecanismos gráficos para representar as informações contextuais. Em aplicações de uso geral uma das ferramentas mais conhecidas usadas para modelagem é a *Unified Modeling Language* (UML), que possui forte componente gráfico e estrutura genérica, possibilitando a modelagem de informações de contexto.

Em seu trabalho, Bauer (2003) modelou os aspectos relevantes para gerenciamento do tráfego aéreo utilizando extensões UML. No contexto da aeronave, a turbulência é uma entidade que pode ser ativada caso aconteça mudanças no clima, por exemplo. Bauer (2003) usou diagramas de classes para descrever essas entidades, definindo atributos considerados relevantes para cada entidade, temperatura, velocidade do vento, são exemplos de atributos interessantes para descrever uma turbulência.

Outro exemplo que aborda modelos gráficos foi desenvolvido por Henricksen et al. (2002). Essa abordagem é uma extensão da modelagem baseada no papel do objeto (*Modelling Object-Role* - ORM). Em ORM, o conceito de modelagem básica é o fato, e a modelagem de um domínio usando ORM envolve a identificação apropriada de tipos de fatos e as funções que cada entidade desempenha. Henricksen et al. (2002) estendeu ORM para tipos de fatos categorizados, de acordo com a persistência e fontes, estáticas ou dinâmicas. Nessa extensão, outro indicador de qualidade introduzido é tipo de fatos históricos para cobrir aspectos temporais do contexto, também foi desenvolvida uma proposta para modelar as dependências dos fatos, na qual representa um tipo especial de relações entre fatos, onde uma alteração em um fato provoca automaticamente uma modificação em outro fato.

Na Figura 6 é mostrado um exemplo de representação gráfica usando ORM, são destacados os componentes principais desse tipo de representação.



**Figura 6: Representação de informações de contexto com ORM**

Lembrando que a abordagem ORM considera as informações contextuais como fatos, na Figura 6 é possível notar a representação dos mesmos. No tipo de fato sensível têm-se duas entidades relacionadas (*Pessoa/Área*) através do relacionamento *interessado em*.

### 3.1.4 MODELO ORIENTADO A OBJETOS

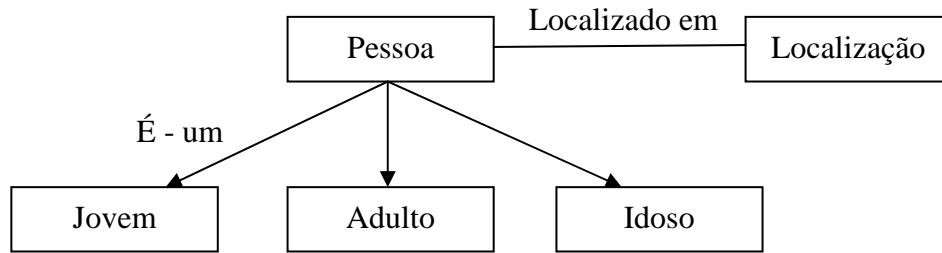
A modelagem orientada a objetos tem como objetivo o encapsulamento e reutilização de informações de contexto, para cobrir problemas comuns em aplicações onde o contexto é dinâmico.

Nessa abordagem, os detalhes de processamento do contexto são encapsulados em nível de objeto, ocultando-os para os demais componentes da aplicação. O acesso às informações contextuais é fornecido através de interfaces específicas.

Um exemplo onde essa abordagem é usada são os *cues* que estão inseridos no TEA *Project*, cujos mecanismos produzem uma abstração de sensores físicos e lógicos (SCHMIDT et al., 1999). O *cue* é considerado uma função para obter o valor de um único sensor físico ou lógico por um determinado tempo como entrada, gerando uma saída simbólica ou sub-simbólica. Um conjunto finito ou infinito de possíveis valores é definido para cada *cue*. A saída de cada *cue* depende de um único sensor, mas diferentes *cues* podem ser baseados nos mesmos sensores.

Outra abordagem dentro da categoria de objeto é o *Active Object Model* do projeto GUIDE (CHERVEST et al., 1999). Esta abordagem foi escolhida pela necessidade de gerenciar várias informações de contexto pessoais ou de ambiente, mantendo escalabilidade. Os objetos ativos serão responsáveis por encapsular todos os detalhes de coletas de dados e relações, ocultando esses detalhes dos demais componentes do sistema.

Na Figura 7, é apresentado um exemplo de representação de contexto usando modelagem orientada a objetos.



**Figura 7: Contexto representado por um Modelo Orientado a Objetos**

Nesta técnica de modelagem, as informações são modeladas usufruindo das principais vantagens do paradigma orientado a objetos (herança, encapsulamento, reusabilidade). Tais informações concentram-se no nível de interface, e os detalhes de processamento são encapsulados nos objetos (VIEIRA et al., 2006). Na Figura 7, nota-se que os objetos Jovem, Adulto e Idoso, herdam os atributos de Pessoa, essa Pessoa mantém uma associação com a Localização, essa Localização guarda atributos que garantem determinar onde está localizada essa pessoa.

### 3.1.5 MODELO BASEADO EM LÓGICA

Na Ciência da Computação a lógica define expressões e fatos que podem resultar em algum valor direto ou em derivações de outras expressões e fatos (processo conhecido como raciocínio ou inferência). Em um modelo de contexto baseado em lógica, o contexto é definido como fatos, expressões e regras, onde as informações do contexto podem ser adicionadas, atualizadas ou excluídas. O alto grau de formalidade é comum em modelos baseados em lógica.

Uma das primeiras abordagens usando lógica para modelar informações de contexto foi realizada por McCarthy (1993). O contexto foi representado por entidades matemáticas abstratas, onde suas propriedades foram úteis na área de inteligência artificial. Em sua proposta, McCarthy (1993) não tentou definir o que é contexto, mas tentou apresentar uma formalização para representação de axiomas que são usados normalmente, por exemplo, relacionar situações do mundo que são estáticas com situações que mudam. Uma situação importante do modelo é a relação entre a verdade de um determinado contexto com a verdade de outro contexto. A relação básica definida é *ist(c, p)*, isso significa que a proposição *p* é verdadeira no contexto *c*. Nesta abordagem já é suportado o conceito de herança.

