

**UNIVERSIDADE FEDERAL DE SANTA CATARINA - UFSC**

**CENTRO TECNOLÓGICO – CTC**

**DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA-INE**

**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**UM SISTEMA DE RECOMENDAÇÃO  
SENSÍVEL AO CONTEXTO PARA ATIVIDADES DE  
LAZER**

VITOR PAULON AVANCINI

**FLORIANÓPOLIS**

**2016**

**UM SISTEMA DE RECOMENDAÇÃO SENSÍVEL AO CONTEXTO PARA  
ATIVIDADES DE LAZER**

VITOR PAULON AVANCINI

Trabalho de conclusão de curso apresentado  
como parte dos requisitos para a obtenção do  
grau de Bacharel em Sistemas de Informação  
pela Universidade Federal de Santa Catarina.

Prof. Elder Rizzon Santos

**FLORIANÓPOLIS**

**2016**



## RESUMO

Este trabalho tem como objetivo propor um modelo de recomendação utilizando as abordagens clássicas da teoria de recomendação, como a filtragem colaborativa, adicionando informações contextuais do usuário para não só limitar as recomendações, mas também incorporar no sistema de pontuação estas informações do usuário. Espera-se utilizar informações acerca dos usuários para identificar seus interesses e informações advindas de aparelhos Smartphones para modelar o contexto. Sistemas de recomendação são objetos de estudo desde o início da década de noventa e desde então a área foi amplamente estudada, no entanto pouco foi feito se considerando o *contexto* do usuário e.g. localização, temperatura e clima, horário. Com a popularização de aparelhos *Smartphones*, diversas informações sobre o contexto do usuário podem ser coletadas sem grandes custos. Uma sub-categoria de sistemas recomendadores que poderia se beneficiar do contexto do usuário são sistemas recomendadores de atividades de lazer. É possível encontrar diversos aplicativos de celular que apresentam opções de atividade de diversas maneiras, como dependendo de um estado de humor que usuário escolher ( triste, alegre, empolgado ..) ou de maneira colaborativa utilizando sugestões de outros usuários. No entanto pouco se encontra sobre sistemas que de fato recomendam atividades de lazer viáveis dentro de um conjunto de restrições, como por exemplo, o sistema não recomendaria a prática de esportes como surfe se o sistema fosse executado às nove horas da noite. Ao final deste trabalho pretende-se apresentar um modelo para recomendação e um protótipo que de fato recomende uma atividade ao usuário.

**Palavras-chave:** Sistemas de recomendação sensível ao contexto, context-aware, Atividade, Lazer, Sistemas recomendadores híbridos.

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>6</b>
<b>2 OBJETIVOS</b>	<b>8</b>
2.1 OBJETIVO GERAL	8
2.2 OBJETIVOS ESPECÍFICOS	8
<b>3 SISTEMAS DE RECOMENDAÇÃO</b>	<b>10</b>
3.1 CLASSES DE SISTEMAS RECOMENDADORES	11
3.1.1 Filtragem Colaborativa	11
3.1.2 Demográfico	12
3.1.3 Baseado em Conhecimento	12
3.1.4 Híbrido	13
3.1.5 Baseado em Conteúdo	14
3.2 PROBLEMAS TÍPICOS DE SISTEMAS RECOMENDADORES	16
3.2.1 Problema do Item Novo	16
3.2.2 Problema do Usuário Novo	16
3.2.3 Sparsity	16
3.2.4 Super Especialização	17
3.3 AVALIAÇÃO DOS SISTEMAS RECOMENDADORES	17
<b>4 SISTEMAS DE RECOMENDAÇÃO CONTEXT AWARE</b>	<b>17</b>
4.1 Definição de contexto	18
4.2 Abordagens de Incorporação de contexto	19
4.2.1 Pré-filtragem contextual	19
4.2.2 Generalização de contexto	19
4.2.3 Pós-filtragem Contextual	20
4.2.4 Modelagem Contextual	21
4.3 Métodos de obtenção de contexto	22
4.3.1 Explicitamente	22
4.3.2 Elicitação e estimativa de preferências de contexto	22
4.3.3 Implicitamente	22
4.4 Trabalhos Relacionados	23
4.4.2 CARLO	23
<b>5 WEB SEMÂNTICA E ONTOLOGIAS</b>	<b>25</b>
5.1 WEB SEMÂNTICA	25
5.2 RDF	27
5.3 ONTOLOGIA	28

	5
5.4 OWL	30
5.4.1 Axiomas OWL	31
5.4.2 Axiomas de Classe	31
5.4.3 Axiomas de Propriedade	31
5.5 SPARQL	33
5.6 Motor de Inferência	33
5.7 Punning	34
5.8 DBPedia	34
<b>6 DESENVOLVIMENTO DE UM SR SENSÍVEL AO CONTEXTO COM ONTOLOGIA</b>	<b>35</b>
6.1 Modelo de Recomendação para Atividades	35
6.1.1 Modelagem do Domínio	36
6.1.2 Modelo De Recomendação baseado em conteúdo sensível ao contexto	38
6.2 Implementação do Sistema	41
6.2.1 Ontologia	41
6.2.2 Arquitetura do Sistema	47
6.2.3 Base de Conhecimento	47
6.2.4 Serviço Web	48
6.2.5 Aplicativo Web	55
<b>7 CONCLUSÕES</b>	<b>59</b>
<b>8 REFERÊNCIAS</b>	<b>62</b>
<b>APÊNDICE A - ARTIGO</b>	<b>66</b>
	77

## 1 INTRODUÇÃO

Sistemas recomendadores são ferramentas que, a partir de dados históricos de um de usuário, filtram conteúdo de maneira a apresentar apenas os itens mais interessantes ao usuário (ABBAR; BOUZEGHOUB; LOPEZ, 2009). Segundo Primo (2013), um sistema de recomendação atua basicamente sugerindo itens de forma pró-ativa, visando auxiliar a escolha de itens em sistemas que possuem informações abundantes. Para Park et. al. (2012), são sistemas que utilizam tecnologia analítica para calcular a probabilidade de um produto ser comprado em um local, de forma que o produto correto possa ser recomendado ao cliente.

Apesar de ser uma área relativamente nova, começando a ganhar notoriedade com artigos nas áreas de filtragem coletiva por volta da metade da década de 1990 (ADOMAVICIUS; TUZHILIN, 2005), muitos artigos e estudos foram realizados com o enorme aumento de informação disponível na internet nas últimas décadas. Entre 2000 e 2012 o aumento do acesso à internet foi de 566,4% (PRIMO, 2013).

Esta categoria de sistema ganhou um interesse notável tanto por parte acadêmica quanto pelo mercado. Um fato interessante é que a empresa Netflix organizou uma competição com premiação de um milhão de dólares para um sistema que recomendasse com mais precisão filmes para seus usuários do que o sistema que a empresa já utilizava (ABBAR; BOUZEGHOUB; LOPEZ, 2009).

No entanto pouco foi feito se considerando o contexto do usuário como localização, temperatura e clima, horário. Os sistemas de recomendação tradicionais consideram apenas duas entidades como base para suas predições: itens e Usuários (ADOMAVICIUS; TUZHILIN, 2010). Se por um lado essa abordagem obteve muito sucesso em algumas ocasiões como venda de livros e discos, em outros domínios essa estratégia pode não ser suficiente. Por exemplo, se considerarmos o contexto temporal e de localização ao recomendar um pacote de viagem, provavelmente recomendar uma viagem à Bariloche no verão argentino seria um erro, já que boa parte do turismo em Bariloche está relacionado à neve. Outro exemplo seria um Website com conteúdo apresentado de acordo com o usuário, um usuário pode preferir ver notícias mundiais e sobre mercado de ações durante a semana, e notícias relacionadas a lazer e compras durante o final de semana. Segundo Abbar,

Bouzeghoub e Lopez (2009), um sistema de recomendação eficiente é um sistema que recomende o item certo, na hora e tempo certo, pela mídia certa. Isso seria inalcançável sem informações contextuais.

No domínio de recomendações de atividades de lazer, objeto de estudo deste trabalho, o contexto do usuário é de vital importância. Imagine receber uma recomendação de corrida na rua em um dia de tempestade: a credibilidade do sistema estaria em cheque na primeira utilização.

Com utilização de técnicas da Web Semântica como ontologias é possível modelar esse contexto de forma a associá-lo aos usuários de maneira flexível e ainda se utilizar de motores de inferências e recursos disponíveis na internet. Não somente o contexto físico do usuário foi modelado a partir de técnicas de Web Semântica, todo o domínio do trabalho se beneficiou do uso de ontologias para sua modelagem computacional, como as próprias atividades e o perfil dos usuários.

Aliando as informações contextuais do usuário às técnicas tradicionais de sistemas de recomendação e técnicas da Web Semântica, acreditamos ser possível desenvolver um sistema que recomende atividades de lazer com precisão apreciável e este é o principal objetivo deste trabalho.

No capítulo 2 do trabalho são apresentados os objetivos do trabalho, tanto gerais como específicos. O capítulo 3 discorre sobre a teoria de sistemas de recomendação. No capítulo 4 é feita uma introdução sobre sistemas de recomendação sensíveis ao contexto, categoria de sistemas de recomendação na qual se enquadra o sistema desenvolvido neste trabalho. O capítulo 5 introduz ontologias e Web Semântica, tecnologias utilizadas para modelagem do conhecimento e pontuação de atividades. O capítulo 6 apresenta o desenvolvimento do sistema deste trabalho e no capítulo 7 são apresentadas as considerações finais e conclusões do trabalho.



## 2 OBJETIVOS

### 2.1 OBJETIVO GERAL

Este trabalho tem como objetivo geral o desenvolvimento de um modelo de recomendação que faz recomendações de atividades de lazer a um usuário baseado em seus interesses e seu contexto.

### 2.2 OBJETIVOS ESPECÍFICOS

- a) Compreender o estado da arte de sistemas de recomendação sensível ao contexto.
- b) Especificar os itens que compõem as atividades de lazer e os fatores que as tornam possíveis de serem recomendadas.
- c) Definir quais informações compõem o perfil do usuário e o seu contexto.
- d) Estabelecer um modelo computacional para as atividades, o contexto e o perfil do usuário.
- e) Propor um modelo de pontuação para as atividades usando como fatores influenciadores o contexto e informações do perfil do usuário.
- f) Desenvolver um protótipo (prova de conceito) para *Smartphones* e *browser* que utilize o modelo proposto para recomendar ao usuário uma atividade.

### 3 SISTEMAS DE RECOMENDAÇÃO

Sistemas atuais lidam com quantidades de dados imensas e por conseguinte oferecem conteúdo de forma abundante. Por este motivo, usuários por vezes não conseguem distinguir conteúdos realmente relevantes de outros secundários (ABBAR; BOUZEGHOUB; LOPEZ, 2009).

Sistemas recomendadores são ferramentas que, a partir de dados históricos de um usuário, filtram conteúdo de maneira a apresentar apenas os itens mais interessantes ao usuário (ABBAR; BOUZEGHOUB; LOPEZ, 2009).

Segundo Primo (2013), um sistema de recomendação atua basicamente sugerindo itens de forma pró-ativa, visando auxiliar a escolha de itens em sistemas que possuem informações abundantes. Para Park et. al. (2012), são sistemas que utilizam tecnologia analítica para calcular a probabilidade de um produto ser comprado em um local, de forma que o produto correto possa ser recomendado ao cliente.

De acordo com Primo (2013), os sistemas de recomendação podem ser classificados de cinco formas: baseado em conteúdo, filtragem colaborativa, demográfico, baseado em conhecimento e sistema híbrido. Uma descrição de cada uma dessas abordagens será feita nas próximas seções deste capítulo.

A classificação de um sistema recomendador dentre uma dessas categorias depende exclusivamente de suas fontes de conhecimento (PRIMO, 2013). Essas fontes de conhecimento podem ser adquiridas de diferentes formas, como por exemplo através uma base de produtos a serem recomendados, contendo informações como descrição do produto e funcionalidades. Outra forma de obtenção de conhecimento seria uma base de avaliações, que seriam informações sobre as avaliações de usuários sobre itens previamente recomendados, por exemplo, *Usuário X avaliou filme Y com uma medida Z* (PRIMO, 2013).

### 3.1 CLASSES DE SISTEMAS RECOMENDADORES

#### 3.1.1 Filtragem Colaborativa

Sistemas de recomendação por filtragem colaborativa são amplamente utilizados em sistemas de comércio eletrônico (BURKE, 1999).

Estes sistemas agregam dados sobre hábitos de usuários e fazem recomendações de itens baseado na similaridade entre os padrões dos usuários. Algoritmos de filtragem colaborativa não consideram o conteúdo dos itens a serem recomendados e sim a similaridade entre as pessoas, visando recriar um comportamento natural de recomendação no estilo "boca a boca" que ocorre corriqueiramente entre pessoas que se conhecem bem (PRIMO, 2013).

De maneira um pouco mais formal, a função de utilidade  $u(c,i)$  de um item  $i$  para um usuário  $c$  é calculada baseada nas utilidades já calculadas desse mesmo item  $i$  para um subconjunto  $C'$  do conjunto de todos os usuários  $C$ , onde  $C'$  são usuários de alguma forma similares ao usuário  $c$ , a quem se quer fazer uma recomendação (ADOMAVICIUS; TUZHILIN, 2005). Por exemplo, para se recomendar um filme a um usuário  $c$ , o sistema de recomendação(SR) por filtragem colaborativa primeiramente identifica usuários que são similares ao usuário  $c$ . Nesse caso a medida de similaridade pode ser a comparação entre as avaliações de filmes já avaliados por  $c$  e os demais usuários. Quanto mais parecidas forem as avaliações, mais similares são considerados os usuários. Após isso, o SR recomenda para  $c$  os filmes melhores avaliados por esse subconjunto de usuários similares a  $c$ .

Uma parte considerável dos algoritmos de filtragem colaborativa se utilizam da técnica dos vizinhos mais parecidos (*Nearest Neighbours*) para calcular a função de utilidade (PRIMO, 2013). Três etapas principais são realizadas na execução desse algoritmo para gerar recomendações para um usuário  $u$ : (1) determina-se  $k$  vizinhos para o usuário  $u$ ; (2) implementa-se uma abordagem para agregar os avaliações desses  $k$  vizinhos para um item não avaliado por  $u$ ; e (3) escolhe as  $n$  recomendações melhores avaliadas baseados no passo (2) (BOBADILLA, 2013). O número  $k$  de vizinhos é determinado pelo engenheiro do sistema (PRIMO, 2013).

### **3.1.2 Demográfico**

Assim como o nome sugere, SRs demográficos recomendam itens à usuários de acordo com informações demográficas do usuário. A ideia é que recomendações devem ser diferentes conforme a localização demográfica do usuário (PRIMO, 2013).

A informação demográfica pode ser tão específica quanto alunos em uma determinada sala de aula, ou abrangentes como estudantes brasileiros. Informações demográficas podem ser consideradas informações de contexto, no entanto o contexto costuma ser bem mais abrangente que a posição geográfica de um usuário (PRIMO, 2013).

### **3.1.3 Baseado em Conhecimento**

Sistemas de recomendação baseado em conhecimento tentam sugerir objetos baseados em inferências a respeito de necessidades e preferências desse usuário (BURKE, 2002). De certa forma, todo SR tenta fazer algum tipo de inferência. O que distingue um SR baseado em conhecimento dos demais é que ele tem conhecimento funcional: um SR baseado em conhecimento sabe de que forma um item satisfaz uma necessidade de um usuário e dessa forma consegue raciocinar sobre a relação entre a necessidade do usuário e uma possível recomendação (BURKE, 2002).

Os métodos baseados em conhecimento visam apresentar transparência quanto ao motivo pelo qual o sistema faz determinada recomendação. Uma técnica utilizada para esse fim é Raciocínio Baseado em Casos (PRIMO, 2013), onde a recomendação é realizada através de uma comparação do caso atual de um usuário à casos anteriores, e as recomendações utilizadas nesses casos similares são feitas a este caso novo (PRIMO, 2013).

### 3.1.4 Híbrido

Sistemas de recomendação híbridos combinam duas ou mais técnicas para se obter melhor performance e reduzir efeitos negativos característicos de uma técnica específica (BURKE, 2002).

Segundo Primo (2013), SRs híbridos podem ser alcançados das seguintes formas: Método Balanceado, Método de Permuta, Método Mesclado, Método de combinação de Características, Método em Cascata, Método de Acréscimo de Características e Método em Níveis.

Para o Método Balanceado, as pontuações de algumas técnicas de recomendação são combinadas para produzir apenas uma recomendação (BURKE, 2002). Cada componente presente em um sistema de recomendação híbrido balanceado faz sua recomendação sobre um item de forma independente, tais resultados são combinados linearmente e um resultado final é apresentado (PRIMO, 2013).