A abordagem proposta por Giunchiglia (1993) tem como foco principal os Sistemas Multicontextos, onde o objetivo maior é raciocinar sobre o contexto, ao invés de modelar o contexto. É definido um contexto para representar um subconjunto do estado de uma entidade

individual, isso será utilizado para raciocinar sobre um objetivo, essa representação é vista como uma teoria (parcial) do mundo que codifica perspectiva subjetiva de um indivíduo sobre o assunto.

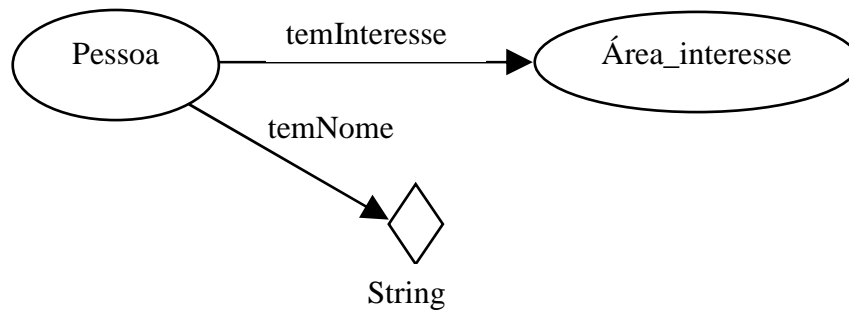
### 3.1.6 MODELO BASEADO EM ONTOLOGIAS

A técnica de modelagem baseada em ontologias surgiu com o intuito de especificar algum conhecimento dentro do contexto da aplicação, representando esse conhecimento de forma abstrata, definindo-o através de regras formais que irão ser executadas somente no momento que forem satisfatórias. A técnica garante a consistência das informações relacionadas ao domínio.

Uma das primeiras abordagens de modelagem de contexto usando ontologias foi proposta por Öztürk et al. (1997), onde foi feita uma análise dos estudos psicológicos sobre a diferença entre recordar e reconhecer das diversas questões em combinação com a informação contextual. Após a análise se entendeu a necessidade de normalizar e combinar os conhecimentos de diferentes domínios. Foi proposto um modelo baseado em ontologias por ter forte normalização e ser bastante formal.

Outra abordagem que se encaixa na categoria de ontologias é o modelo *Aspect-Scale-ContextInformation* (ASC) (STRANG, 2003). Usando ontologias esse modelo fornece uma maneira uniforme para especificar os conceitos do modelo do núcleo e uma quantidade arbitrária de subconceitos e fatos. Esse modelo permite também compartilhar e reutilizar o conhecimento contextual. O conhecimento contextual é avaliado usando raciocínio ontológico. Para implementar o modelo foi usado a linguagem *Context Ontology Language* (CoOL), que é complementado com o uso *Web Services*.

Outra linguagem usada para descrever ontologias é a *Ontology Web Language* (OWL), um exemplo onde foi usada esta linguagem foi o Sistema CoBrA (CHEN et al., 2003), este sistema provê um conjunto de conceitos ontológicos para caracterizar as entidades, essas entidades são pessoas, lugares e outros tipos de objetos interessantes para o sistema. O sistema COBrA usa uma arquitetura central capaz de prover suporte em tempo real a aplicações sensíveis ao contexto.



**Figura 8: Exemplo de representação de ontologia**

Na Figura 8 é apresentado um exemplo de relacionamento entre informações usando ontologias, onde há duas classes relacionadas (*Pessoa/Área\_interesse*) utilizando o relacionamento *temInteresse*. Há também em ontologias a possibilidade de relacionar uma classe diretamente a um dado, como visto em *Pessoa temNome*, o nome será uma *String*, não necessitando assim de outra classe.

## **4 METODOLOGIA**

Para realização deste trabalho, utilizaram-se conceitos de Computação Sensível ao Contexto para desenvolvimento de uma aplicação no cenário do turismo que use informações de contexto para tomada de decisão.

Este capítulo apresenta as fases do desenvolvimento de uma aplicação sensível ao contexto voltada ao turismo. Nas unidades seguintes abordou-se a visão geral da aplicação, um comparativo entre as técnicas de modelagem de contexto, a técnica selecionada e as ferramentas utilizadas para aplicar na prática essa modelagem, por fim, a representação do contexto modelado com a técnica escolhida.

### **4.1 VISÃO GERAL DA APLICAÇÃO**

Utilizando uma aplicação de mapa, um grupo de turistas que visita uma cidade pode pesquisar em um mapa colaborativo por pontos de interesse, como restaurantes, museus, praias, entre outros pontos turísticos. Os membros do grupo podem saber a localização uns dos outros em tempo real, a partir do compartilhamento da localização de cada membro.

Um membro pode informar que somente está interessado em pontos de interesse de determinado tipo ou que estejam em certo raio de distância, ou saber somente a localização de determinados membros do grupo. Cada membro do grupo pode criar um novo ponto de interesse e associar a este ponto um arquivo de mídia, como uma foto ou um vídeo, e decidir se este ponto deve ser compartilhado com o resto do grupo.

Os recursos de Computação Sensível ao Contexto irão decidir quais pontos são interessantes para determinado usuário, a técnica de modelagem deverá suportar as informações referentes ao contexto para ser possível tirar conclusões com os dados que a aplicação gerar.

### **4.2 COMPARATIVO ENTRE AS TÉCNICAS**

A partir da aplicação proposta, é necessário encontrar a técnica que mais se encaixa na modelagem das informações de contexto dessa aplicação. Para isso, foi realizado um comparativo entre as técnicas, levando em consideração os requisitos desejáveis em aplicações sensíveis ao contexto.