No Método de Permuta a hibridização é feita no nível de item. O sistema utiliza um critério para permutar entre técnicas de recomendação e fazer a recomendação final (BURKE, 2002). Por exemplo, em um sistema híbrido conteúdo/colaborativo, primeiro calcula-se a pontuação de um determinado ítem utilizando a técnica baseada em conteúdo, se a pontuação ficar abaixo de um limite, então a técnica de filtragem colaborativa é aplicada e a de melhor pontuação é utilizada.

Já o Método Mesclado apresenta ao usuário recomendações de técnicas variadas sem fazer nenhum julgamento (PRIMO, 2013). Recomendações de diferentes técnicas são apresentadas ao usuário (BURKE, 2002).

Para o Método de Combinação de Características, características de mais de uma técnica são utilizadas no mesmo algoritmo, visando produzir uma única recomendação combinando as propriedades das técnicas envolvidas (PRIMO, 2013). Por exemplo, tratar informação colaborativa simplesmente como propriedades adicional de cada item e aplicar um algoritmo baseado em conteúdo (BURKE, 2002).

O Método de Cascata realiza a hibridização em estágios (BURKE, 2002). Neste caso um algoritmo é aplicado para formar um conjunto preliminar de recomendação e então uma segunda técnica é utilizada para refinar os resultados. No método de cascata existe um fluxo de dados entre técnicas tradicionais, podendo por exemplo existir um resultado de um

algoritmo de filtragem colaborativo servindo de entrada para um algoritmo de filtragem baseada em conteúdo, e por fim apresentado ao usuário (PRIMO, 2013).

O Método de Acréscimo de Características incorpora a recomendação gerada por uma técnica no processo da próxima técnica de recomendação (BURKE, 2002). Por exemplo o sistema *Libra* (MOONEY; ROY, 1999) faz recomendações de livros baseadas em conteúdo, nas informações sobre os livros a serem recomendados, existem os campos “autores relacionados” e “títulos relacionados”, que são resultado de uma filtragem colaborativa aplicada por sistemas recomendadores internos da Amazon (BURKE, 2002).

Por fim, o Método em Níveis combina técnicas de recomendação utilizando o modelo inteiro gerado por uma primeira técnica como entrada para uma segunda técnica (BURKE, 2002). Este método difere do Método de Acréscimo de Características pelo fato de não estar sendo utilizado resultados de um modelo de outra técnica de recomendação como entrada para o segundo passo de recomendação: o modelo inteiro gerado no primeiro passo é entrada para o segundo (BURKE, 2002).

### 3.1.5 Baseado em Conteúdo

A abordagem de sistemas de recomendação baseados em conteúdo tem suas raízes no campo da Recuperação de Informação (*Information Retrieval*) e, fundamentalmente, recomenda itens a um usuário levando em consideração o seu perfil (PRIMO, 2013). Por exemplo, se um usuário de um serviço de reprodução de músicas ouve uma música clássica, outras músicas do gênero serão sugeridas em um momento posterior. Estes sistemas tentam recomendar ao usuário conteúdo relacionado à itens já conhecidos pelo usuário e por isso o feedback do usuário é essencial neste caso (CARRER-NETO et al. 2012)

De maneira mais formal, seja uma função de utilidade  $u$  que representa o quão útil é um item  $i$  a um usuário  $c$ , no contexto de sistemas de recomendação baseados em conteúdo, a função  $u(c,i)$  é estimada baseada nos resultados de  $u(c,I')$ , onde  $I'$  é um subconjunto do conjunto de todos itens  $I$  que são similares ao item a ser recomendado  $i$  e que já foram avaliados de alguma forma pelo usuário  $c$  (ADOMAVICIUS; TUZHILIN, 2005). Por exemplo, em um processo de recomendação de filmes, um usuário  $c'$  deu notas acima de 8 (em uma escala de 0 a 10) para Harry Potter, Titanic e Robocop. O sistema busca filmes

similares a estes por algum critério definido (atores, diretor, gênero) e recomenda somente filmes com um grau de similaridade considerado alto.

Por ser ter suas raízes no campo de Recuperação de Informação, e por este campo ter alcançado avanços significativos em sua área, muitos sistemas recomendadores baseados em conteúdo tem seu foco na recomendação de itens contendo informação textual, como websites e documentos (ADOMAVICIUS; TUZHILIN, 2005). Nestes casos a similaridade entre os itens é calculada a utilizando a medida TF-IDF (Text frequency, Inverse document frequency), técnica consolidada no campo de recuperação de informação (ADOMAVICIUS; TUZHILIN, 2005).

Sistemas recomendadores puramente baseados em conteúdo, por depender quase que somente do histórico e perfil do usuário, tendem a sofrer de um problema conhecido por superespecialização (CARRER-NETO et al., 2012). No entanto estes sistemas tendem a sofrer menos com problemas de *cold start*, problema típico de sistemas de filtragem colaborativa quando ainda não existem recomendações de outros usuários (ADOMAVICIUS; TUZHILIN, 2005). Mais sobre esse problemas será abordado na seção 3.2.4 deste capítulo. Por sofrer menos com problemas de *cold start* e devido ao fato que de não existem usuários do sistema no momento deste trabalho, foi decidido adotar abordagem baseado em conteúdo para desenvolvimento do sistema deste trabalho.

## **3.2 PROBLEMAS TÍPICOS DE SISTEMAS RECOMENDADORES**

### **3.2.1 Problema do Item Novo**

O problema do item novo surge devido ao fato que itens novos não possuem avaliações, logo tem pouca probabilidade de serem recomendados. Como esses itens não são recomendados, passam despercebidos por boa parte dos usuários, e não recebem avaliações, gerando um ciclo vicioso levando esses itens a permanecerem despercebidos (ADOMAVICIUS; TUZHILIN, 2005).

### **3.2.2 Problema do Usuário Novo**

O problema do usuário novo representa uma das grandes dificuldades dos SRs em operação hoje (BOBADILLA, 2013).

Para que um SR possa realmente entender as preferências de um usuário, este precisa ter avaliado um número suficiente de itens. Portanto um usuário novo com poucas avaliações receberá, provavelmente, recomendações sem precisão (ADOMAVICIUS; TUZHILIN, 2005).

### **3.2.3 *Sparsity***

Problema comum em sistemas recomendadores de filtragem coletiva, é caracterizado pelo elevado número de itens possíveis de ser recomendados em relação ao número de usuários do sistema, correndo o risco de a base de dados possuir usuários sem avaliações ou com poucas avaliações dificultando a identificação de usuários similares (PRIMO, 2013).

O sucesso de um SR de filtragem colaborativa depende da disponibilidade de uma massa crítica de usuário. Por exemplo, em um sistema que recomenda filmes, podem haver filmes com somente notas altas porém avaliados por poucos usuários, tornando-os bastante raros de ser recomendados (ADOMAVICIUS; TUZHILIN, 2005). Além disso, para usuários com gostos peculiares, o SR não encontrará muitos usuários com características parecidas levando a más recomendações (ADOMAVICIUS; TUZHILIN, 2005).



### 3.2.4 Super Especialização

O fenômeno da super especialização, típico de sistemas de recomendação baseado em conteúdo, é caracterizado por recomendar apenas itens muito similares à itens bem avaliados por determinado usuário, deixando de recomendar itens relevantes ao usuário. Por exemplo, recomendar apenas filmes de ficção à alguém que assistiu um filme de ficção (BOBADILLA, 2013).

## 3.3 AVALIAÇÃO DOS SISTEMAS RECOMENDADORES

A avaliação dos sistemas de recomendação visa buscar um grupo de usuários para testar o SR realizando tarefas específicas deste sistema. Este grupo deve ser representativo em relação aos reais usuários do sistema para este método ter real validade. Este método pode ser utilizado em conjunto com medidas quantitativas para avaliar o desempenho do sistema (PRIMO, 2013).

O problema relacionado a esta abordagem é o custo envolvido para sua realização, visto que deve se coletar o maior número de informações possíveis com o mínimo de intervenção com o usuário testador (PRIMO, 2013).

## 4 SISTEMAS DE RECOMENDAÇÃO CONTEXT AWARE

Sistemas de recomendação tradicionais costumam levar em conta em seus modelos apenas duas entradas: itens e usuários. As técnicas clássicas de classificação, como filtragem colaborativa e filtragem baseada em conteúdo, tradicionalmente consideram apenas essas duas entradas (ADOMAVICIUS; TUZHILIN, 2010).

Diferente destes sistemas tradicionais que fazem suas recomendações incorporando em seu modelo apenas essa relação *Usuário X Item -> Recomendação*, os sistemas de recomendação sensível ao contexto adicionam ao modelo o contexto do usuário:

*Usuário X Item X Contexto -> Recomendação.*

Este contexto pode ser qualquer informação em tempo real acerca do usuário que possa ser útil para a recomendação, tal como tempo, localização e temperatura (PANIELLO, 2009).

Em Park et. al. (2012), apresenta-se um exemplo em que um cliente recebe recomendações de uma rede de varejo estadunidense, e este cliente as considera ofensiva. O exemplo então mostra que a raiz dessas recomendações era uma compra de um presente para uma outra pessoa. Este mesmo artigo demonstra como a adição do contexto na modelagem do cliente para esse sistema poderia evitar esta situação e também demonstra empiricamente como os resultados deste sistema de recomendação teve sua performance aumentada com a incorporação do contexto ao seu modelo. O argumento apresentado é que se o sistema identificasse o contexto, a compra de um presente, esta compra não seria levada em consideração em novas recomendações.

Outro exemplo é a recomendação de um sistema de viagem que faria recomendações diferentes dependendo se o destino procurado na data em questão estivesse no verão ou inverno, de modo que ,por exemplo um hotel de ski não seria recomendado no verão (RENDLE; FREUDENTHALER; SCHMIDT-THIEME, 2010).

Um sistema de recomendação sensível ao contexto pode ser construído incorporando as informações do contexto de três formas: pré filtragem, modelagem contextual e pós filtragem (ADOMAVICIUS; TUZHILIN, 2010). Neste trabalho optou-se por incorporar o contexto pela abordagem de pós filtragem.

Nas próximas seções deste capítulo serão apresentadas estas três abordagens assim como uma discussão sobre a definição de contexto.

#### **4.1 Definição de contexto**

A definição de contexto pode ser vista de diversas forma e mesmo dentro do domínio específico de SR, algumas definições de contexto distintas já foram apresentadas.

Para Adomavicius e Tuzhilin (2010), dentro do campo de Sistemas de Recomendação, contexto pode ser pensado como a informação adicional que possa ser

relevante para se recomendar itens (RENDLE, FREUDENTHALER, SCHMIDT-THIEME, 2010). Já para (SALMAN et. al 2015) contexto é qualquer informação em tempo real sobre um usuário que ajuda a fazer recomendações melhores, como tempo, localização e temperatura (PANIELLO, 2009).

Para esse trabalho, seguiremos a definição de (SALMAN et. al, 2015).

## **4.2 Abordagens de Incorporação de contexto**

### **4.2.1 Pré-filtragem contextual**

Assim como sugere o nome, a abordagem de pré-filtragem contextual utiliza as informações de contexto para fazer um subconjunto dos dados bidimensionais (*Usuário X item*), eliminando do processo de recomendação os dados que não são relevantes ao contexto da recomendação em questão. Uma grande vantagem desta abordagem é a possibilidade de utilizar qualquer algoritmo tradicional de recomendação (ADOMAVICIUS; TUZHILIN, 2005).

A pré-filtragem contextual pode ser utilizada como uma informação de consulta parecida com consultas de bancos de dados. Por exemplo, para recomendar um filme em um sábado, procurar-se-á apenas por registros que foram ranqueados em sábados para se construir a lista de recomendação. Isso funcionaria como um filtro exato, que introduz um novo desafio: a alta especialização do contexto (RENDLE, FREUDENTHALER, SCHMIDT-THIEME, 2010). Adomavicius e Tuzhilin (2005) introduzem a idéia de generalização de contexto para amenizar essa especialização.

### **4.2.2 Generalização de contexto**

Utilizar o contexto exato de um usuário como filtro ao se fazer uma recomendação pode levar a casos onde o sistema recomendador não tem informação suficiente para fazer uma recomendação, por exemplo, tentar dar uma pontuação ao filme *Gladiador* à um rapaz acompanhado da namorada em um sábado (ADOMAVICIUS; TUZHILIN, 2005). Adomavicius propõe usar ao invés do contexto exato do usuário, um super conjunto desse

contexto. Utilizando o mesmo exemplo, ao invés de usar como entrada para o sistema R (Gladiador, namorada, sábado), utilizar R' (Gladiador, acompanhado, final de semana).

A generalização do contexto pode ser feita de diversas formas. No exemplo acima, seria possível generalizar somente a data, R (Gladiador, namorada, final de semana), ou somente a situação do usuário em questão R (Gladiador, acompanhado, sábado). A generalização do contexto introduz esse desafio: como generalizar o contexto de forma positiva. Existem duas abordagens mais comuns, sendo elas a generalização manual utilizando conhecimento específico sobre o domínio do contexto e a outra uma abordagem automática testando várias formas de generalização e escolhendo a de melhor resultado. Essa abordagem pode elevar consideravelmente a complexidade computacional do sistema em questão (RENDLE, FREUDENTHALER, SCHMIDT-THIEME, 2010).

#### 4.2.3 Pós-filtragem Contextual

Ao contrário da abordagem de pré-filtragem, a pós-filtragem utiliza a informação contextual após um procedimento de recomendação clássico, e.g. filtragem colaborativa (PANIELLO, 2009).

Nesta abordagem, todo o conjunto de dados bidimensional (*Usuário X item*) disponível para o SR é utilizado de maneira tradicional e então alguma técnica de pós-filtragem é utilizada para introduzir a informação de contexto ao sistema (ADOMAVICIUS; TUZHILIN, 2005).

Existem algumas formas de se aplicar a pós-filtragem contextual, sendo dois exemplos a filtragem de fato das recomendações e a atribuição de pesos às recomendações (PANIELLO, 2009).

No caso da filtragem, as recomendações são simplesmente removidas do conjunto de recomendações final, por exemplo, pesquisando lugares para ficar em uma férias de inverno, remove-se todas as recomendações de locais onde a alta temporada não ocorra no inverno. Já no caso da atribuição de pesos, a pontuação das recomendações é recalculada de acordo com as informações contextuais. Citando o mesmo exemplo das férias, a pontuação de locais onde a alta temporada fosse na primavera ou no outono poderiam ser menos influenciados negativamente que a pontuação de locais que tem alta temporada no verão.

A pós-filtragem contextual também se beneficia do fato de qualquer abordagem tradicional de recomendação poder ser utilizada para gerar as recomendações iniciais (ADOMAVICIUS; TUZHILIN, 2005).

#### 4.2.4 Modelagem Contextual

Modelagem contextual incorpora as informações contextuais diretamente ao modelo de predição (ADOMAVICIUS; TUZHILIN, 2005). Neste caso não existe mais uma matriz bi dimensional de usuários e itens, e sim um conjunto de dados multi-dimensional, de forma que as técnicas e algoritmos tradicionais não se aplicam, e outras formas de gerar recomendações se fazem necessárias.

Algumas técnicas utilizadas em sistemas que não levam contexto em consideração podem ser extendidas para esse novo conjunto de dados multi- dimensionais, abordagem de vizinhança (*Nearest Neighbours*), por exemplo a pode ser utilizada se a métrica de similaridade for multi-dimensional (ADOMAVICIUS; TUZHILIN, 2005).

Abordagens específicas para recomendações em sistemas *context-aware* também foram propostas. Oku et al. (2009) propõe adicionar informações contextuais tais como companhia e clima diretamente nas dimensões do problema de recomendação ( *Usuário X Item X Companhia X Clima*) e usar técnicas de *Machine Learning*, como *Support Vector Machine*(SVM) nesse caso, para realizar as recomendações.

### **4.3 Métodos de obtenção de contexto**

#### **4.3.1 Explicitamente**

A obtenção de contexto de forma explícita consiste em adquirir informações contextuais de fontes relevantes de informações, como pessoas, fazendo perguntas diretas ou coletando essas informações por algum outro meio (ADOMAVICIUS; TUZHILIN, 2010).

Por exemplo, um site pode adquirir informações de contexto relevantes pedindo ao usuário preencher um formulário Web antes de conceder acesso a certas páginas.

#### **4.3.2 Elicitação e estimativa de preferências de contexto**

Existe uma terceira forma de obtenção de contexto que se dá através de técnicas de estatística e mineração de dados. Por exemplo, a identidade de uma pessoa em um lar (pai, mãe, filho) não é explícita à uma empresa provedora de TV a cabo, mas essa identidade pode ser inferida utilizando modelos preditivos gerados a partir de técnicas de mineração de dados baseadas nos programas assistidos e nos horários em que esses programas foram assistidos (ADOMAVICIUS; TUZHILIN, 2010).

#### **4.3.3 Implicitamente**

A obtenção de contexto de forma implícita adquire as informações contextuais de forma transparente, como por exemplo informações oferecidas por dispositivos móveis (localização, sensores..). Outro exemplo seria um timestamp representando o momento de uma compra. Na obtenção de contexto de forma implícita não existe necessidade de interação com o usuário ou a fonte da informação, a informação é acessada de forma direta e extraída (ADOMAVICIUS; TUZHILIN, 2010)

Por ser menos invasivo e incômodo ao usuário, o contexto neste trabalho foi obtido de forma implícita utilizando informações coletadas do dispositivo móvel utilizado pelo usuário para acessar o sistema.

## 4.4 Trabalhos Relacionados

### 4.4.1 MOTIVATE

Motivate é um sistema recomendador sensível ao contexto que promove a adoção de um estilo de vida ativo e saudável. Trata-se de uma aplicação para smartphones que fornece recomendações personalizadas e contextualizadas de atividades físicas (LIN et. al., 2011). A aplicação consiste em três componentes: Motivate service, aplicação web Motivate e Motivate API móvel.

O componente que de fato produz a recomendação é o Motivate service. Este componente é constituído dos serviços de localização, agenda, clima, perfil de usuário, tempo e conselho. O serviço de conselho possui um banco de conselhos no formato de regras se-então. O Motivate service recebe uma requisição com a localização do usuário, e com essa localização o serviço de conselhos faz consultas nos outros serviços tentando satisfazer as regras do banco de conselhos. As regras que tiverem todos seus requisitos atendidos são então recomendadas ao usuário. Um exemplo de regra para um conselho de caminhada antes do almoço seria o seguinte: o dia é um dia de semana, o almoço do usuário começa em trinta minutos, a agenda do usuário está livre por uma hora, o clima atual é neutro e existe um local verde a menos de 300 metros.

### 4.4.2 CARLO

CARLO, Model for Context-Aware Recommendation of Learning Objects (Modelo para Recomendação Sensível ao Contexto de Objetos de Aprendizagem) , é um modelo ontológico de contexto que em conjunto com um filtro descrito por regra semântica seleciona objetos de aprendizagem situados próximos ao usuário (MACHADO; PALAZZO, 2014).

O modelo CARLO apresenta quatro dimensões de informação contextual: informações sobre o perfil do usuário, informações sobre localização, informações sobre elementos tecnológicos e informações sobre objetos de aprendizagem (MACHADO; PALAZZO, 2014).

No modelo CARLO, a recomendação é feita através do uso de um motor de inferências que raciocina sobre o modelo ontológico e aplica a regras semântica fazendo uma correspondência entre as características do perfil do usuário, do objeto de aprendizagem e da localização de ambos. De forma genérica, a regra semântica considera um usuário  $u$ , uma característica do perfil desse usuário  $c$  e verifica se o usuário está em uma localização  $l$  e tem em seu perfil a característica  $c$ , após verificar o usuário a regra se preocupa em verificar se um objeto de aprendizagem  $o$  está em uma localização  $x$  (próxima à localização  $l$ ) e se o objeto tem a mesma característica  $c$  que o usuário. Caso todas essas condições sejam verdadeiras, o objeto de aprendizagem  $o$  é então recomendado ao usuário  $u$ .



## 5 WEB SEMÂNTICA E ONTOLOGIAS

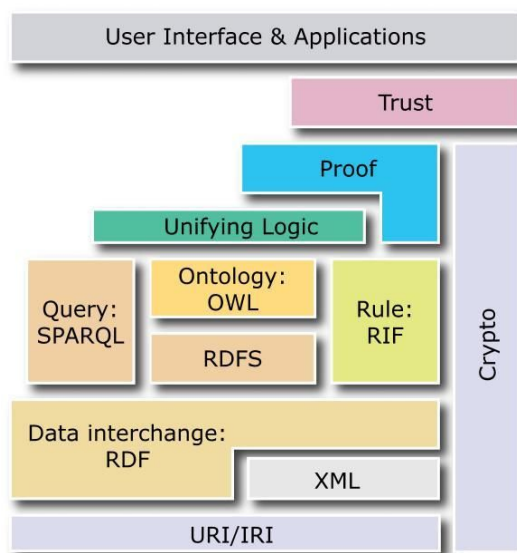
### 5.1 WEB SEMÂNTICA

A Web semântica é uma extensão da Web tradicional na qual a informação recebe um significado bem definido, aumentando a possibilidade de pessoas e máquinas trabalharem em conjunto (BERNERS-LEE; HENDLER; LASSILA, 2001).

Segundo a visão de Tim Berners-Lee, a Web semântica irá trazer estrutura para o conteúdo significativo das páginas Web, criando uma ambiente onde agentes de software caminhando de página em página serão capazes de resolver tarefas sofisticadas para usuários (BERNERS-LEE; HENDLER; LASSILA, 2001).

A Web semântica pode ser melhor visualizada como uma pilha de padrões e tecnologias formando uma arquitetura como a representada na figura 1.

Figura 1- Pilha de tecnologias da Web semântica



Fonte: [https://en.wikipedia.org/wiki/Semantic\\_Web\\_Stack](https://en.wikipedia.org/wiki/Semantic_Web_Stack)

Os componentes previstos na arquitetura da figura 1 são descritos a seguir:

1. **User Interface & Applications:** Representa a aplicação para para o usuário final\*;
2. **Criptografia:** Assegura confiabilidade das fontes da rede semântica por métodos como assinatura digital\*;
3. **Trust:** fornece confiança a sentenças geradas por verificar premissas de fontes confiáveis e por se basear em lógica formal durante derivação de novas informações\*;
4. **Proof:** Os resultados da execução de regras das camadas inferiores são utilizados para provar deduções (PRIMO, 2013);
5. **SPARQL:** Linguagem de consulta em Web semântica feita como padrão pela W3C. Pode ser utilizada para consultas em arquivos de dados baseados no RDF, como RDFS e OWL;
6. **RDF:** Resource Description Framework, é a estrutura para criar sentenças na forma de triplas. Permite a representação de recursos na forma de grafos;
7. **RDFS:** RDF Schema, é o vocabulário básico para a RDF. Permite criar, por exemplo, classes e hierarquias de classes assim como propriedades;
8. **OWL:** *Ontology Web Language*, ou em português, linguagem de ontologia Web. É a linguagem padrão da W3C para a criação de ontologias na Web. É mais expressiva que RDFS por adicionar construções mais sofisticadas para a descrição de semântica de sentenças RDF;
9. **RIF:** *Rule Interchange Format*, é uma recomendação da W3C para padronizar o formato de intercâmbio de regras. Permite descrever relações que não podem ser diretamente descritas utilizando OWL e RDFS;

10. **URI:** Provê meios para identificar de forma única um recurso na rede através de um string padronizado.
11. **XML:** Linguagem de marcação tradicional na internet que permite criação de documentos formados por dados estruturados.

Nas Seções 5.2 e 5.4 os conceitos de RDF e OWL serão explicados com mais detalhes, já que estes serão utilizados para a representação do conhecimento neste trabalho.

Foi escolhido a utilização de Web Semântica devido a possibilidade de utilizar recursos já disponíveis na internet em um formato consumível para máquinas, como por exemplo o DBPedia, que será melhor explicado no final deste capítulo, que fornece dados de páginas da wikipédia em formato de ontologia. Também motivou a escolha da Web Semântica e suas tecnologias a possibilidade de usar ontologias para modelagem do domínio de conhecimento e realizar inferências sobre essa modelagem.

## 5.2 RDF

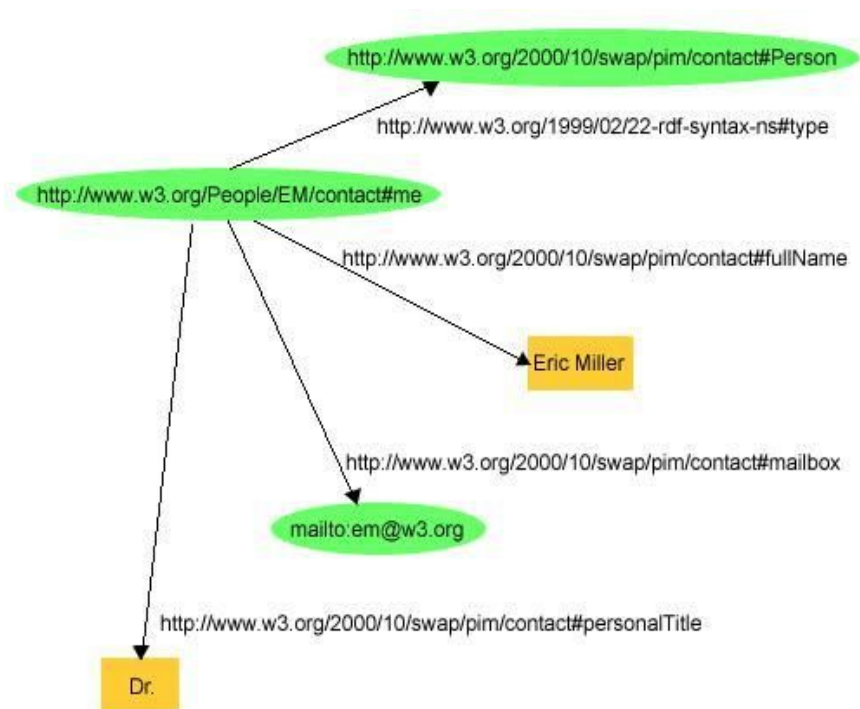
Resource Description Framework provê um modelo de dados simples para gerar sentenças no formato de triplas (sujeito, predicado, valor). Este modelo é modelo padrão estabelecido pela W3C para a troca de documentos na Web semântica (W3C,2016). Junto a esse modelo de dados RDF providencia uma sintaxe em XML que possibilita a serialização desse modelo de dados (VAN OSSENBRUGGEN; HARDMAN; RUTLEDGE, 2006).

O sujeito e o valor de uma tripla RDF podem estar definidos no próprio documento ou podem ser referências a termos em documentos remotos. O predicado pode ser qualquer nome XML com um namespace qualificado (VAN OSSENBRUGGEN; HARDMAN; RUTLEDGE, 2006).

Também é possível gerar sentenças a respeito de conjunto de recursos e a respeito de outras sentenças. O ato de referenciar uma sentença em uma segunda sentença é chamado de retificação.

O framework RDF atribui URIs aos seus campos. A figura 2 mostra no formato de um grafo a representação de uma pessoa chamada Eric Miller. Vale ressaltar que não somente o sujeito e valor possuem URI, mas também o predicado.

Figura 2 -Representação de um documento RDF em formato de grafo



Fonte: WIKIPÉDIA. Disponível em:

<[https://en.wikipedia.org/wiki/Resource\\_Description\\_Framework](https://en.wikipedia.org/wiki/Resource_Description_Framework)> Acesso em: 14 de Jun, 2016.

O exemplo de uma tripla nessa imagem seria (me, fullname, Eric Miller), representados pelos URI *http://www.w3.org/People/EM/contact#me*, *http://www.w3.org/2000/10/swap/pim/contact#fullname* e pelo string Eric Miller. O RDF também usa valores como strings, inteiros, datas e outros (SHADBOLT; HALL, 2006).

### 5.3 ONTOLOGIA

De origem origem muito antiga, dos tempos de Aristóteles e a filosofia antiga, o termo ontologia caracterizava o estudo da essência do ser. Cientistas da computação adotaram este termo e expandiram sua interpretação para "especificação de uma conceitualização" (DING et. al, 2007) e diversas novas definições foram criadas para o termo ontologia.

Segundo Fosket (1997),

“a ontologia é um dispositivo de controle de termos usado na representação de documentos. As ontologias provêm mapas de conhecimento, apresentando conceitos ou ideias do domínio de aplicação e indicando relações entre eles. Estes conceitos podem aparecer representados através de termos, os quais, indicam quando um determinado conceito está sendo tratado”.

Para Guarino e Giaretta (1995), ontologia é conceitualização de um domínio em um formato de possível compreensão para humanos e legível para máquinas, formato esse composto por entidades, atributos, relacionamentos e axiomas.

De forma simplificada, ontologia define um vocabulário comum, interpretável por computadores, que possibilita o compartilhamento de informações sobre um domínio (DING et. al, 2007).

Alguma das motivações para utilizar ontologias são: (1) compartilhar conhecimento comum da estrutura da informação entre pessoas ou computadores; (2) permitir o reuso de conhecimento sobre um domínio; (3) tornar suposições sobre um domínio explícitas; (4) separar o conhecimento do domínio do conhecimento operacional e (5) analisar o conhecimento sobre um domínio (DING et. al, 2007). Noy e McGuiness (2001) propõe uma simples metodologia para a criação de ontologias dividida em sete etapas:

Etapa 1. Determinar o domínio e o escopo da ontologia;

Etapa 2. Considerar a reutilização de ontologias;

Etapa 3. Enumerar termos importantes nessa ontologia;

Etapa 4. Definir as classes e a hierarquia dessas classes;

Etapa 5. Definir as propriedades das classes;

Etapa 6. Definir as características e das propriedades de classes;

Etapa 7. Instanciar as classes.

Esta metodologia, chamada de Ontology Development 101, será seguida neste trabalho e o desenvolvimento de ontologias será feito com o auxílio do software Protégé. software desenvolvido na universidade de Stanford e recomendado em Ding et. al, (2007).

A linguagem de ontologias para Web(OWL), foi utilizada para modelar e instanciar a base de conhecimento utilizada neste trabalho. A utilização de ontologias e

especificamente OWL permite uma modelagem do conhecimento mais flexível e apta a novas mudanças e menos específica a um determinado negócio (HAPPEL, H. J.; SEEDORF, S. 2006). A utilização de ontologias também estimula a reutilização de ontologias já existentes e permite a utilização de motores de inferência que possibilita extração de mais conhecimento dos dados(HAPPEL, H. J.; SEEDORF, S.2006).

#### **5.4 OWL**

A linguagem de ontologia Web (OWL) é o padrão W3C para representar ontologias na Web semântica (CARRER-NETO et al, 2012 ). A principal ideia da OWL é habilitar representações eficientes de ontologias que também são capazes de participar de processos de decisão (SHADBOLT; HALL, 2006). OWL permite a descrever e instanciar ontologias para a Web (SMITH et al, 2003). Na prática, OWL é uma extensão do RDF schema e adiciona mais vocabulários e expressividade ao RDFS (SMITH et al, 2003 ).

Os elementos básicos de uma ontologia OWL são classes, propriedades, instâncias de classes(indivíduos) e relacionamentos entre essas instâncias.

Classes representam um conjunto de recursos com características similares (HEFLIN, J). Toda classe OWL é associada a um grupo de indivíduos, e esse grupo é chamado de extensão da classe. Uma classe é relacionada mas não é igual à sua extensão, permitindo que duas classes possuam extensões iguais(mesmos indivíduos) e mesmo assim não serem classes iguais (SMITH et al, 2003 ).

A OWL prevê dois tipos de propriedades, propriedades de objetos e propriedade de dados. As propriedades de objetos associam dois indivíduos, enquanto propriedades de dados associam indivíduos á valores literais, como números e strings ( SMITH et al, 2003 ).

### 5.4.1 Axiomas OWL

Axiomas são declarações que definem o que é verdade para um determinado domínio (W3C, 2011) e a seguir serão apresentados os axiomas que a linguagem OWL oferece para classes, propriedades de dados e propriedades de objetos.

### 5.4.2 Axiomas de Classe

Classes possuem três principais axiomas: *subclasse*, *classe equivalente* e *classe disjunta*.

O axioma de sub-classe permite dizer que a extensão de uma classe A (indivíduos dessa classe) é um subconjunto da extensão de uma outra classe B (W3C, 2004).

Já o axioma de classe equivalente afirma que uma classe A tem exatamente a mesma extensão de classe que a uma outra classe B. Ou seja, a classe A tem exatamente os mesmos indivíduos que a classe B (W3C, 2004).

O axioma de classe disjunta por sua vez afirma que a extensão de uma classe A não tem nenhum indivíduo em comum com a extensão de uma outra classe B. Ou seja, este axioma afirma que nenhum indivíduo da classe A é também indivíduo da classe B (W3C, 2004).

### 5.4.3 Axiomas de Propriedade

Axiomas de propriedades definem características de uma propriedade. A forma mais simples de um axioma deste tipo apenas definem a existência de uma propriedade (W3C, 2014). Os axiomas de propriedade previstos pela OWL são: *sub-propriedade*, *domínio*, *imagem*, *propriedade equivalente*, *propriedade inversa*, *propriedade funcional*, *propriedade funcional inversa*, *propriedade simétrica* e *propriedade transitiva*.

O axioma de sub-propriedade define que uma propriedade A é sub-propriedade de uma outra propriedade B. Isso implica que se A é sub-propriedade de B e um indivíduo X está ligado a um indivíduo Y por A, então X também está ligado a Y por B.

O axioma de domínio liga uma propriedade a uma classe. Este axioma assegura que o sujeito de uma propriedade deve pertencer a classe definida pelo axioma de domínio (W3C, 2004). Uma propriedade pode ter vários domínios.