Este trabalho levou em consideração os requisitos propostos por Bettini et al. (2010) e por Barth (2004), onde se analisou o cenário do turismo e definiram-se os seguintes requisitos como método para avaliar as técnicas de modelagem e decidir a que melhor se encaixa, são eles: relacionamento e dependência, capacidade de raciocínio, nível de formalidade, heterogeneidade e mobilidade, validação parcial, aplicabilidade.

Para entendimento da Tabela 1, onde foi realizada a comparação entre as técnicas de modelagem seguindo os requisitos definidos anteriormente, assumiu-se a seguinte legenda quanto ao atender ou não um requisito:

- NA – Não Atende. O modelo não possui mecanismos para suportar o requisito.
- AP – Atende Parcialmente. O modelo atende parcialmente, pode ocorrer problema em algum estudo de caso.
- AT – Atende Totalmente. O modelo atende completamente, prove mecanismos para gerenciar o requisito.

<b>TÉCNICA</b> <b>REQUISITO</b>	<b>Chave-Valor</b>	<b>Marcação</b>	<b>Gráfico</b>	<b>O.Objetos</b>	<b>Lógica</b>	<b>Ontologia</b>
<b>Relacionamento</b>	AP	AP	AT	AT	AT	AT
<b>Raciocínio</b>	NA	NA	AP	NA	AT	AT
<b>Formalidade</b>	NA	AP	AP	AP	AT	AT
<b>Heterogeneidade</b>	NA	NA	AT	AP	AP	AT
<b>Val. Parcial</b>	NA	AT	NA	AP	NA	AT
<b>Aplicabilidade</b>	AP	AT	AP	AP	NA	AP

**Tabela 1: Comparativo entre técnicas de modelagem**

#### **4.2.1 TÉCNICA SELECIONADA**

Um cenário de turismo onde as informações são capazes de gerar conhecimento dentro da aplicação requer uma técnica de modelagem de contexto mais robusta, partindo desse princípio a **Técnica baseada em Ontologias** foi utilizada neste trabalho, pois se observou que neste cenário as informações estão em constantes alterações e há necessidade manter relações entre as mesmas.

O desenvolvimento da aplicação necessitou de mecanismos capazes de representar o cenário da aplicação na prática, portanto, a linguagem de descrição de ontologias (OWL) foi necessária. O raciocínio com base no contexto modelado foi possível através da ferramenta

*Semantic Web Rule Language* (SWRL) (HORROCKS et al., 2012) e também, da ferramenta *Protégé*, usada para modelar as informações do contexto.

#### 4.2.2 OWL

A Ontologia gerada neste trabalho foi descrita na linguagem OWL<sup>1</sup>. Essa linguagem foi projetada para aplicações que precisam processar o conteúdo da informação ao invés de apenas apresentá-las ao usuário (MCGUINNESS, HARMELEN, 2004). A Figura 9 destaca a representação da ontologia em OWL.

```
<owl:Class rdf:ID="Esporte">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Area_I"/>
  </rdfs:subClassOf>
</owl:Class>
```

**Figura 9: Exemplo de representação da ontologia em OWL**

A linguagem OWL descreve as informações referentes à ontologia em hierarquias, na Figura 9 é exibido a classe *Esporte* que é uma subclasse de *Area\_I*.

A ontologia desenvolvida em OWL de forma isolada não é uma aplicação funcional, para isso é necessário integrá-la com outra linguagem de programação, por exemplo, Java, que se utilizará de mecanismos para navegar na representação OWL.

#### 4.2.3 SWRL

Baseada em OWL, a linguagem SWRL foi utilizada para possibilitar o raciocínio baseado nas informações de contexto modeladas. Esse mecanismo possui recursos para inferir sobre o contexto, por exemplo, quais usuários tem interesse em esporte.

As regras SWRL são expressas utilizando conceitos de OWL, classes, propriedades e indivíduos. Na Figura 10 há um exemplo de uma regra SWRL.

```
Usuario(?x) ∧ temNome(?x, ?nome) → sqwrl:select(?nome)
```

**Figura 10: Exemplo de regra SWRL**

<sup>1</sup> Recomendação W3C, manual de referência disponível no site <http://www.w3.org/TR/owl-features/>



A regra utilizada na Figura 10 exemplifica o uso dos conceitos OWL, onde *Usuario* é uma classe e *temNome* uma propriedade da ontologia. O símbolo “ $\wedge$ ” representa a operação lógica “e”, “*?x*” e “*?nome*” representam variáveis, por fim, *sqwrl:select()* é uma função para selecionar valores, neste caso os nomes dos usuários.

#### 4.2.4 PROTÉGÉ

*Protégé*<sup>2</sup> é um editor de ontologia e um sistema de aquisição de conhecimento, é livre e de código aberto. É semelhante à ferramenta Eclipse, pois é possível que projetos sugiram *plugins* para que sejam integrados nesta ferramenta.

A ferramenta *Protégé* possui mecanismos para descrever a ontologia através de interface gráfica. A Figura 11 detalha a interface desta ferramenta.