Axioma de imagem liga uma propriedade com uma classe ou um tipo de dados como um inteiro ou um string. Este axioma garante que o valor de uma propriedade deve pertencer a classe ou ser do tipo especificado pelo axioma (W3C, 2004). Uma propriedade pode ter diversas imagens.

O axioma de propriedade equivalente é utilizado para afirmar que duas propriedades são equivalentes. Se a propriedade A é equivalente a propriedade B, então A pode ser substituída por B sem nenhuma alteração de significado.

Propriedades tem direção do domínio para a imagem. Pessoas tendem a achar útil expressar relações pelas duas direções, por exemplo carros pertencem a pessoas, e pessoas possuem carros.(W3C,2004). O axioma de propriedade inversa faz exatamente isso, afirma que uma propriedade é inversa da outra, seguindo no mesmo exemplo, *pertencer a* e *possuir* são propriedades inversas.

Uma propriedade funcional é uma propriedade que tem no máximo um valor Y para cada instância X (W3C,2004). Um exemplo de uma propriedade funcional pode ser, por exemplo, *tem cpf*. Uma pessoa pode ter apenas um cpf. O axioma de propriedade funcional serve esse propósito. Garante que uma propriedade tenha apenas um valor para um determinado indivíduo.

Se uma propriedade é definida como inversa funcional, então indivíduo que possuir essa propriedade define unicamente o valor dessa propriedade. De maneira mais formal, se uma propriedade P é inversa funcional, isso garante que um valor Y somente pode ser valor dessa propriedade P para um único indivíduo X (W3C, 2004). O mesmo exemplo de cpf se aplica aqui, se um CPF pertence a um indivíduo X, ele não pode pertencer a nenhum outro indivíduo. O axioma de propriedade inversa funcional garante isso.

Uma propriedade simétrica é uma propriedade que se X é ligado a Y por essa propriedade, Y também é ligado a X por essa propriedade. O axioma de propriedade simétrica garante essa relação. Um exemplo seria uma propriedade *namora com*. Se X *namora com* Y, Y *namora com* X.

Se uma propriedade P é dita transitiva e um indivíduo X é ligado a Y por P, e Y é ligado a Z por P, então X é ligado a Z também por essa propriedade P. Um exemplo seria a propriedade *sub região*. Se X é sub-região de Y, e Y é sub-região de Z, X é sub-região de Z. O axioma de propriedade transitiva garante essa relação.



## 5.5 SPARQL

SPARQL, do acrônimo recursivo SPARQL Protocol and RDF Query Language, é o padrão W3C que traz para a Web Semântica uma linguagem de consulta similar à uma consulta SQL e como em boa parte dos padrões para Web Semântica, é altamente baseado em RDF apesar de também se utilizar de padrões de serviços Web como WSDL(Web Services Description Language) (RAPOZA, 2006).

Consultas SPARQL permitem recuperar e manipular dados no formato RDF( triplas sujeito-predicado-objeto) assim como dados em outros formatos vistos como RDF através de middlewares. A linguagem SPARQL oferece capacidades como busca de padrões obrigatórios e opcionais em consultas, agregações, subconsultas, negações, criação de valores por expressões e teste de valores extensível (W3C, 2013).

Existem diversas implementações de SPARQL sendo algumas delas mais difundidas que outras. Alguns exemplos das implementações mais difundidas são o Apache Jena, OpenLink Virtuoso(utilizado pelo DBPedia) e o Stardog ( utilizado neste trabalho).

## 5.6 Motor de Inferência

O propósito das propriedades OWL é possibilitar inferência. Com toda a informação explicitamente modelada, que informação implícita pode ser inferida? (CRAIG TRIM, 2014). O motor de inferência é um software capaz de gerar conclusões a partir de fatos e axiomas.

No caso do OWL, o motor de inferência, também chamado de raciocinador semântico (semantic reasoner), consegue inferir conclusões a partir das propriedades, classes e indivíduos definidos pela ontologia. Os axiomas definidos pelo OWL auxiliam nesse processo. Por exemplo, se o indivíduo João é da classe Humano, Humano é uma subclasse de Ser Vivo, o motor de inferência pode inferir que João é um ser vivo, mesmo essa informação não sendo explícita.

Alguns exemplos de motores de inferência que implementam os axiomas da OWL são HermiT, Pellet e Racer.

## 5.7 Punning

*Punning* é uma técnica introduzida pela OWL2 onde nomes podem ser utilizados para vários propósitos, por exemplo, o nome Pessoa pode ser ao mesmo tempo nome de uma classe e nome de um indivíduo (GRAU et al. 2006). Em ontologias OWL objetos são identificados por identificadores chamados IRIs, o que é feito com punning é interpretar o objeto baseado em como este é usado contextualmente, um IRI pode ser o mesmo para classe e instância mas pode se referir a um ou outro dependendo do contexto (BERGMAN, 2010)

Esta técnica permite utilizar uma classe como o valor de uma propriedade. Por exemplo podemos dizer que Pessoa e Serviço são classes, e podemos dizer que uma instância s1 de Serviço tem uma propriedade tem Entrada com valor Pessoa. Isso é possível devido ao punning, Serviço nesse exemplo é tanto uma classe quanto um indivíduo, já que propriedades em OWL tem como valor indivíduos, e não classes (W3C, 2012).

## 5.8 DBPedia

O DBpedia é um esforço comunitário para extrair informações estruturadas do Wikipedia e fazer essas informações disponíveis nas internet. DBpedia permite que perguntas sofisticadas sejam feitas ao Wikipedia e permite que diferentes conjuntos de dados na Web sejam conectados a dados do Wikipedia (LEHMAN et. al, 2016).

O DBpedia mapeia as informações contidas nas *infoboxes*(caixas de texto na direita das páginas do Wikipédia) do Wikipedia em uma única e compartilhada ontologia composta por 320 classes e 1650 e propriedades (LEHMAN et. al, 2016). O conjunto de dados do DBpedia inteiro descreve 4.58 milhões de entidades, das quais 4.22 são classificados na ontologia, incluindo 1.44 milhões de pessoas, 735 mil lugares, 123 mil discos de música, 19 mil videogames, 241 mil organizações, 251 mil espécies e 6 mil doenças.

Os dados do DBpedia estão disponíveis no formato RDF e podem ser descarregados no site oficial. Também é oferecido pelo DBpedia um *endpoint* para consultas SPARQL, permitindo consultas online sem necessidade de download dos arquivos de dados disponibilizados pelo DBpedia.

## **6 DESENVOLVIMENTO DE UM SR SENSÍVEL AO CONTEXTO COM ONTOLOGIA**

Dentre os objetivos específicos propostos para este trabalho os seguintes tópicos estão relacionados ao desenvolvimento de um SR:

- g) Estabelecer um modelo computacional para as atividades, o contexto e o perfil do usuário.
- h) Propor um modelo de pontuação para as atividades usando como fatores influenciadores o contexto e informações do perfil do usuário.
- i) Desenvolver um protótipo (prova de conceito) para Smartphones que utiliza o modelo proposto para recomendar ao usuário uma atividade.

Nesta etapa do trabalho esses objetivos foram atacados e esta Seção tem como objetivo descrever e apresentar as ferramentas e tecnologias utilizadas no desenvolvimento do sistema recomendador que abrange esses três objetivos.

### **6.1 Modelo de Recomendação para Atividades**

O sistema recomendador proposto deve recomendar aos usuários atividades que sejam do interesse do usuário. Estes interesses são identificados de duas formas, através de uma coleta explícita por meio de um aplicativo e através de um histórico de atividades já praticadas pelo usuário. Outro ponto que o sistema deve respeitar é que essas atividades devem ser plausíveis de serem praticadas de acordo com o contexto físico do usuário. Por exemplo, este sistema não recomenda uma corrida na rua em um dia de tempestade.

Considerando os requisitos supracitados um modelo ontológico para representação do domínio e um modelo de recomendação baseado em conteúdo foram propostos para desenvolver este sistema.

A abordagem de filtragem colaborativa também foi estudada e considerada para o desenvolvimento deste trabalho. No entanto para um resultado satisfatório utilizando filtragem colaborativo seria necessário um conjunto de dados que apresentasse algum tipo

de relação entre usuários e atividades, já que, como explicado na Seção 3.1.2, um sistema de recomendação de filtragem colaborativa procura usuários semelhantes e recomenda itens bem avaliados por esses usuários similares. Foram avaliadas algumas fontes de dados para aplicar uma abordagem de filtragem colaborativa, como extrair dados da api da rede social de atletas Strava. No entanto esse conjunto de dados continha apenas três tipos de atividades: corrida, bicicleta e natação. Por esse motivo esse conjunto de dados foi descartado e pela falta de um conjunto que atendesse as necessidades deste trabalho a abordagem de filtragem colaborativa foi também descartada.

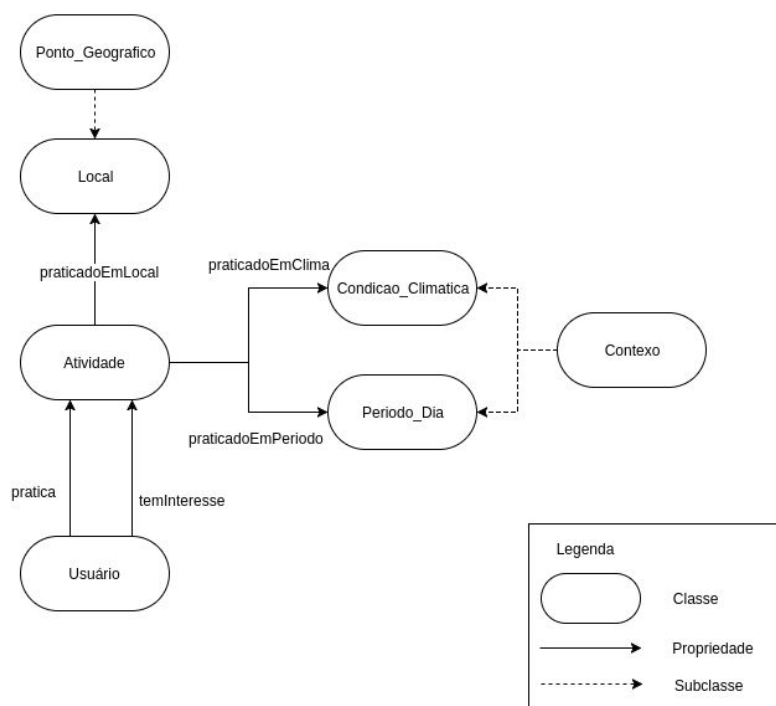
Devido ao requisito de que as atividades recomendadas devem fazer sentido com o contexto do usuário, trata-se também de um sistema de recomendação context-aware. A modelagem do domínio será apresentada na Seção 6.1.2 e uma proposta de pontuação para as atividades será apresentada na Seção 6.1.3. Ao final dessas duas seções os objetivos específicos do trabalho h) e i) devem ser atendidos.

### **6.1.1 Modelagem do Domínio**

Para a modelagem do domínio quatro classes principais foram propostas: Contexto, Atividade, Usuário e Local. A classe Contexto foi dividida em duas subclasses, Condição Climática e Período Dia. A figura 3 apresenta uma visão simplificada de como o domínio foi modelado.

Para as classes Condição Climática e Período do Dia foram aplicadas generalizações de contexto. As condições climáticas foram generalizadas em três possíveis condições: Clima Bom, Clima Ruim e Clima Neutro. Essa generalização introduz algumas limitações. Se considerarmos, por exemplo, a atividade esquiar, que é praticada na neve. O sistema classifica o clima nevando como Clima Ruim, então ou precisamos modelar que esquiar é praticado em clima ruim ou esquiar nunca será recomendado. Para a classe Período do Dia, os horários foram classificados em período matutino, vespertino e noturno.

**Figura 3 - Visão geral da Ontologia proposta**



O perfil do usuário, modelado pela classe Usuário da ontologia proposta, é composto basicamente por duas listas, uma de interesses explicitamente selecionados pelo usuário e uma de atividades praticadas que foram recomendadas pelo sistema. A lista de interesses pode conter qualquer valor dentro da hierarquia das classes Atividade e Local, seja uma subclasse ou um indivíduo (é possível ter subclasses de Local e Atividade como valor de interesse devido à técnica de *punning* explicado na Seção 5.6). Por exemplo, suponhamos que a hierarquia de classes de Atividades consista em duas subclasses, Atividade Física e Atividade Sedentária, e essas subclasses tenham como indivíduos Futebol(física) e Cinema(sedentária).

Um usuário pode ter em sua lista de interesses a classe Atividade Física e o indivíduo Cinema. Já a lista de atividades praticadas serve como apoio para um mecanismo de feedback implícito do usuário, sempre que um usuário aceita uma recomendação de atividade, essa atividade é adicionada à lista de atividades praticadas e será utilizada para pontuar próximas recomendações. O perfil do usuário também conta com propriedades básicas de um usuário como nome, idade e gênero.

Figura 4 - Exemplo de indivíduo da classe Usuario



As atividades, que para este trabalho representam os itens de recomendação, são associadas ao contexto a partir de duas propriedades: `praticadoEmPeriodo` e `praticadoEmClima`. Essas propriedades permitem consultas que filtrem atividades apropriadas para o período do dia e a condição climática de onde o usuário se encontra.

A classe `Local` é subclasse da classe `Ponto_Geográfico`, que exige que todas suas instâncias possuam as propriedades `latitude` e `longitude`. Desta forma, tendo as coordenadas do usuário, é possível saber se os locais nos quais uma atividade é praticada estão próximos ou não.

### 6.1.2 Modelo De Recomendação baseado em conteúdo sensível ao contexto

A recomendação do sistema proposto neste trabalho se dá em duas etapas. Primeiramente é calculado uma pontuação para cada atividade de acordo com o perfil do usuário para quem se está recomendando. O cálculo da pontuação das atividades é explicado a seguir.

Para a lista de interesses explícitos do usuário existem dois tipos de pontuação: direta e indireta. A pontuação direta é caracterizada quando a lista de interesses contém a atividade em questão ou um local onde a atividade em questão é praticada. Por exemplo, se a atividade em questão for `Surfe` e o usuário explicitar que se interessa por `surfe`, esta é uma pontuação direta. A figura 4 demonstra este caso. Se o usuário também explicitasse

interesse em praias, local onde o Surfe é praticado, a atividade receberia mais uma pontuação direta.

A pontuação indireta ocorre quando o usuário possui em seu perfil uma classe que é o tipo da atividade a ser pontuada. Seguindo com o exemplo da atividade Surfe, e supondo que esta atividade é do tipo Atividade Física, se o usuário possui em sua lista de interesse esta classe Atividade Física, Surfe recebe pontuação indireta. A figura 4 também demonstra este caso, e nesta situação Surfe seria pontuado duas vezes, indireta e diretamente. A pontuação direta resulta em um ponto e a pontuação indireta em um terço de ponto. É importante lembrar que a classe Atividade Física só pode estar presente na lista de interesses devido a técnica de punning, esta lista é representada pelas propriedades interest no perfil do usuário e por isso nesse momento Atividade Física é interpretada, na verdade, como um indivíduo.

Já para a lista de atividades que o usuário já praticou, se a atividade em questão está presente na lista, é adicionado meio ponto ao somatório. Voltando para o exemplo da atividade Surfe, como ela recebe pontuação direta e uma pontuação indireta (atividades podem pontuar indiretamente mais de uma vez), sua pontuação é de um  $1 + 1 * \frac{1}{2} = 1,5$  pontos. Atividades com maior pontuação são recomendadas primeiro. Atividades com a mesma pontuação são ordenadas de forma aleatória.

$$S = (Nd) + (Ni * 1/3) + (Np * 1/2)$$

S - pontuação da atividade

Nd - Número de pontuações diretas

Ni - Número de pontuações indiretas

Np - Número de vezes que atividade foi praticada

#### Fórmula 1 - Pontuação de atividade

As multiplicadores de pontuação  $\frac{1}{2}$ ,  $\frac{1}{3}$  e 1 foram definidos de forma arbitrária. O que eles representam é que uma pontuação direta vale três vezes o que vale uma pontuação indireta e duas vezes o que vale uma pontuação por já ter praticado uma atividade.

A segunda etapa da recomendação é a incorporação do contexto à recomendação. Um requisito definido para o sistema anteriormente é a compatibilidade do contexto do usuário com as características da atividade a ser recomendada. Para manter esta compatibilidade, os requisitos de contexto para uma atividade foram definidos da seguinte maneira:

- O valor da propriedade `praticadoEmPeriodo` deve ser exatamente o período do dia atual do usuário;
- O valor da propriedade `praticadoEmClima` deve ser exatamente a condição climática de onde o usuário se encontra;
- Deve existir pelo menos um indivíduo da classe `Local` que é valor da propriedade `praticadoEmLocal`, e esse indivíduo deve estar situado dentro de um raio de  $X$  quilômetros da localização do usuário.