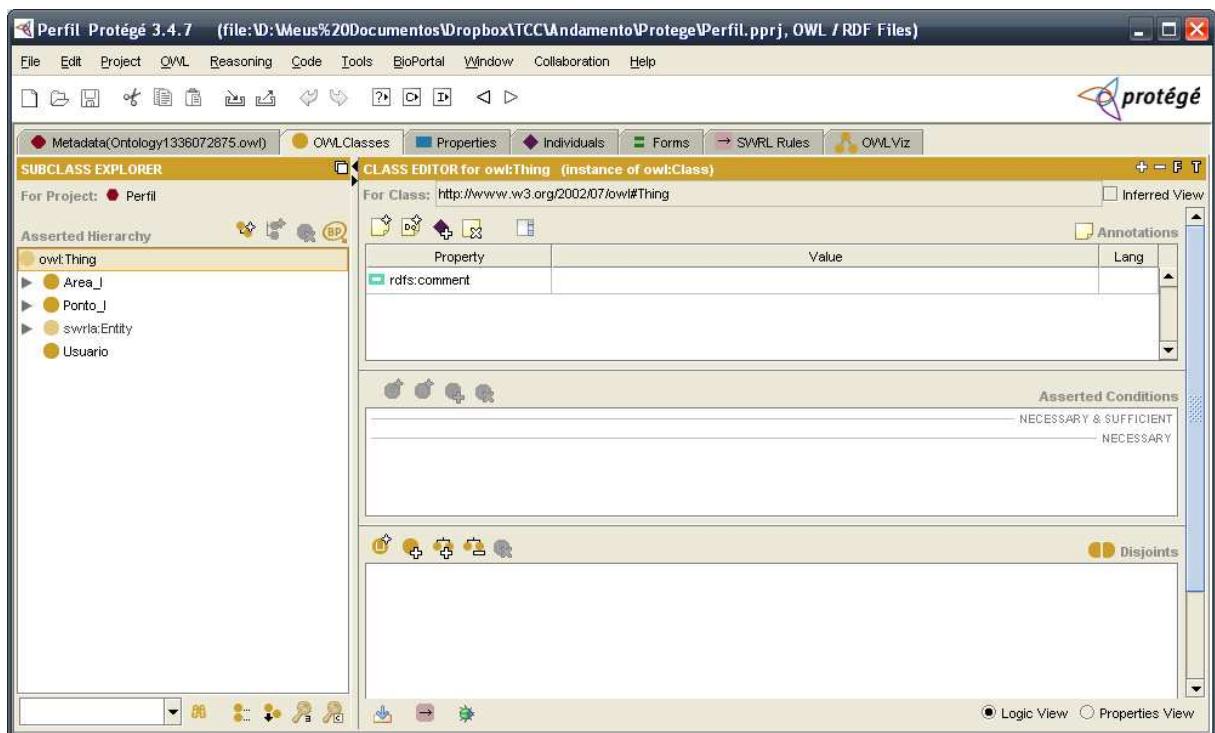


Figura 11: Interface gráfica da Ferramenta *Protégé*

Na Figura 11 a ferramenta está na guia *OWLClasses*, esta é a guia utilizada para desenvolver as classes da ontologia. A guia *Properties* é utilizada para descrever as propriedades da ontologia. A guia *Individuals* serve para gerar as instâncias. Por fim, a guia *SWRL Rules* serve para utilização de regras SWRL. Neste trabalho foi utilizada a ferramenta *Protégé* na versão 3.4.7

<sup>2</sup> Disponível no site <http://protege.stanford.edu/>

### 4.3 MODELAGEM DA APLICAÇÃO

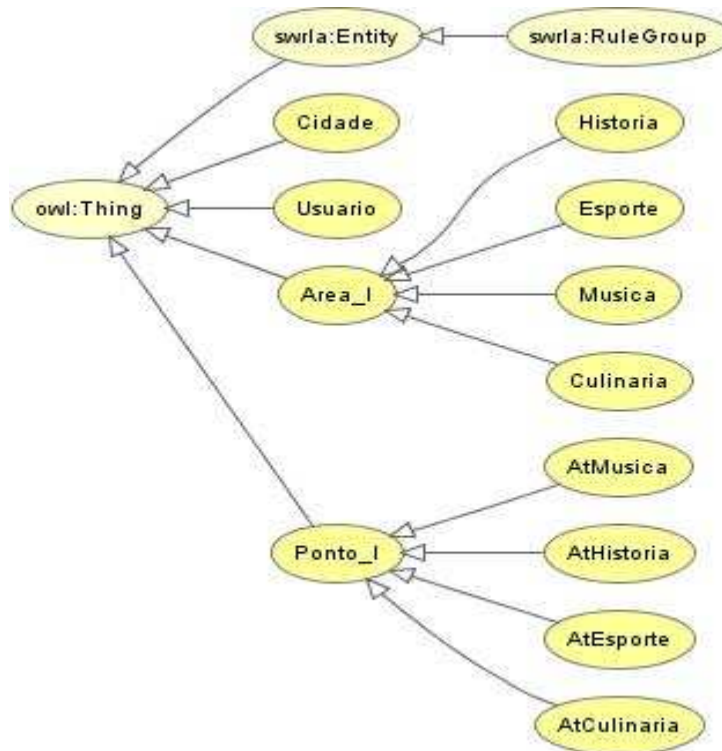
Baseado no cenário do turismo foi necessário propor uma forma de relacionar pontos turísticos aos usuários, por exemplo, se determinado usuário está interessado em eventos esportivos, logo, a aplicação deve ser capaz de classificar os eventos existentes por áreas de interesse e relacionar aos usuários interessados.

A proposta do trabalho foi avaliar as áreas de interesse através de notas. Essas notas são números inteiros que variam de 1 a 5, sendo:

- 1 - o usuário não tem interesse;
- 2 - pouco ou nenhum interesse;
- 3 - interesse mediano;
- 4 - bom interesse;
- 5 - o usuário está interessadíssimo em determinada área.

Por tratar de um cenário de turismo, foram definidas quatro grandes áreas de interesse, sendo elas culinária, esporte, história e música, desta maneira, cada área englobou eventos ou locais que foram cadastrados previamente na inserção do ponto de interesse na aplicação, por exemplo, pontos de interesse classificados como esporte possui subitens como futebol, voleibol, automobilismo, estádios de futebol, entre outros.

O papel da Ontologia foi classificar quais pontos de interesse devem ser relacionados a cada usuário, ou seja, se o usuário se interessar por culinária, a ontologia irá classificar bares e restaurantes como pontos de interesse, bem como, identificar atrações do dia corrente, exibindo somente os locais mais próximos do usuário, e também o horário de realização em caso de eventos. Na Figura 12 é mostrada a Ontologia.



**Figura 12: Visão geral da Ontologia**

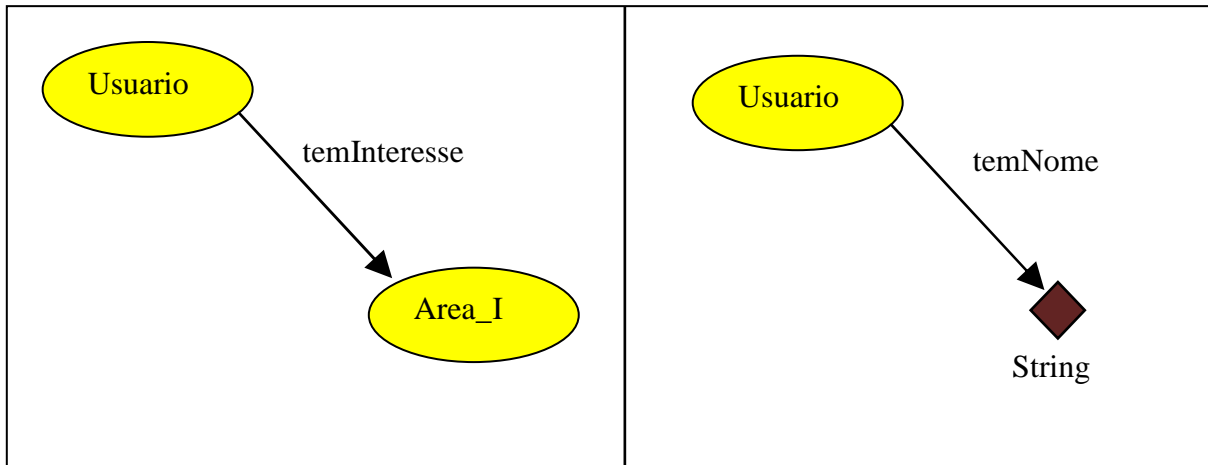
Na Figura 12 cada elipse representa uma classe da ontologia, onde as que possuem cor em destaque são classes desenvolvidas neste trabalho, sendo as restantes classes existentes por padrão na ferramenta e importadas para complementar o trabalho desenvolvido. As setas direcionais representam que a subclasse é uma derivação da classe, por exemplo, Culinária é uma Área de Interesse.

A ontologia é formada por quatro classes principais, a seguir foi detalhada cada uma delas:

- *Area\_I*: possui as áreas de interesse que são representadas pelas subclasses, *Culinaria*, *Historia*, *Esporte* e *Musica*. Esta classe será classificada de acordo com os interesses do usuário.
- *Ponto\_I*: possui as atrações cadastradas na aplicação, sendo categorizadas nas subclasses, *AtCulinaria*, *AtEsporte*, *AtHistoria* e *AtMusica*
- *Usuario*: representa o usuário da aplicação com suas informações.
- *Cidade*: representa a cidade onde são localizados os pontos de interesse.

O relacionamento entre classes foi possível através do uso de propriedades de ontologia. Na ferramenta *Protégé*, essas propriedades podem relacionar classes entre si

(propriedade de objeto) como, *Usuario* tem interesse em *Culinaria*, ou pode relacionar uma classe a um dado diretamente (propriedade de dado), neste caso, *Usuario* tem um nome, nota-se que nome não é uma classe é apenas um dado formado por um conjunto de caracteres. A Figura 13 detalha a diferença entre as propriedades.



**Figura 13: Representação da relação de classes entre si e a relação de classes com dados**

No caso da superclasse estar relacionada com outra classe, suas subclasses herdam esse relacionamento, isto acontece em *Area\_I*, onde o *Usuario* tem interesse em *Area\_I*, logo *Usuario* tem interesse em *Culinaria*, *Esporte*, *Historia* e *Musica*.

As áreas de interesse foram modeladas contendo apenas a informação referente ao valor de interesse, sendo esta uma propriedade de dado denominada *temNota*. Com isto, cada área de interesse possui cinco instâncias (indivíduos) referentes a valores de 1 a 5. Estes valores são relacionados com o usuário para identificar o grau de interesse do mesmo em cada área.

Neste trabalho, as informações do usuário foram modeladas utilizando as propriedades: *temNome*, *temDN*, *temSexo* e *temInteresse*, correspondendo sucessivamente a nome, data de nascimento, sexo e o valor referente ao interesse em cada área, sendo os três primeiros modelados em forma de propriedade de dados e o último como propriedade de objeto.

As informações referentes aos pontos de interesse foram modeladas utilizando as propriedades: *temNome*, *temEndereco*, *temDR* e *temCidade*, correspondendo sucessivamente a nome, endereço, data e hora de realização e a cidade, sendo os três primeiros modelados em forma de propriedade de dados e o último como propriedade de objeto. Os pontos de interesse seguem esse padrão, no caso em que seja necessário diferenciar um evento de um

local, as regras SWRL levam em consideração a propriedade *temDR*, se essa informação tiver valor válido preenchido, é um evento, caso contrário é um local.

## 5 RESULTADOS

Os testes a seguir mostram a capacidade de raciocínio da modelagem baseada em ontologias. Utilizando-se da ferramenta SWRL foi possível definir regras para chegar aos resultados desejados. Um exemplo de busca de informações poderia ser descobrir quais usuários se interessam por esporte.

Para representar os resultados finais deste trabalho foram utilizados dois exemplos, primeiramente para descobrir quais usuários se interessam por cada área e outro para determinar os pontos de interesse exibidos para certo usuário.

### 5.1 CASO 1

O primeiro caso utilizado para obtenção de resultados trata das seguintes questões:

- Quais usuários se interessam por Culinária?
- Quais usuários se interessam por Esporte?
- Quais usuários se interessam por História?
- Quais usuários se interessam por Música?

As instâncias de usuários cadastradas na aplicação são mostradas na Tabela 2, juntamente com suas notas de interesse por área.