Um possível exemplo seria a atividade Remo. Para este exemplo, digamos que esta atividade é praticada em lagoas, apenas no período matutino e em dias de sol. Para um usuário receber uma recomendação de praticar remo, este usuário precisa estar a uma distância de até  $X$  quilômetros de alguma lagoa, no período da manhã e em um local com clima ensolarado. A escolha de  $X$  vai impactar fortemente na quantidade de recomendações retornadas, para um  $X$  pequeno apenas atividades praticadas em locais muito perto do usuário podem ser recomendadas e é possível que não exista nenhuma para  $X$  demasiadamente pequenos. Por outro lado  $X$  muito grandes podem retornar atividades que não façam sentido para o contexto do usuário. Existe uma troca entre qualidade e quantidade na escolha do tamanho de  $X$ .

As atividades que não satisfazem esses requisitos de contexto são removidas do conjunto de atividades a ser recomendado. Essa abordagem de incorporação de contexto se enquadra na definição de pós-filtragem citada na Seção 4.2.3 deste trabalho.



## 6.2 Implementação do Sistema

Para o desenvolvimento deste sistema as seguintes etapas foram realizadas:

- Desenvolvimento de uma ontologia que modele o usuário, o seu contexto, as atividades e os relacionamentos entre essas entidades.
- Desenvolvimento de um banco de conhecimento para armazenar de forma eficiente e a possibilitar consultas nesta ontologia desenvolvida.
- Desenvolvimento de um serviço Web que retorne recomendações geradas a partir de consultas no banco de conhecimento
- Desenvolvimento de um aplicativo Web que consome o serviço Web supracitado

As próximas seções deste capítulo detalham como cada uma dessas etapas foram realizadas bem como as ferramentas, tecnologias e metodologias utilizadas.

### 6.2.1 Ontologia

A ontologia desenvolvida neste trabalho seguiu como base para sua construção metodologia Ontology Development 101 (NOY; MCGUINNESS, 2001). A metodologia prevê sete passos e esses passos serão descritos a seguir.

O primeiro passo do método tem como objetivo determinar o domínio e o escopo da ontologia. O autor sugere perguntas que ele denomina como perguntas de competência, que são perguntas para verificar se a ontologia possui detalhes o suficiente para responder a essas perguntas (NOY; MCGUINNESS, 2001). Algumas das perguntas que surgiram para o desenvolvimento são:

-Que atividade posso praticar em são paulo de manhã em um dia de sol?

-Que atividade uma pessoa com mais de 60 anos pode praticar em um dia chuvoso em Florianópolis?

-Que atividade sugerir à uma pessoa que está em Criciúma e se interessa por escaladas indoor?

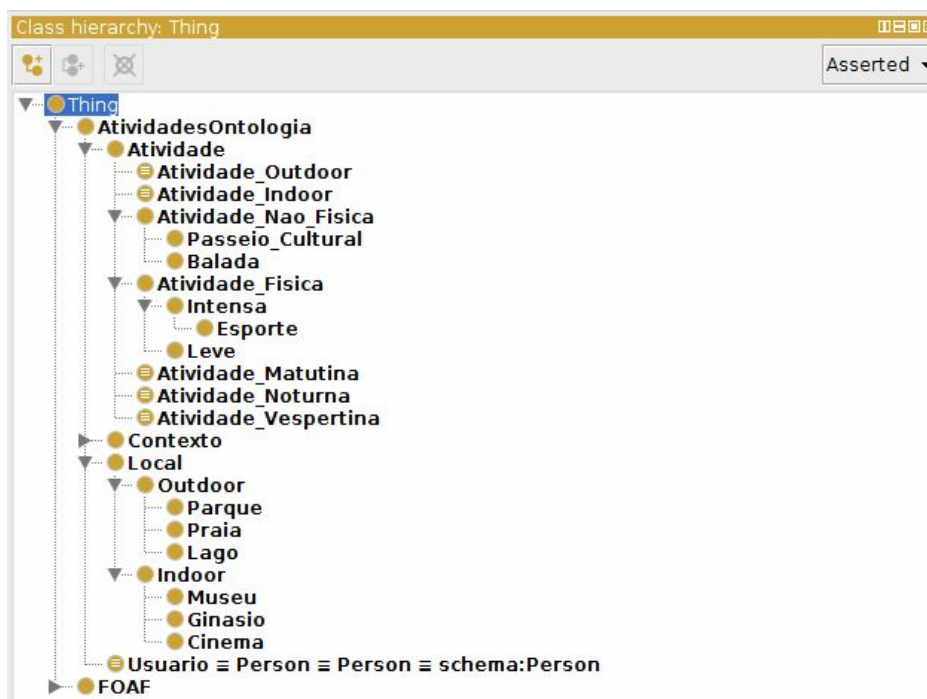
Permitir a reutilização de conhecimento sobre um domínio é um dos cinco motivos citado por Ding et. al. (2007) para se utilizar ontologias e é exatamente isso que a segunda

etapa da metodologia visa alcançar. No contexto deste trabalho apenas uma ontologia externa será utilizada, a ontologia Friend of a Friend (FOAF). A idéia básica desta ontologia é que se pessoas publicam documentos no formato FOAF, máquinas são capazes de interpretar essas informações (BRICKLEY; MILLER, 2014). Esta ontologia foi utilizada para representar os usuários e seus interesses através das propriedades e classes definidas em seu vocabulário.

O terceiro passo da metodologia Ontology Development 101 visa enumerar termos importantes na ontologia. Alguns dos termos levantados nessa etapa foram *atividade*, *atividade física*, *outdoor*, *indoor*, *usuário*, *local*(tipo de local em que se pratica uma atividade), *clima*, *período do dia*, *contexto*.

Na quarta etapa o objetivo é definir classes e a hierarquia das classes. A imagem z apresenta as classes e a hierarquia utilizada neste trabalho.

Figura 5 - Hierarquia de classes da ontologia desenvolvida.

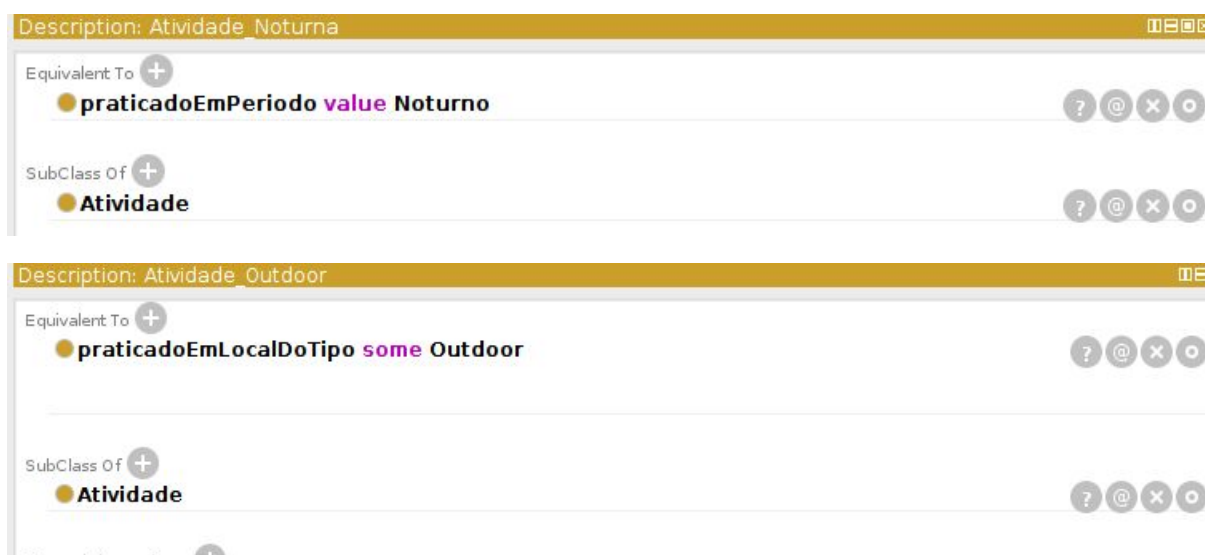


A imagem acima demonstra a materialização do modelo do domínio deste trabalho proposto na Seção 6.1.1. Foram criadas as quatro classes principais Contexto, Local, Atividade e Usuário bem como uma hierarquia de subclasses para essas classes.

As classes *Atividade\_Matutina*, *Atividade\_Noturna*, *Atividade\_Vespertina*, *Atividade\_Outdoor* e *Atividade\_Indoor* foram definidas através de regras de equivalência.

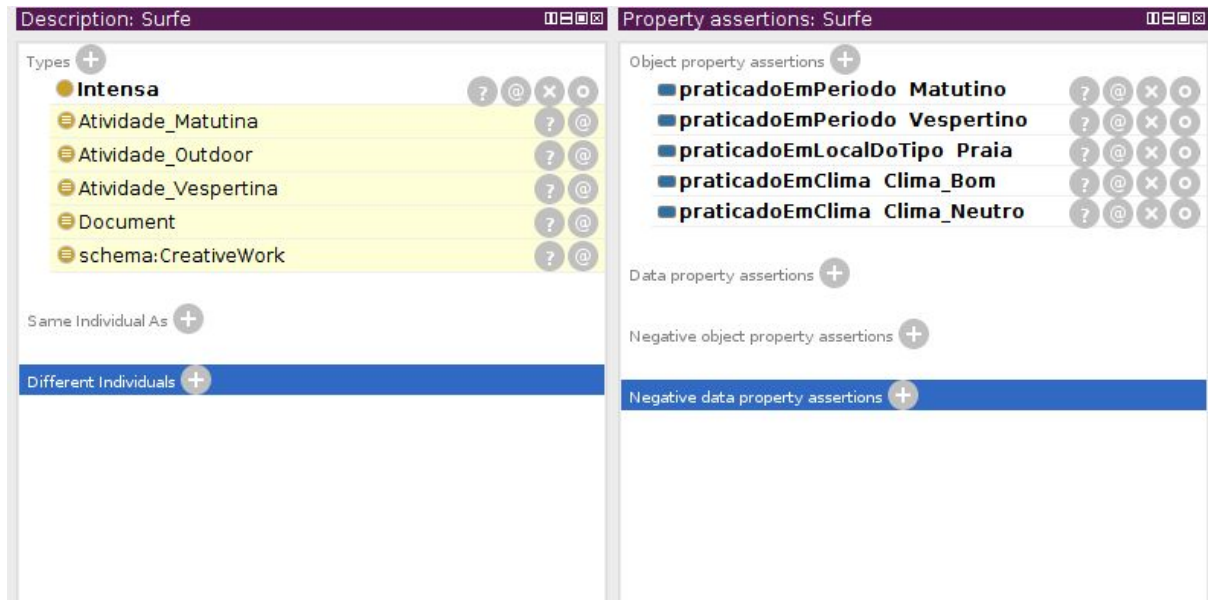
Por exemplo a classe *Atividade\_Vespertina*, uma atividade é considerada vespertina caso possua a propriedade *praticadoEmPeriodo* com valor *Vespertino*. A mesma lógica se aplica para *Atividade\_Matutina* e *Atividade\_Noturna*. Já para as classes *Atividade\_Indoor* e *Atividade\_Outdoor* a regra utilizada definida foi possuir propriedade *praticadoEmLocal* com valores pertencentes às classes *Indoor* e *Outdoor*. Um exemplo para uma instância considerada *Atividade\_Outdoor* é a atividade *Vela*. *Vela* tem o valor *Lago* para a propriedade *praticadoEmLocal*, e *Lago* é um indivíduo da classe *Outdoor*.

Figura 6 - Restrições das classes *Atividade\_Noturna* e *Atividade\_Outdoor*



As atividades, objeto de recomendação do sistema aqui proposto, são indivíduos da classe *Atividade* ou de suas subclasses. Na figura abaixo é apresentado um exemplo de uma instância da classe *Atividade\_Intensa*, subclasse de *Atividade*. Na imagem também é possível ver exemplos de algumas das classes definidas por regras citadas anteriormente para a atividade *Surfe*, as linhas na coluna da esquerda (Types) com fundo amarelo são classes inferidas, não foram explicitamente associadas à atividade.

Figura 7 - Exemplo de uma atividade dentro da ontologia



Indivíduos da classe Atividade possuem obrigatoriamente três propriedades: *praticadoEmPeriodo*, *praticadoEmLocal* e *praticadoEmClima*. Essas propriedades são utilizadas para executar a filtragem contextual como explicado na Seção 6.1.2

Usuários do sistema são representados como indivíduos da classe *Usuario*. Esses indivíduos, como explicado na Seção 6.1.1, possuem duas listas de atividades, sendo uma dessas listas a lista de interesses, e a outra uma lista de atividades praticadas que foram recomendadas pelo sistema. Essas listas são representadas pelas propriedades *interest* e *praticou*. Por exemplo o usuário representado na figura 8, sua lista de interesses é composta por Praia e Museu, e sua lista de atividades praticadas é composta pela atividade Vela.

Figura 8 - Exemplo de usuário do sistema



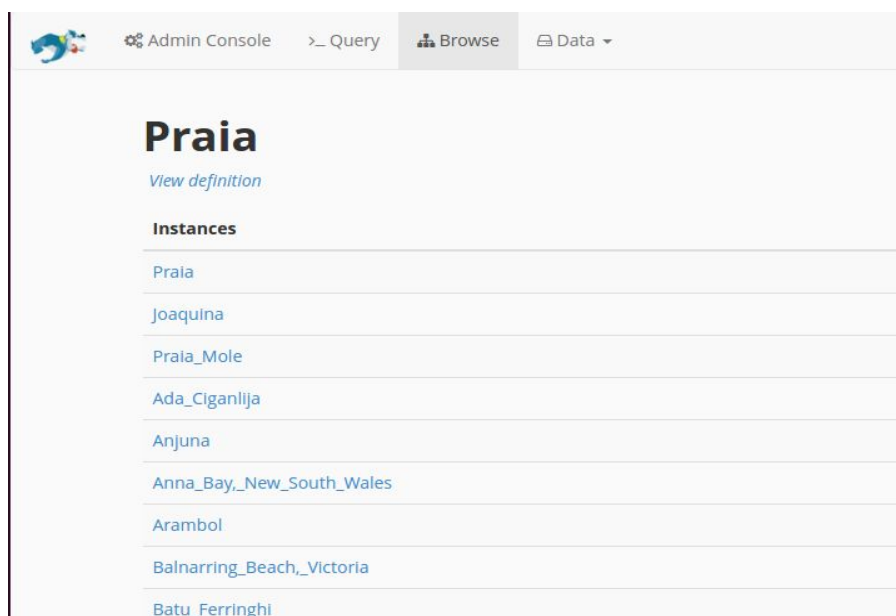
No quinto passo da metodologia Ontology Development 101 são definidas as propriedades das classes. A classe Usuário foi definida como equivalente a classe FOAF Person e por essa razão tem todas as propriedades que a classe Person. A classe Person define diversas propriedades de uma pessoas tais como nome, sobrenome e idade. Para este trabalho também foi utilizada a propriedade *interest* definida pela ontologia FOAF para a classe Person, para representar que atividades o usuário se interessa. A classe Local tem como sua única propriedade definida o seu ponto geográfico(suas coordenadas). A classe Atividade, como citado anteriormente, possui três principais propriedades: *praticadoEmPeriodo*, *praticadoEmClima* e *praticadoEmLocal*. Essas propriedades relacionam a atividade com o contexto do usuário de forma a fazer a filtragem contextual dessas atividades.

O objetivo do sexto passo é definir as características do valor das propriedades. Nesta etapa foi definido que o valor da propriedade *praticadoEmPeriodo* deve ser uma instância da classe *Periodo\_Dia*, que o valor da propriedade *praticadoEmClima* deve ser uma instância da classe *Condição Climática* e que o valor da propriedade *praticadoEmLocalDoTipo* deve ser uma instância da classe *Local*. A propriedade *interest* da ontologia FOAF tinha como seu objeto instâncias da classe *Document* também definida por FOAF. Para integrar *interest* à ontologia desenvolvida neste trabalho foi adicionada a classe *Atividade* aos possíveis valores da propriedade.

No sétimo e último passo da metodologia instâncias são criadas. Para a classe de usuário não foram criadas instâncias manualmente, elas são criadas quando um usuário se registra pela aplicação Web. Para a classe Atividade todas as instâncias foram criadas manualmente com a ferramenta Protégé. Exemplos de instâncias para essa classe são Surfe, Vela, Corrida e Trilha. Para as classes de contexto Período\_Dia e Condição\_Climatica as instâncias foram criadas manualmente também com o auxílio da ferramenta Protégé. Exemplos de instância são Noturno e Diurno para Período\_Dia e Clima\_Ruim e Clima\_Bom para Condição\_Climatica.

A classe Local teve um processo de criação de instâncias diferente. Parte das instâncias foram criadas manualmente assim como as outras classes. No entanto parte foi importado da DBPedia. Por exemplo a classe Praia, subclasse de Local. As instâncias de Praia são na verdade instâncias do tipo yago:Beach extraídas de DBPedia. O mesmo acontece para Lago e Museu, que são instâncias das classes dbo:Museum e yago:Lake respectivamente.