Nome	Data de N.	Sexo	Culinária	Esporte	História	Música
Joao	12/11/1988	M	4	5	2	3
Maria	01/08/1983	F	5	2	4	5
Pedro	19/06/1991	M	2	5	1	4
Joana	28/06/1970	F	5	1	4	2

**Tabela 2: Instâncias de usuários.**

Para cada área de interesse foram definidas regras SWRL particulares, com isso foram obtidos resultados separadamente. As regras utilizadas para obtenção dos resultados são as seguintes:

- *Culinária*:  $Culinaria(?c) \wedge temNota(?c, ?nota) \wedge swrlb:greaterThanOrEqualTo(?nota, 3) \wedge Usuario(?x) \wedge temNome(?x, ?nome) \wedge temInteresse(?x, ?c) \rightarrow sqwrl:select(?nome, ?nota);$

- *Esporte*:  $\text{Esporte}(?e) \wedge \text{temNota}(?e, ?nota) \wedge \text{swrlb:greaterThanOrEqual}(?nota, 3) \wedge \text{Usuario}(?x) \wedge \text{temNome}(?x, ?nome) \wedge \text{temInteresse}(?x, ?e) \rightarrow \text{sqwrl:select}(?nome, ?nota)$ ;
- *História*:  $\text{Historia}(?h) \wedge \text{temNota}(?h, ?nota) \wedge \text{swrlb:greaterThanOrEqual}(?nota, 3) \wedge \text{Usuario}(?x) \wedge \text{temNome}(?x, ?nome) \wedge \text{temInteresse}(?x, ?h) \rightarrow \text{sqwrl:select}(?nome, ?nota)$ ;
- *Música*:  $\text{Musica}(?m) \wedge \text{temNota}(?m, ?nota) \wedge \text{swrlb:greaterThanOrEqual}(?nota, 3) \wedge \text{Usuario}(?x) \wedge \text{temNome}(?x, ?nome) \wedge \text{temInteresse}(?x, ?m) \rightarrow \text{sqwrl:select}(?nome, ?nota)$ .

Para exemplificar o funcionamento das regras SWRL, utilizou-se como exemplo a regra que identifica os usuários interessados em culinária. Primeiramente são obtidas todas as instâncias da classe *Culinária* através de *Culinaria(?c)*, depois pega-se as notas referentes a essas instâncias com a regra *temNota(?c, ?nota)*, após utiliza-se a função de SWRL para identificar as notas maiores ou iguais a 3, na regra isso acontece em *swrlb:greaterThanOrEqual(?nota, 3)*. Tendo as instâncias maiores ou iguais a 3, pega-se todas instâncias de usuários através de *Usuario(?x)*, identifica-se os nomes desse usuários com *temNome(?x, ?nome)* e por fim é testado se o usuário tem interesse em determinada área, *Culinária* neste caso, através de *temInteresse(?x, ?c)*, depois usa-se a função para selecionar estes usuários, *sqwrl:select(?nome, ?nota)*.

Os resultados obtidos estão destacados a seguir na Figura 14:

SQWRLQueryTab → Usuario-culinaria	
?nome	?nota
Joao	4
Joana	5
Maria	5

SQWRLQueryTab → Usuario-esporte	
?nome	?nota
Pedro	5
Joao	5

SQWRLQueryTab → Usuario-historia	
?nome	?nota
Joana	4
Maria	4

SQWRLQueryTab → Usuario-musica	
?nome	?nota
Pedro	4
Joao	3
Maria	5

Figura 14: Resultados de quais usuários tem interesse em determinada área

Os resultados obtidos nestes testes foram satisfatórios de acordo com as regras formuladas, notou-se que os resultados foram coerentes com os dados cadastrados previamente na aplicação.

## 5.2 CASO 2

Este caso foi utilizado supondo que cada usuário possui sua aplicação, para este teste foi utilizado o usuário João. Com isso foi tratada a seguinte questão:

- Quais atrações devem ser exibidas para o usuário João?

Para realização dos testes, foram cadastradas duas instâncias para cada área de interesse. Na Tabela 3 são exibidas estas informações. Os valores referentes ao interesse do usuário João por área são os mesmos utilizados no caso 1.

Classe	Atração 1	Atração 2
AtCulinária	Restaurante do Chico	Bar da Praça
AtEsporte	Estádio Olimpico	Futebol: Gremio x Palmeiras
AtHistoria	Museu da Arte	Museu Oswaldo Aranha
AtMusica	Show Banda do Ze	Show de Rock

**Tabela 3: Instâncias de Atrações turísticas**

Levando-se em consideração as áreas de interesse, foi necessário executar quatro regras SWRL, cada regra para identificar as áreas de interesse do usuário João e com isso exibir as atrações que ele tem interesse. As regras utilizadas são as seguintes:

- *Culinária*:  $Culinaria(?c) \wedge temNota(?c, ?nota) \wedge swrlb:greaterThanOrEqualTo(?nota, 3) \wedge temInteresse(Joao, ?c) \wedge AtCulinaria(?ac) \wedge temNome(?ac, ?nome) \wedge temEndereco(?ac, ?end) \wedge temCidade(?ac, ?cid) \rightarrow sqwrl:select(?nome, ?end, ?cid);$
- *Esporte*:  $Esporte(?e) \wedge temNota(?e, ?nota) \wedge swrlb:greaterThanOrEqualTo(?nota, 3) \wedge temInteresse(Joao, ?e) \wedge AtEsporte(?ae) \wedge temNome(?ae, ?nome) \wedge temEndereco(?ae, ?end) \wedge temCidade(?ae, ?cid) \rightarrow sqwrl:select(?nome, ?end, ?cid);$
- *História*:  $Historia(?h) \wedge temNota(?h, ?nota) \wedge swrlb:greaterThanOrEqualTo(?nota, 3) \wedge temInteresse(Joao, ?h) \wedge AtHistoria(?ah) \wedge temNome(?ah, ?nome) \wedge temEndereco(?ah, ?end) \wedge temCidade(?ah, ?cid) \rightarrow sqwrl:select(?nome, ?end, ?cid);$



- *Música*:  $Musica(?m) \wedge temNota(?m, ?nota) \wedge swrlb:greaterThanOrEqual(?nota, 3) \wedge temInteresse(Joao, ?m) \wedge AtMusica(?am) \wedge temNome(?am, ?nome) \wedge temEndereco(?am, ?end) \wedge temCidade(?am, ?cid) \rightarrow sqwrl:select(?nome, ?end, ?cid)$ .