Figura 9 - Instâncias de Praia no triplestore Stardog



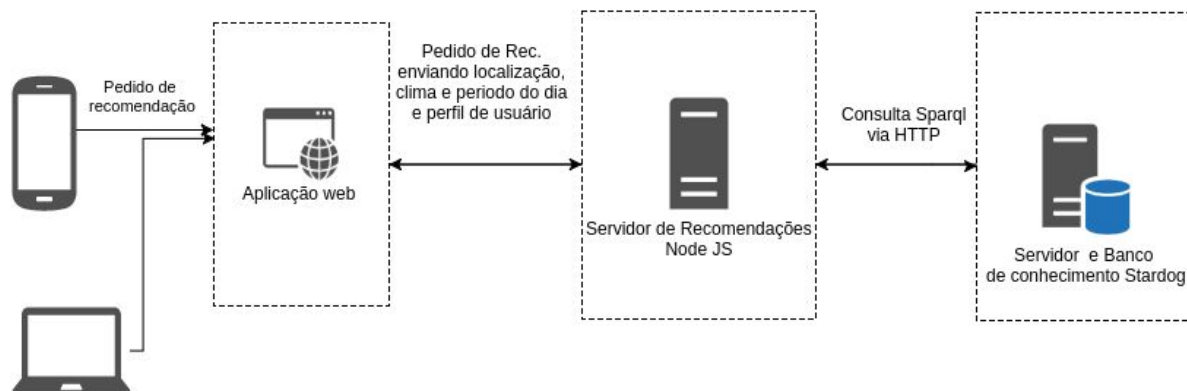
Consultas SPARQL permitem executar partes das consultas em endpoints diferentes e seria possível realizar parte da consulta no DBPedia sem necessidade de importar essas instâncias para a ontologia local, no entanto não é possível se utilizar de motores de

inferências em consultas com endpoints remoto(COPPENS et al, 2013). Por essa limitação foi decidido importar as instâncias localmente.

### 6.2.2 Arquitetura do Sistema

O sistema proposto por este trabalho é composto por três unidades: uma aplicação Web responsiva para fácil uso em smartphones, uma api Web escrita em Node JS que retorna recomendações a partir do terceiro componente do sistema, um banco de conhecimento. Este banco de conhecimento foi construído com a ferramenta Stardog, que além de armazenar as triplas, possui um motor de inferência e expõe uma interface HTTP para consulta.

Figura 10 - Arquitetura do sistema recomendador



### 6.2.3 Base de Conhecimento

Para armazenar e realizar consultas na ontologia desenvolvida neste trabalho foi necessário se utilizar de um banco que pudesse armazenar triplas RDF, também conhecidos como Triplestore. Além disso este banco deveria ser capaz expor uma interface para consultas SPARQL e ter suporte a motores de inferência.

Da documentação oficial, Stardog 4.2 suporta o modelo de dados de grafo RDF, a linguagem de consulta SPARQL, protocolo HTTP para consulta remota, OWL 2 e regras definidas pelo usuário para inferência e interface de programação para diversas linguagens. Por esses motivos o Stardog versão 4.2 foi escolhido como a ferramenta para armazenar as triplas RDF. Outra opção que atendia aos requisitos citados é o banco

Virtuoso, no entanto sua documentação não era clara e sua instalação parecia bastante mais complexa que a opção escolhida.

Dentro da arquitetura do sistema recomendador implementado neste trabalho, o banco de conhecimento Stardog roda em um servidor diferente da aplicação web e do servidor NodeJS. O banco é acessado através da API HTTP exposta pelo próprio Stardog. Neste endpoint é possível realizar diversas operações, tais como consultas SPARQL, inserções de dados, operações de manutenção do banco entre outras operações.

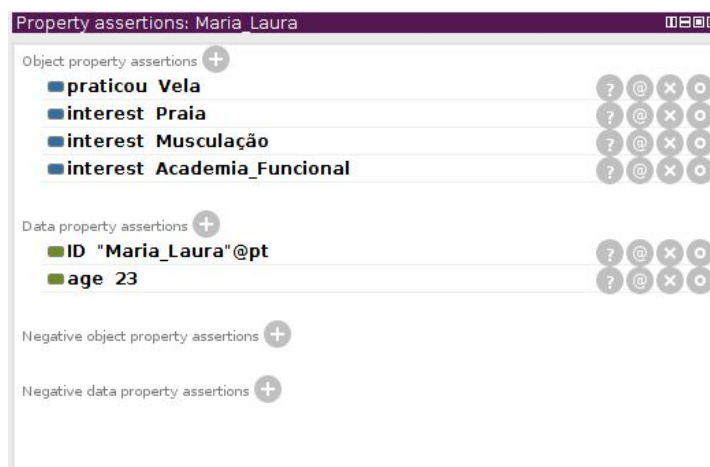
Os dados dessa base de conhecimento foram inseridos através de ferramenta também nativa do Stardog. Esta ferramenta permite inserir por linha comando dados em arquivos de diversos formatos. No caso deste trabalho, a ontologia desenvolvida na ferramenta Protégé foi exportada no formato .ttl e inserido no banco pelo cliente de linha de comando oferecido pelo Stardog.

#### 6.2.4 Serviço Web

O serviço Web é onde a recomendação é de fato gerada. É neste componente do sistema que as atividades têm sua pontuação calculada para um usuário e seu contexto. O serviço expõe um endpoint que recebe da aplicação Web uma requisição HTTP com um identificador do usuário, suas coordenadas geográficas, e o horário local de onde o usuário fez a chamada. Com essas informações o serviço Web calcula a pontuação para as atividades e remove da lista atividades que não satisfazem os requisitos de contexto como explicado anteriormente.

A pontuação das atividades é realizada em algumas etapas e estas serão descritas e exemplificadas a seguir para o usuário Maria Laura, demonstrado na figura 11.

Figura 11 - Definição do usuário Maria Laura





O primeiro passo é buscar para o usuário em questão a lista de interesses explícitos. Caso a atividade sendo pontuada ou o local de sua prática esteja nessa lista, uma pontuação direta é adicionada a conta. A figura abaixo apresenta a query que busca estes interesses e o resultado para o usuário em questão.

Figura 12 - Consulta dos interesses do usuário

```

1  SELECT  DISTINCT ?usuario ?interesse
2
3  WHERE {
4
5      {
6          ?usuario a :Usuario.
7          ?usuario :ID "Maria_Laura"@pt.
8          ?usuario foaf:interest ?interesse.
9          ?interesse rdf:type ?tipo.
10         ?tipo rdfs:subClassOf* :Atividade.
11     } UNION
12     {
13         ?usuario a :Usuario.
14         ?usuario :ID "Maria_Laura"@pt.
15         ?usuario foaf:interest ?interesse.
16         ?interesse rdf:type* ?tipo.
17         ?tipo rdfs:subClassOf* :Local.
18     }
19 }
20 }
21

```

Results

SPARQL Results	
usuario	interesse
 :Maria_Laura	 :Academia_Funcional
 :Maria_Laura	 :Musculação
 :Maria_Laura	 :Praia

Como explicado na Seção 6.1.3, são passíveis de pontuação direta os indivíduos das classes da hierarquia de atividades e as próprias classes da hierarquia de locais. Essa consulta busca exatamente isso, indivíduos de atividades (Academia\_Funcional e

Musculação) e classes de locais (Praia) onde atividades são praticadas. O resultado dessa consulta popula uma lista de pontuadores diretos.

O segundo passo do cálculo de pontuação é procurar as classes das atividades que o usuário se interessa, esta parte é realizada com outra consulta e a mesma é demonstrada abaixo. Para cada classe da atividade sendo pontuada presente no retorno desta consulta, uma pontuação indireta é adicionada à conta. O resultado dessa consulta popula uma lista de pontuadores indiretos e para o usuário em questão essa lista seria composta por *Atividade\_Intensa* e *Atividade\_Noturna*.

Figura 13 - Consulta das classes das atividades de interesse do usuário

```
1 SELECT DISTINCT ?tipo
2 WHERE {
3     ?usuario a :Usuario.
4     ?usuario :ID "Maria Laura"@pt.
5     ?usuario foaf:interest ?interesse.
6     ?interesse rdf:type ?tipo.
7     ?tipo rdfs:subClassOf* :Atividade.
8 }
```

Results

SPARQL Results

tipo
<a href="#">:Atividade_Intensa</a>
<a href="#">:Atividade_Noturna</a>

O terceiro passo da pontuação busca as atividades já praticadas pelo usuário. Essas atividades são associadas ao usuário pela propriedade *praticou*. Para buscar essas atividades praticadas uma terceira consulta é executada e está demonstrada na figura 14. O resultado dessa consulta popula uma lista de atividades praticadas, contendo apenas Vela para este exemplo.

Figura 14 - Consulta das atividades já praticadas

```
1 SELECT DISTINCT ?atividade
2     WHERE {
3         ?usuario a :Usuario.
4         ?usuario :ID "Maria Laura"@pt.
5         ?usuario :praticou ?atividade
6     }
```

Results

SPARQL Results

atividade
 :Vela

O quarto passo da pontuação consiste em buscar todas as atividades presentes na base de conhecimento e para cada atividade verificar se a mesma está presente na lista de pontuadores diretos, se o local onde esta atividade é praticada está na lista de pontuadores diretos, se existem classes dessa atividade na lista de pontuadores indiretos e quantas vezes essa atividade aparece na lista de atividades já praticadas. Tomarei como exemplo a atividade Surfe, que é classificada como Atividade\_Intensa e tem como valor para a propriedade praticadoEmLocal a classe Praia.

Figura 15 - Consulta das propriedades da atividade Surfe.







```

1 SELECT ?atividade ?tipo ?local
2 WHERE {
3   ?atividade rdf:type ?classe.
4   ?atividade :praticadoEmLocalDoTipo ?local.
5   ?classe rdfs:subClassOf* :Atividade.
6   ?classe rdfs:subClassOf? ?tipo
7
8   filter regex(str(?atividade), "Surfe")
9 }
10
11

```

Results

SPARQL Results

atividade	tipo	local
 :Surfe	 :Atividade_Esporte	 :Praia
 :Surfe	 :Atividade_Intensa	 :Praia

Seguindo o exemplo do usuário Maria Laura com a Atividade surfe, existe um pontuador direto (praticado em local Praia), um pontuador indireto ( Surfe classificado como Atividade\_Intensa) e nenhuma pontuação como atividade já praticada. Com essas quantidades podemos aplicar a fórmula 1 e calcular a pontuação desta atividade:  $1 + 1/3 + 0/2 = 1.33$ .

As atividades que pontuam menos do que 0.33 pontos, ou seja, atividades que não pontuaram nem de forma indireta, são descartadas e não são recomendadas ao usuário.

Para o conjunto de atividades com pontuação suficiente para serem recomendadas o filtro contextual é aplicado. Para cada atividade recomendável uma consulta SPARQL é realizada e três informações sobre o contexto físico do usuário são inseridos nesta consulta: o período do dia, a condição do clima na atual localização do usuário e a

localização do usuário. Para se obter o período do dia e a condição climática do local onde se encontra o usuário, dois pequenos componentes foram escritos.

O componente do período do dia é extremamente simples, e sua única função é generalizar a informação contextual do horário. Para horários anteriores a 11:59pm e posteriores a 06:00 am o componente retorna o valor “Matutino”. Para horários entre 12:00pm e 17:59pm o componente retorna “Vespertino”. Para qualquer outro horário que não se incluem nessas duas faixas, o componente retorna “Noturno”.

Já o componente climático, apesar de ainda muito simples, realiza algumas operações a mais. Este componente recebe de entrada as coordenadas geográficas do usuário, e realiza uma requisição HTTP com essas coordenadas a um serviço Web de informações meteorológicas. Este serviço chama-se Darksky (<https://darksky.net/>) e oferece um plano gratuito para até 300 requisições por dia. A resposta da requisição a este serviço contém diversas informações sobre o clima do local informado através das coordenadas, tais como temperatura e um resumo sobre o clima. Todos os possíveis resultados foram classificados em bom, ruim ou neutro da seguinte forma: "clear-day" em bom, "clear-night" em bom, "rain" em ruim, "snow" em ruim, "sleet" em ruim, "wind" em neutro, "fog" em neutro, "cloudy" em neutro, "partly-cloudy-day" em bom e "partly-cloudy-night" em bom.

Com essas informações contextuais preparadas, a consulta final é construída. A união do resultado de cada uma dessas consultas forma o conjunto final de recomendações. A consulta é demonstrada a seguir ainda para a atividade Surfe, sendo esta consulta realizada em Florianópolis pela manhã em um dia ensolarado.

Figura 16 - Filtragem contextual







```

1 SELECT ?atividade ?local
2     WHERE {
3         ?atividade :praticadoEmClima :Clima Bom.
4         ?atividade :praticadoEmPeriodo :Matutino .
5         ?atividade :praticadoEmLocalDoTipo ?tipo .
6         ?local a ?tipo .
7         ?local geo:lat ?lat .
8         ?local geo:long ?long.
9         FILTER( ?atividade=:Surfe &&
10                ?long > -48.4831672 - 1
11                && ?long < -48.4831672 + 1
12                && ?lat > -27.6733903 - 1
13                && ?lat < -27.6733903 + 1)
14     }
15
16

```

Results

SPARQL Results

atividade	local
 :Surfe	 dbpedia:Canasvieiras
 :Surfe	 dbpedia:Mole_Beach
 :Surfe	 dbpedia:Praia_dos_Ingleses

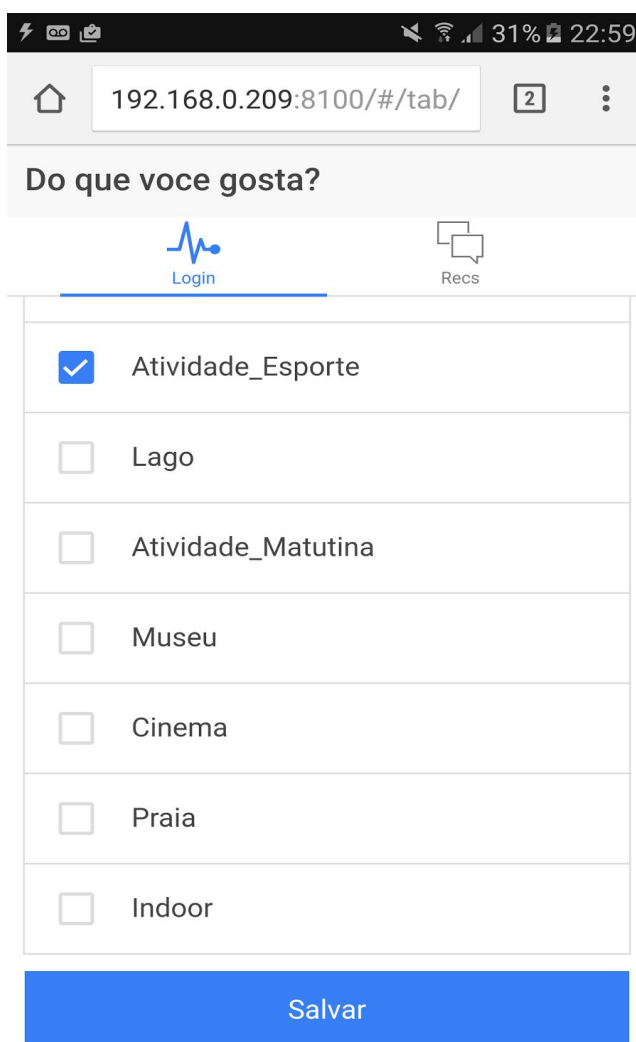
Vale notar que esse retorno demonstra uma limitação do sistema. A única restrição de local praticado para a atividade Surfe é que seja em uma praia, não existe nenhuma informação que diga se essa praia tenha ondas ou não, e dessa forma praias como Canasvieiras são recomendadas para a prática de Surfe.

As consultas SPARQL citadas neste capítulo são realizadas através de requisições HTTP ao endpoint de consulta exposto pelo banco de conhecimento Stardog. O serviço Web é implementado com a tecnologia para desenvolvimento de servidores Web com Javascript chamada NodeJS.

### 6.2.5 Aplicativo Web

A aplicação Web tem duas funções principais, coletar os interesses de usuários que usam o sistema pela primeira vez e coletar as informações contextuais de usuários que já informaram ao sistema atividades e locais que interessam. A Figura 17 abaixo demonstra a tela de coleta de interesses do usuário. As opções demonstradas nesta tela são resultados de uma consulta na base de conhecimento que busca indivíduos e classes da hierarquia de atividades e classes da hierarquia de locais. Essa consulta está demonstrada na imagem xu.