Para detalhar o funcionamento das regras SWRL, é utilizado como exemplo a regra que verifica se as atrações de culinária devem ser exibidas para o usuário *João*. Primeiramente, identificam-se as instâncias de *Culinária*, utilizando *Culinaria(?c)*, depois são obtidas as notas dessas instâncias, através de *temNota(?c, ?nota)*, tendo as notas, verifica-se as maiores ou iguais a 3, através de *swrlb:greaterThanOrEqual(?nota, 3)*, por fim é verificado se o usuário *João* tem interesse em alguma das instâncias com nota maior ou igual a 3, utilizando *temInteresse(Joao, ?c)*, se essa condição for válida, obtém-se as instâncias de atrações culinárias através de *AtCulinaria(?ac)*, depois pega-se o nome e o endereço de tais atrações utilizando *temNome(?ac, ?nome)* e *temEndereco(?ac, ?end)*, por fim, obtém-se a cidade de tal atração através de *temCidade(?ac, ?cid)*. Para exibir os resultados é utilizada a função *sqwrl:select(?nome, ?end, ?cid)*, que exibe as atrações de culinária para o usuário João.

Na Figura 15 são apresentados os resultados obtidos.

The figure displays four windows from a SWRL query interface. The top-left window, titled 'Culinaria-Joao', shows a table with columns '?nome', '?end', and '?cid'. The results are:

?nome	?end	?cid
Bar da Praca	Praca Getulio Vargas	Alegrete
Restaurante do Chico	Rua Uruguai	Alegrete

The top-right window, titled 'Esporte-Joao', shows a table with columns '?nome', '?end', and '?cid'. The results are:

?nome	?end	?cid
Estadio Olimpico	Largo Patrono Fernando ...	Porto_Alegre
Futebol: Gremio x Palmeiras	Avenida Prefeito Joao Vil...	Sao_Paulo

The bottom-left window shows a text area with instructions: 'Select a SQWRL query from the list above and press the 'Run' button. If the selected query generates a result, the result will appear in a new sub tab.' Below this, a red-bordered box contains the error message: 'SQWRL query Historia-Joao did not generate any result.' A 'Run' button is visible at the bottom.

The bottom-right window, titled 'Musica-Joao', shows a table with columns '?nome', '?end', and '?cid'. The results are:

?nome	?end	?cid
Show Banda do Ze	Rua Brasil	Alegrete
Show de Rock	Praca dos Patinhos	Alegrete

Figura 15: Resultados das atrações exibidas para o usuário João

Os resultados obtidos neste teste foram satisfatórios de acordo com as regras definidas. Notou-se que os resultados foram coerentes com os dados cadastrados. No caso do teste para exibir as atrações de História, quando não houve resultado para a consulta, a ferramenta gerou um aviso.

## 6 CONCLUSÃO

A Computação Sensível ao Contexto é uma área de pesquisa recente na Ciência da Computação. Para que as aplicações dessa área sejam desenvolvidas é preciso utilizar de técnicas de modelagem de contexto. Partindo dessa informação foi realizado um trabalho com o intuito de pesquisar e aplicar na prática os conceitos de Computação Sensível ao Contexto, então, foi realizado um estudo detalhado sobre algumas das principais técnicas utilizadas para modelagem de informações de contexto, comparando-as através de um conjunto de requisitos e por fim, aplicou-se uma técnica na prática que detalhou vantagens e desvantagens dessa abordagem.

Durante o estudo foi utilizada a técnica baseada em Ontologias para alcançar os objetivos propostos. Utilizaram-se ferramentas voltadas ao desenvolvimento de Ontologias, OWL, SWRL e *Protégé*, para os testes práticos. Os resultados obtidos foram satisfatórios para o conjunto de testes realizados. A vantagem dessa abordagem foi à possibilidade de raciocinar com base nas informações de contexto, por exemplo, descobrir os interesses de cada usuário, uma aplicação tradicional de forma autônoma, não seria capaz de produzir tais resultados. A desvantagem em utilizar essa técnica é o desempenho da aplicação, supondo uma Ontologia com muitas classes, as regras SWRL provavelmente serão grandes, essa relação pode afetar a velocidade de produção de resultados.

Neste trabalho verificou-se que utilizar perguntas é uma forma de obter as informações de contexto que irão fazer parte da aplicação, pois a tarefa de definir quais as informações de contexto que devem ser modeladas não é trivial. Ao utilizar perguntas, o objetivo principal é chegar às respostas delas, o que implicitamente leva o usuário a limitar o conjunto de informações a serem modeladas por vez.

Portanto, ao término deste trabalho foram identificados detalhes que podem ser abordados no futuro. Um deles é o aprimoramento do mecanismo de avaliação de interesse do usuário por área, nesta abordagem a avaliação é um valor que não será alterado com o tempo, uma proposta é o uso de heurísticas, nesta proposta supõe-se que o usuário comece a frequentar certos lugares que não eram de interesse, baseando-se nessas informações a aplicação alteraria essa avaliação de interesse automaticamente. Outro detalhe é a utilização desta Ontologia em uma aplicação prática, pois neste trabalho foi desenvolvido um protótipo para testes, uma proposta é utilizá-la em uma aplicação Android, onde as informações sobre localização de atrações e usuários poderiam ser obtidas através do uso de Sistema de Posicionamento Global (GPS).