Figura 17 - Tela de coleta de interesses



The screenshot shows a mobile web browser interface. At the top, the status bar displays signal strength, Wi-Fi, 31% battery, and the time 22:59. Below the status bar is a browser address bar with the URL '192.168.0.209:8100/#/tab/'. The main content area has a header 'Do que voce gosta?' and two navigation tabs: 'Login' (with a pulse icon) and 'Recs' (with a speech bubble icon). The 'Login' tab is active. Below the tabs is a list of interest items, each with a checkbox and a label:

- Atividade\_Esporte
- Lago
- Atividade\_Matutina
- Museu
- Cinema
- Praia
- Indoor

At the bottom of the screen is a blue button labeled 'Salvar'.

Para cada item selecionado dessa lista uma propriedade *interest* é criada para o usuário em questão tendo como valor a atividade ou local designado pelo item.

Figura 18 - Consulta de possíveis interesses

```
1 SELECT ?conceito
2     WHERE {
3         {
4             ?conceito rdf:type ?type.
5             ?type rdfs:subClassOf* :Atividade.
6         }
7         UNION
8         { ?conceito rdfs:subClassOf* :Atividade}
9         UNION
10        { ?conceito rdfs:subClassOf* :Local}
11    }
12
```

Results

SPARQL Results

**conceito**

- [:Museu](#)
- [:Teatro](#)
- [:Vela](#)
- [:Surfe](#)
- [:Academia\\_Funcional](#)

Para usuários que já popularam sua lista de interesses, a aplicação Web coleta do usuário sua localização e o horário local. Com a localização e o horário em mãos, a aplicação Web faz uma chamada a API exposta pelo servidor Web que responde com uma lista de possíveis atividades a serem praticadas.



Figura 19 - Tela de recomendações de atividades



A figura 19 demonstra recomendações para um usuário que tinha como interesses surfe, bares, baladas e museus. Porém essa recomendação foi feita no período da manhã e apenas surfe e museus são consideradas atividades matutinas. O ícone de mapa ao lado de cada atividade é um link que leva ao Google Maps com as coordenadas do local para onde a prática da atividade é sugerida. A figura 20 mostra o exemplo de clique no mapa para a opção Museu Casa das Rosas. Para atividades coletadas do DBPedia, ao clicar no nome do local o usuário é direcionado à página do DBPedia deste local.

Figura 20 - Resultado do clique no ícone de mapa



Esta aplicação Web foi escrita com o framework Ionic JS. Ionic é um framework para desenvolvimento de aplicações Web e mobile utilizando tecnologias Web. Aplicações Web desenvolvidas com este framework são responsivas, ou seja se adaptam ao tamanho da tela do dispositivo. Este foi um dos motivos para ter sido escolhido para o desenvolvimento da aplicação Web deste trabalho.

A localização do usuário é obtida através da API *Geolocation* do HTML5. Para dispositivos móveis esta API tenta primeiro buscar as coordenadas do GPS do dispositivo, caso falhe tenta fontes alternativas como a rede WIFI ou rede móvel que o dispositivo está conectado. Para dispositivos desktop, a API busca da rede WIFI do dispositivo.

## 7 CONCLUSÕES

Durante o desenvolvimento da etapa teórica deste trabalho diversos algoritmos e abordagens relacionados à área de sistemas de recomendação sensível ao contexto foram estudadas e alguns trabalhos neste campo foram analisados com maior detalhe. No entanto para o domínio específico deste trabalho, recomendações sensíveis ao contexto de atividades de lazer, pouco foi encontrado e esta área ainda pode ser explorada muito mais a fundo.

Ainda é difícil integrar dados de variadas fontes RDF, essa integração é comumente feita carregando todo o conteúdo remoto para um repositório e fazendo consultas nesse repositório local. Em vários casos isso não será possível devido a restrições técnicas e legais.

Restrições técnicas incluem casos como tamanho da fonte remota de dados muito grande ou fonte remota se atualiza constantemente gerando outro desafio de manter a ontologia local atualizada. Restrições legais incluem casos como fontes de dados não permitirem cópias de bancos inteiros (QUILITZ; LESER, 2008).

Durante o desenvolvimento deste trabalho foi possível ver este problema na prática. A base de conhecimento DBPedia foi utilizada para enriquecer a ontologia desenvolvida neste trabalho, no entanto se a integração fosse de forma remota, não seria possível realizar inferências que classificassem indivíduos situados na base DBPedia com as classes desenvolvidas localmente, o que inutilizaria os indivíduos remotos, já que o processo de pontuação de atividades é baseado na classificação dos indivíduos. Por esse motivo muitos dados do DBPedia foram extraídos e inseridos na base local. Caso atualizações sejam feitas no DBPedia, para refletir essas atualizações um processo manual de extração e inserção seria necessário.

Apesar da dificuldade supracitada, o uso da Web Semântica traz possibilidades extremamente interessantes, utilizar a base de conhecimento DBPedia permitiu que a base desenvolvida para este trabalho tivesse acesso a todas informações de boa parte das praias e museus registrados no Wikipedia por exemplo (DBPedia é atualizado periodicamente e por isso não é completamente atualizado com o Wikipedia).

O fato da Web Semântica ser uma tecnologia ainda recente e ser tecnológica e socialmente disruptiva (PASSANT, 2010) sua adoção ainda não é muito ampla. Isso causa algumas dificuldades como encontrar ontologias maduras para reuso. Para este

trabalho, fora a ontologia FOAF, nenhuma outra ontologia teve seu vocabulário incorporado à ontologia desenvolvida.

Por outro lado, modelar o domínio com ontologia permitiu modelar de forma clara e flexível as atividades e o perfil de usuários, pontos centrais para o desenvolvimento do trabalho. O uso de ontologia também permitiu separação total do domínio com o código do sistema e isso permitiu o desenvolvimento das duas partes em paralelo sem uma grande dor na integração das duas partes. Outra vantagem de usar ontologia foi a capacidade de inferências oferecida. Parte importante da pontuação das atividades foi realizada através de classes inferidas para as atividades através de restrições de classes.

Para modelagem contextuais de período do dia e clima, foram realizadas generalizações de contexto para que esses valores fornecessem mais informações do que simplesmente horário e temperatura. No entanto a generalização para a classe clima foi feito de maneira demasiadamente simples, de forma que algumas atividades podem não possuir nenhum clima adequado para sua prática.

Na etapa prática do trabalho um modelo simples de pontuação para atividades foi proposto e implementado. Para possibilitar testes mais realistas do sistema, um protótipo de aplicação para navegadores Web e Smartphones foi desenvolvido. Os resultados do sistema foram interessantes durante os testes realizados pelo autor e o objetivo de recomendar atividade de lazer de prática possível de acordo com o contexto do usuário foi atingido. No entanto não foram realizados testes efetivos com um número razoável de usuários de forma que, na verdade, a qualidade das recomendações realizadas por este sistema é desconhecida.

Como trabalhos futuros ficam os seguintes pontos:

Realizar testes do sistema com um número significativo de usuários e definir métricas quantitativas para avaliar o desempenho do teste.

Explorar mais o uso do perfil de usuário para as recomendações. A utilização do perfil do usuário pode resolver uma limitação do sistema de precisar de interesses explícitos do usuário antes de realizar uma primeira recomendação. Informações como idade, endereço, profissão, índices de saúde (talvez coletados com utensílios como pulseiras inteligentes) possam fazer uma pré classificação do usuário e já associar a alguma categoria de atividade.

Testar algoritmo de filtragem colaborativa dentro do sistema desenvolvido e realizar análises estatísticas comparativas de desempenho das recomendações. Para isso seria importante o sistema já estar em uso e existir uma base razoável de usuários para que um algoritmo de filtragem colaborativa pudesse realmente fazer sentido.

Rever modelagem do domínio para considerar atividades que sejam praticadas em grupo. Nada na ontologia atual considera quantos participantes uma certa atividade precisa para ser praticada. Muitas atividades não fazem sentido se praticadas de maneira individual ou então são muito menos atrativas e sua recomendação teria uma performance baixa. Unir no contexto do usuário sua companhia com atividades de prática em grupo poderiam trazer resultados positivos.

Criar um mecanismo dentro do aplicativo móvel para usuários de forma pró-ativa adicionarem novas atividades e locais para evoluir o sistema de forma colaborativa.

## 8 REFERÊNCIAS

ABBAR, S; BOUZEGHOUB, B; LOPEZ, S. **Context-aware recommender systems: A service-oriented approach**. *VLDB PersDB workshop*. 2009. August 24-28, 2009, Lyon, France. Disponível <em <http://persdb09.stanford.edu/proceedings/persdb-6.pdf>>. Acesso em: 20 out. 2016.

ADOMAVICIUS, G; TUZHILIN A. **Context-Aware Recommender Systems**. 2010.

ADOMAVICIUS, G.; TUZHILIN, A.. **Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions**. *Knowledge and Data Engineering, IEEE Transactions on* ,v.17, n.6, p. 734-749, 2005.

BERGMAN, K. **Metamodeling in Domain Ontologies** 2010. Disponível em: <<http://www.mkbergman.com/913/metamodeling-in-domain-ontologies/>> Acesso em: 10 de Jul, 2016.

BERNERS-LEE, T.; HENDLER, LASSILA, J.O.. **The Semantic Web** 17,2001

BOBADILLA, J. **Recommender systems survey, 2013**.

BRICKLEY, D; MILLER, L **FOAF Vocabulary Specification** 2014 Disponível em: <<http://xmlns.com/foaf/spec/#sec-foafsw>> Acesso em: 25 de Out, 2016.

BURKE, R. **Integrating Knowledge-based and Collaborative-filtering Recommender Systems, 1999**.

BURKE, R. **Hybrid Recommender Systems: Survey and Experiments** Department of Information Systems and Decision Sciences, California State University, Fullerton, EUA, 2002.

CARRER-NETO, W. *et al.* **Social knowledge-based recommender system. Application to the movies domain 2012**.

COPPENS, S; VANDER, M; SANDE,R; DE MANNENS, E; ,WALLER,R. **Reasoning over SPARQL**. In *LDOW*. 2013.

DING, L.; KOLARI, P.; DING, Z.; AVANCHA, S. **Using ontologies in the semantic web: A survey**. In *Ontologies* (pp. 79-113). Springer US. 2007

FOSKET, D. J. **Theory of clumps**. Readings in Information Retrieval, [S.l.], 1997.

GRAU, B,C; HORROCKS,I.; PARSIA, B.; PATEL-SCHNEIDER, F. P.; SATTLER, U. **Next Steps for OWL**. In *OWLED*. 2006.

GUARINO, N.; GIARETTA, P. **Ontologies and knowledge bases. Towards Very Large Knowledge Bases**, [S.l.], 1995

HEFLIN, J **AN INTRODUCTION TO THE OWL WEB ONTOLOGY LANGUAGE**  
Lehigh University,

KANGNING, W; HUANG, J; SHAOHONG, F. **A Survey of E-Commerce Recommender Systems**

LEHMANN, J; ISELE, R; JAKOB,M; JENTZSCH, A; KONTOKOSTAS, D; MENDES, P; HELLMANN, T; MORSEY, M; VAN KLEEF, P; AUER,S; BIZER, C  
**DBpedia – A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia**  
Semantic Web 1, 2012

LIN, Y.; JESSURUN, J.; DE VRIES,B; TIMMERMANS, H. **Motivate: Towards context-aware recommendation mobile system for healthy living. 2011 5th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops**, pp. 250-253. IEEE, 2011.

MACHADO, G; PALAZZO,J. **CARLO: Modelo Ontológico de Contexto para Recomendação de Objetos de Aprendizagem em Ambientes Pervasivos. 29th SBBD – SBBD Proceedings**, 2014

NOY, N. F.; MCGUINNESS, D. L. **Ontology Development 101: A Guide to Creating Your First Ontology** Stanford University, Stanford, CA, 94305, 2001

OKU, K *et. al.* **Context-aware SVM for context-dependent information recommendation**. In Proceedings of the 7th International Conference on Mobile Data Management, page 109, 2006.

PANIELLO, U. **Experimental Comparison of Pre- vs. Post-Filtering Approaches in Context-Aware Recommender Systems**, 2009.

PARK, D. H., et al. A literature review and classification of recommender systems research. *Expert Systems with Applications*, v. 39, n.11, p. 10059-10072, sept, 2012.

PASSANT, A. *Semantic Web Technologies for Enterprise 2.0* (Vol. 9). IOS Press. 2010.

PRIMO, T. T.. **Método de representação de conhecimento baseado em Ontologias para apoiar Sistemas de Recomendação Educacionais**. 2013. 120 f. Tese (Doutorado) - Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

QUILITZ, B.; LESER, U. **Querying distributed RDF data sources with SPARQL**. In *European Semantic Web Conference* (pp. 524-538). Springer Berlin Heidelberg, 2008.

RAPOZA, J. "SPARQL Will Make the Web Shine". *eWeek*. 2007-01-17.

RENDLE, S.; FREUDENTHALER C.; SCHMIDT-THIEME, L. **Factorizing personalized markov chains for next-basket recommendation**. In: *Proceedings of the 19th International Conference on World Wide Web*. ACM, 2010. WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA. ACM 978-1-60558-799-8/10/04.

SALMAN, Y; ABU-ISSA, A.; TUMAR, I; HASSOUNEH, Y. **A Proactive Multi-type Context-Aware Recommender System in the Environment of Internet of Things**. In *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on* (pp. 351-355). IEEE. 2015.

HAPPEL, H. J.; SEEDORF, S. **Applications of ontologies in software engineering**. In *Proc. of Workshop on Semantic Web Enabled Software Engineering"(SWESE) on the ISWC* (pp. 5-9). 2006.

SHADBOLT, N.; HALL, W. **The Semantic Web Revisited**. University of Southampton Tim Berners-Lee, Massachusetts Institute of Technology, 2006.

Smith, M.K ; Welty,C. ; McGuinness, D. **OWL Web Ontology Language Guide, W3C Working Draft**, 31 March 2003. Disponível em <http://www.w3.org/TR/2003/WD-owl-guide-20030331/>

STARDOG. **Stardog 4: The Manual** 2016. Disponível em: <http://docs.stardog.com/> Acesso em: 20 de Set, 2016.

TRIM,C . **Inference using OWL 2.0 Semantics** abril de 2013

VAN OSSENBRUGGEN, J.; HARDMAN,L; RUTLEDGE, L. **Hypermedia and the Semantic Web: A Research Agenda**. CWI Amsterdam, Kruislaan 413 Amsterdam, The Netherlands 2006

WIKIPEDIA. **Semantic Web Stack** Disponível em: [https://en.wikipedia.org/wiki/Semantic\\_Web\\_Stack](https://en.wikipedia.org/wiki/Semantic_Web_Stack) Acesso em: 15 de Jun, 2016.

W3C. **OWL Web Ontology Language**. 2009. Disponível em: <http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1> Acesso em: 12 de Jun, 2016.

W3C, **OWL 2 Web Ontology Language Structural Specification and Functional-style Syntax (Second Edition)** 12, 2012.

W3C **OWL Web Ontology Language Reference** fevereiro de 2014.

W3C. **SPARQL 1.1 Query Language** 2013. Disponível em: <https://www.w3.org/TR/2013/REC-sparql11-query-20130321/> Acesso em: 05 de Set, 2016.



W3C, **OWL 2 Web Ontology Language New Features and Rationale (Second Edition)**

2012. Disponível em: <<https://www.w3.org/TR/owl2-new-features>>

Acesso em: 21 de Out, 2016.

WIKIPEDIA. **DBpedia** 2016. Disponível em:

<<https://en.wikipedia.org/wiki/DBpedia>>. Acesso em: 25 julho de 2016.

## APÊNDICE A - ARTIGO

### UM SISTEMA DE RECOMENDAÇÃO SENSÍVEL AO CONTEXTO PARA ATIVIDADES DE LAZER

**Vitor Paulon Avancini**

Departamento de Informática e Estatística - Universidade Federal de Santa Catarina

Florianópolis, SC - Brasil

vitorpavancini@gmail.com

**Resumo.** Com a popularização de aparelhos *Smartphones*, diversas informações sobre o contexto do usuário podem ser coletadas sem grandes custos. Uma sub-categoria de sistemas recomendadores que poderia se beneficiar do contexto do usuário são sistemas recomendadores de atividades de lazer. É possível encontrar diversos aplicativos de celular que apresentam opções de atividade de diversas maneiras, como dependendo de um estado de humor que usuário escolher ( triste, alegre, empolgado ..) ou de maneira colaborativa utilizando sugestões de outros usuários. No entanto pouco se encontra sobre sistemas que de fato recomendam atividades de lazer viáveis dentro de um conjunto de restrições, como por exemplo, o sistema não recomendaria a prática de esportes como surfe se o sistema fosse executado às nove horas da noite. Este trabalho apresenta um modelo para recomendação e um protótipo que de fato recomenda atividades ao usuário de acordo com seus interesses e seu contexto.

## 1 INTRODUÇÃO

Sistemas recomendadores são ferramentas que, a partir de dados históricos de um de usuário, filtram conteúdo de maneira a apresentar apenas os itens mais interessantes ao usuário (ABBAR; BOUZEGHOUB; LOPEZ, 2009). Segundo Primo (2013), um sistema de recomendação atua basicamente sugerindo itens de forma pró-ativa, visando auxiliar a escolha de itens em sistemas que possuem informações abundantes. Para Park et. al. (2012), são sistemas que utilizam tecnologia analítica para calcular a probabilidade de um produto ser comprado em um local, de forma que o produto correto possa ser recomendado ao cliente.