## 7 REFERÊNCIAS

- ASTHANA, Abhaya; CRAVATTS, Mark; KRZYZANOWSKI, Paul. **An indoor wireless system for personalized shopping assistance.** In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pag. 69-74, Santa Cruz, California, 1994.
- BAUER, Joseph. **Identification and Modeling of Contexts for Different Information Scenarios in Air Traffic**, Diplomarbeit. 2003.
- BETTINI, Claudio; BRDICZKA, Oliver; HENRICKSEN, Karen; INDULSKA, Jadwiga; NICKLAS, Daniela; RANGANATHAN, Anand; RIBONI, Daniele. **A survey of context modelling and reasoning techniques.** *Pervasive and Mobile Computing* 6 (2010) 161-180. 2010.
- BULCÃO, Renato. **Um processo de software e um modelo ontológico para apoio ao desenvolvimento de aplicações sensíveis ao contexto.** Instituto de Ciências Matemáticas e de Computação (ICMC). São Paulo. 2006.
- CHEN, Guanling; KOTZ, David. **A Survey of Context-Aware Mobile Computing Research.** Dartmouth Computer Science Technical Report TR2000-381. 2000.
- CHEN, Harry; FININ, Tim; JOSHI, Anupam. **Using OWL in a Pervasive Computing Broker.** In *Proceedings of Workshop on Ontologies in Open Agent Systems (AAMAS 2003)*. 2003.
- CHEVERST, Keith; MITCHELL, Keith; DAVIES, Nigel. **Design of an object model for a context sensitive tourist GUIDE.** *Computers and Graphics* 23, 883–891. 1999.
- DEY, Anind; SALBER, Daniel; ABOWD, Gregory; FUTAKAWA, Masayasu. **The Conference Assistant: Combining context-awareness with wearable Computing.** In *Proceedings of the 3rd IEEE International Symposium on Wearable Computers*, pp. 21-28, San Francisco, California, USA. 1999.
- DEY, Anind. **Understanding and using context.** *Personal and Ubiquitous Computing*. 2001.
- DEY, Anind; SALBER, Daniel; ABOWD, Gregory. **A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications.** *Human-Computer Interaction Journal*. 2001.
- GIUNCHIGLIA, Fausto. **Contextual reasoning.** *Epistemologica - Special Issue on I Linguaggi e le Macchine* 16 (1993), 345–364. Also IRST-Technical Report 9211-20, IRST, Trento, Italy. 1993.
- HEER, Jeffrey; NEWBERGER, Alan; BECKMANN, Chris; HONG, Jason. **Liquid: Context-aware distributed queries.** In *Proceedings of the International Conference Ubiquitous Computing*. Seattle. Washington. USA. 2003.
- HELD, Albert; BUCHHOLZ, Sven; SCHILL, Alexander. **Modeling of context information for pervasive computing applications.** In *Proceedings of SCI 2002/ISAS 2002* (2002).

HENRICKSEN, Karen; INDULSKA, Jadwiga; RAKOTONIRAINY, Andry. **Modeling context information in pervasive computing systems.** In *LNCS 2414: Proceedings of 1st International Conference on Pervasive Computing*, F. Mattern and M. Naghshineh, Eds., Lecture Notes in Computer Science (LNCS), Springer, pp. 167–180. Suíça. 2002.

HORROCKS, Ian; PATEL-SCHNEIDER, Peter F.; BOLEY, Harold; TABET, Said; GROSOFF, Benjamin; DEAN, Mike. **SWRL: A Semantic Web Rule Language.** Disponível no site <http://www.w3.org/Submission/SWRL/#1>.

INDULSKA, Jadwiga; ROBINSON, Ricky; RAKOTONIRAINY, Andry; HENRICKSEN, Karen. **Experiences in using cc/pp in context-aware systems.** *Proceedings of the 4th International Conference on Mobile Data Management (MDM2003)*. M.-S. Chen, P. K. Chrysanthis, M. Sloman, and A. Zaslavsky, Eds., Lecture Notes in Computer Science (LNCS), Springer, pp. 247–261. Melbourne. Australia. 2003.

MCGUINNESS, Deborah L.; HARMELEN, Frank van; **OWL Web Ontology Language.** Disponível no site <http://www.w3.org/TR/owl-features/>.

NEVES, Ana Regia de M.; FONSECA, Humprey Corrêa da; ZOBY, Letícia; RALHA, Célia Ghedini. **Localização de usuários em ambiente interno utilizando abordagem de sistema multiagente.** *iSys - Revista Brasileira de Sistemas de Informação*, 3. ISSN Eletrônico: 1984-2902. 2010.

OLOFSSON, Sarah; CARLSSON, Veronica, SJOLANDER, Jessica. **The friend locator: supporting visitors at large-scale events.** *Personal and Ubiquitous Computing*. 2006.

ÖTZTÜRK, Pinar; AAMODT, Agnar. **Towards a model of context for case-based diagnostic problem solving.** In *Context-97; Proceedings of the interdisciplinary conference on modeling and using context* . pp. 198–208. Rio de Janeiro, 1997.

PASCOE, Jason. **Adding generic contextual capabilities to wearable computers.** In *Proceedings of the International Symposium on Wearable Computers*. Pensilvânia, USA. 1998.

SAMULOWITZ, Michael; MICHAHELLES, Florian; LINNHOFF-POPIEN, Cláudia. **Capeus: An architecture for context-aware selection and execution of services.** In *New developments in distributed applications and interoperable systems*, Kluwer Academic Publishers, pp. 23–39. Polônia. 2001.

SCHILIT, Bill; ADAMS, Norman; WANT, Roy. **Context-aware computing applications.** In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pag 85-90, Santa Cruz, California, 1994.

SCHILIT, Bill; THEIMER, Marvin. **Disseminating active map information to mobile hosts.** *IEEE Network*, 1994.

SCHMIDT, Albrecht; BEIGL, Michael; GELLERSEN, Hans-W. **There is more to context than location.** *Computers and Graphics* 23, 6 (1999), 893–901. 1999.

STRANG, Thomas; LINNHOFF-POPIEN, Claudia; FRANK, Korbinian. **CoOL: A Context Ontology Language to enable Contextual Interoperability**. In *LNCS 2893: Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems (DAIS2003)*, J.-B. Stefani, I. Dameure, and D. Hagimont, Eds., vol. 2893 of *Lecture Notes in Computer Science (LNCS)*, Springer Verlag, pp. 236–247. França. 2003.

VIEIRA, Vaninha; SOUZA, Damires; SALGADO, Ana Carolina; TEDESCO, Patricia. **Uso e Representação de Contexto em Sistemas Computacionais**. In: Cesar A. C. Teixeira; Clever Ricardo G. de Farias; Jair C. Leite; Raquel O. Prates. (Org.). *Tópicos em Sistemas Interativos e Colaborativos*. São Carlos: UFSCAR: 2006. v. 1, p. 127-166.