Apesar de Sistemas de Recomendação ser uma área que recebeu uma atenção interessante na última década pouco foi feito se considerando o contexto do usuário como localização, temperatura e clima, horário. Os sistemas de recomendação tradicionais consideram apenas duas entidades como base para suas predições: itens e Usuários (ADOMAVICIUS; TUZHILIN, 2010). Se por um lado essa abordagem obteve muito sucesso em algumas ocasiões como venda de livros e discos, em outros domínios essa estratégia pode não ser suficiente. Por exemplo, se considerarmos o contexto temporal e de localização ao recomendar um pacote de viagem, provavelmente recomendar uma viagem à Bariloche no verão argentino seria um erro, já que boa parte do turismo em Bariloche está relacionado à neve. Segundo Abbar, Bouzeghoub e Lopez (2009), um sistema de recomendação eficiente é um sistema que recomende o item certo, na hora e tempo certo, pela mídia certa. Isso seria inalcançável sem informações contextuais.

No domínio de recomendações de atividades de lazer, objeto de estudo deste trabalho, o contexto do usuário é de vital importância. Imagine receber uma recomendação de corrida na rua em um dia de tempestade: a credibilidade do sistema estaria em cheque na primeira utilização.

Com utilização de técnicas da Web Semântica como ontologias é possível modelar esse contexto de forma a associá-lo aos usuários de maneira flexível e ainda se utilizar de motores de inferências e recursos disponíveis na internet. Não somente o contexto físico do usuário foi modelado a partir de técnicas de Web Semântica, todo o domínio do trabalho se beneficiou do uso de ontologias para sua modelagem computacional, como as próprias atividades e o perfil dos usuários.

Aliando as informações contextuais do usuário às técnicas tradicionais de sistemas de recomendação e técnicas da Web Semântica, acreditamos ser possível desenvolver um sistema que recomende atividades de lazer com precisão apreciável e este é o principal objetivo deste trabalho.

## **2 SISTEMAS DE RECOMENDAÇÃO**

Sistemas recomendadores são ferramentas que, a partir de dados históricos de um de usuário, filtram conteúdo de maneira a apresentar apenas os itens mais interessantes ao usuário (ABBAR; BOUZEGHOUB; LOPEZ, 2009).

Segundo Primo (2013), um sistema de recomendação atua basicamente sugerindo itens de forma pró-ativa, visando auxiliar a escolha de itens em sistemas que possuem informações abundantes. Para Park et. al. (2012), são sistemas que utilizam tecnologia analítica para calcular a probabilidade de um produto ser comprado em um local, de forma que o produto correto possa ser recomendado ao cliente.

De acordo com Primo (2013), os sistemas de recomendação podem ser classificados de cinco formas: baseado em conteúdo, filtragem colaborativa, demográfico, baseado em conhecimento e sistema híbrido. Para este trabalho foi utilizada a abordagem baseada em conteúdo, onde a pontuação de um item é calculada de acordo com critérios de similaridade entre o usuário e o item.

## **3 SISTEMAS DE RECOMENDAÇÃO SENSÍVEL AO CONTEXTO**

Diferente dos sistemas tradicionais que fazem suas recomendações incorporando em seu modelo apenas essa relação *Usuário X Item -> Recomendação*, os sistemas de recomendação sensível ao contexto adicionam ao modelo o contexto do usuário:

*Usuário X Item X Contexto -> Recomendação.*

Este contexto pode ser qualquer informação em tempo real acerca do usuário que possa ser útil para a recomendação, tal como tempo, localização e temperatura (PANIELLO, 2009).

Um exemplo de recomendação sensível ao contexto é a recomendação de um sistema de viagem que faria recomendações diferentes dependendo se o destino procurado na data em questão estivesse no verão ou inverno, de modo que, por exemplo um hotel de ski não seria recomendado no verão (RENDLE; FREUDENTHALER; SCHMIDT-THIEME, 2010).

Um sistema de recomendação sensível ao contexto pode ser construído incorporando as informações do contexto de três formas: pré-filtragem, modelagem contextual e pós-filtragem (ADOMAVICIUS; TUZHILIN, 2010). Neste trabalho optou-se por incorporar o contexto pela abordagem de pós filtragem.

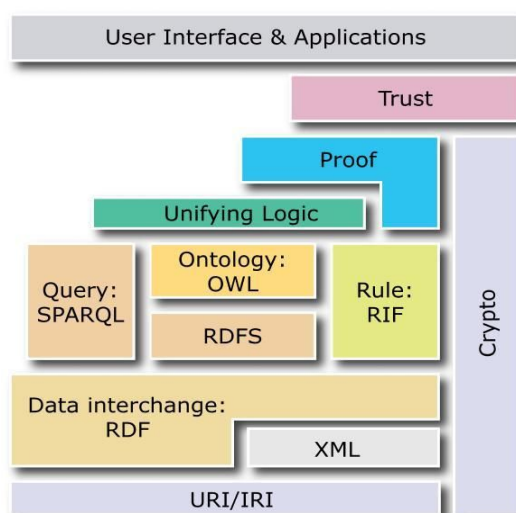
#### 4 WEB SEMÂNTICA

A Web semântica é uma extensão da Web tradicional na qual a informação recebe um significado bem definido, aumentando a possibilidade de pessoas e máquinas trabalharem em conjunto (BERNERS-LEE; HENDLER; LASSILA, 2001).

Segundo a visão de Tim Berners-Lee, a Web semântica irá trazer estrutura para o conteúdo significativo das páginas Web, criando uma ambiente onde agentes de software caminhando de página em página serão capazes de resolver tarefas sofisticadas para usuários (BERNERS-LEE; HENDLER; LASSILA, 2001).

A Web semântica pode ser melhor visualizada como uma pilha de padrões e tecnologias formando uma arquitetura como a representada na figura 1.

Figura 1- Pilha de tecnologias da Web semântica



Fonte: [https://en.wikipedia.org/wiki/Semantic\\_Web\\_Stack](https://en.wikipedia.org/wiki/Semantic_Web_Stack)

Dessa pilha de tecnologias, é especialmente interessante no contexto deste trabalho a ontologia e mais especificamente a linguagem de ontologia Web OWL.

## **5 OWL**

A linguagem de ontologia Web (OWL) é o padrão W3C para representar ontologias na Web semântica (CARRER-NETO et al, 2012 ). A principal ideia da OWL é habilitar representações eficientes de ontologias que também são capazes de participar de processos de decisão (SHADBOLT; HALL, 2006). OWL permite a descrever e instanciar ontologias para a Web (SMITH et al, 2003). Na prática, OWL é uma extensão do RDF schema e adiciona mais vocabulários e expressividade ao RDFS (SMITH et al, 2003 ).

Os elementos básicos de uma ontologia OWL são classes, propriedades, instâncias de classes(indivíduos) e relacionamentos entre essas instâncias.

Classes representam um conjunto de recursos com características similares (HEFLIN, J). Toda classe OWL é associada a um grupo de indivíduos, e esse grupo é chamado de extensão da classe. Uma classe é relacionada mas não é igual à sua extensão, permitindo que duas classes possuam extensões iguais(mesmos indivíduos) e mesmo assim não serem classes iguais (SMITH et al, 2003 ).

A OWL prevê dois tipos de propriedades, propriedades de objetos e propriedade de dados. As propriedades de objetos associam dois indivíduos, enquanto propriedades de dados associam indivíduos á valores literais, como números e strings ( SMITH et al, 2003 )

## **6 SISTEMA DE RECOMENDAÇÃO SENSÍVEL AO CONTEXTO PARA ATIVIDADES**

O sistema recomendador desenvolvido neste trabalho deve recomendar aos usuários atividades que sejam do interesse do usuário. Estes interesses são identificados de duas formas, através de uma coleta explícita por meio de um aplicativo e através de um histórico de atividades já praticadas pelo usuário. Outro ponto que o sistema deve respeitar é que essas atividades devem ser plausíveis de serem praticadas de acordo com o contexto físico

do usuário. Por exemplo, este sistema não recomenda uma corrida na rua em um dia de tempestade.

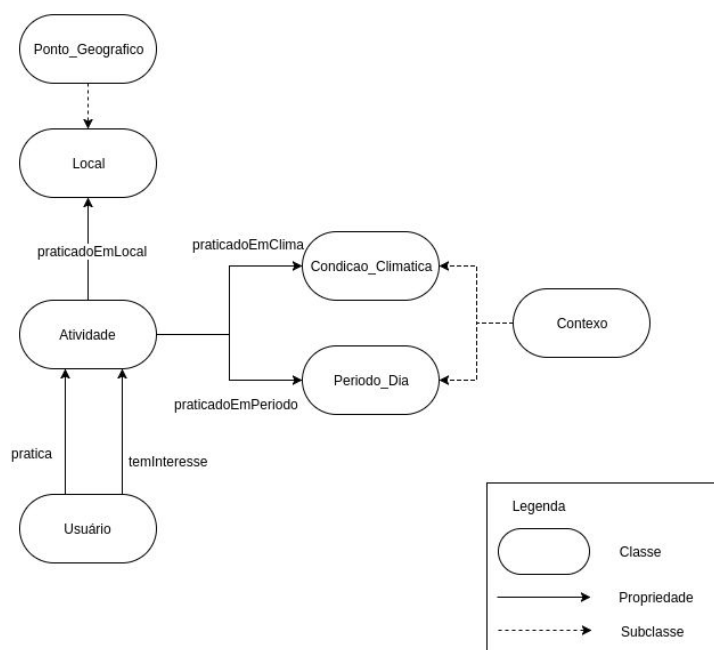
Considerando os requisitos supracitados um modelo ontológico para representação do domínio e um modelo de recomendação baseado em conteúdo foram propostos para desenvolver este sistema. Devido ao requisito de que as atividades recomendadas devem fazer sentido com o contexto do usuário, trata-se também de um sistema de recomendação context-aware.

## **6.1 Modelagem do Domínio**

Para a modelagem do domínio uma ontologia foi desenvolvida e quatro principais classes principais foram propostas: Contexto, Atividade, Usuário e Local. A classe Contexto foi dividida em duas subclasses, Condição Climática e Período Dia. A figura 2 apresenta uma visão simplificada de como o domínio foi modelado.

Para as classes Condição Climática e Período do Dia foram aplicadas generalizações de contexto. As condições climáticas foram generalizadas em três possíveis condições: Clima Bom, Clima Ruim e Clima Neutro. Essa generalização introduz algumas limitações. Se considerarmos, por exemplo, a atividade esquiar, que é praticada na neve. O sistema classifica o clima nevando como Clima Ruim, então ou precisamos modelar que esquiar é praticado em clima ruim ou esquiar nunca será recomendado. Para a classe Período do Dia, os horários foram classificados em período matutino, vespertino e noturno.

### **Figura 2 - Visão geral da Ontologia proposta**



O perfil do usuário, modelado pela classe Usuário da ontologia proposta, é composto basicamente por duas listas, uma de interesses explicitamente selecionados pelo usuário e uma de atividades praticadas que foram recomendadas pelo sistema. A lista de interesses pode conter qualquer valor dentro da hierarquia das classes Atividade e Local, seja uma subclasse ou um indivíduo

## 6.2 Modelo De Recomendação baseado em conteúdo sensível ao contexto

A recomendação do sistema proposto neste trabalho se dá em duas etapas. Primeiramente é calculado uma pontuação para cada atividade de acordo com o perfil do usuário para quem se está recomendando. O cálculo da pontuação das atividades é explicado a seguir.

Para a lista de interesses explícitos do usuário existem dois tipos de pontuação: direta e indireta. A pontuação direta é caracterizada quando a lista de interesses contém a atividade em questão ou um local onde a atividade em questão é praticada. Por exemplo, se a atividade em questão for Surfe e o usuário explicitar que se interessa por surfe, esta é uma pontuação direta. A figura 4 demonstra este caso. Se o usuário também explicitasse interesse em praias, local onde o Surfe é praticado, a atividade receberia mais uma pontuação direta.

A pontuação indireta ocorre quando o usuário possui em seu perfil uma classe que é o tipo da atividade a ser pontuada. Seguindo com o exemplo da atividade Surfe, e supondo



que esta atividade é do tipo Atividade Física, se o usuário possui em sua lista de interesse esta classe Atividade Física, Surfe recebe pontuação indireta. A figura 4 também demonstra este caso, e nesta situação Surfe seria pontuado duas vezes, indireta e diretamente. A pontuação direta resulta em um ponto e a pontuação indireta em um terço de ponto. É importante lembrar que a classe Atividade Física só pode estar presente na lista de interesses devido a técnica de punning, esta lista é representada pelas propriedades interest no perfil do usuário e por isso nesse momento Atividade Física é interpretada, na verdade, como um indivíduo.

Já para a lista de atividades que o usuário já praticou, se a atividade em questão está presente na lista, é adicionado meio ponto ao somatório. Voltando para o exemplo da atividade Surfe, como ela recebe pontuação direta e uma pontuação indireta (atividades podem pontuar indiretamente mais de uma vez), sua pontuação é de um  $1 + 1 * \frac{1}{2} = 1,5$  pontos. Atividades com maior pontuação são recomendadas primeiro. Atividades com a mesma pontuação são ordenadas de forma aleatória.

$$S = (Nd) + (Ni * 1/3) + (Np * 1/2)$$

S - pontuação da atividade

Nd - Número de pontuações diretas

Ni - Número de pontuações indiretas

Np - Número de vezes que atividade foi praticada

#### Fórmula 1 - Pontuação de atividade

As multiplicadores de pontuação  $\frac{1}{2}$ ,  $\frac{1}{3}$  e 1 foram definidos de forma arbitrária. O que eles representam é que uma pontuação direta vale três vezes o que vale uma pontuação indireta e duas vezes o que vale uma pontuação por já ter praticado uma atividade.

A segunda etapa da recomendação é a incorporação do contexto à recomendação. Um requisito definido para o sistema anteriormente é a compatibilidade do contexto do usuário com as características da atividade a ser recomendada. Para manter esta compatibilidade, os requisitos de contexto para uma atividade foram definidos da seguinte maneira:

- O valor da propriedade `praticadoEmPeriodo` deve ser exatamente o período do dia atual do usuário;
- O valor da propriedade `praticadoEmClima` deve ser exatamente a condição climática de onde o usuário se encontra;
- Deve existir pelo menos um indivíduo da classe `Local` que é valor da propriedade `praticadoEmLocal`, e esse indivíduo deve estar situado dentro de um raio de  $X$  quilômetros da localização do usuário.

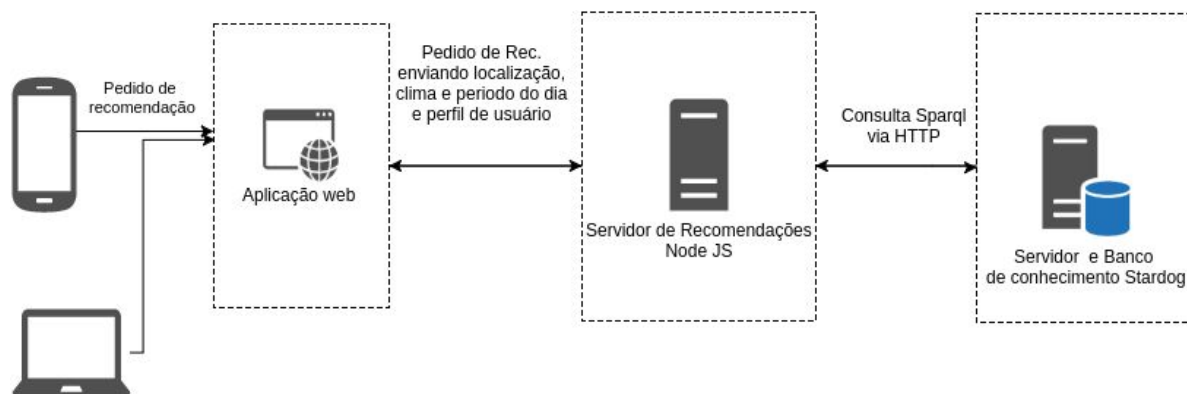
Um possível exemplo seria a atividade Remo. Para este exemplo, digamos que esta atividade é praticada em lagoas, apenas no período matutino e em dias de sol. Para um usuário receber uma recomendação de praticar remo, este usuário precisa estar a uma distância de até  $X$  quilômetros de alguma lagoa, no período da manhã e em um local com clima ensolarado. A escolha de  $X$  vai impactar fortemente na quantidade de recomendações retornadas, para um  $X$  pequeno apenas atividades praticadas em locais muito perto do usuário podem ser recomendadas e é possível que não exista nenhuma para  $X$  demasiadamente pequenos. Por outro lado  $X$  muito grandes podem retornar atividades que não façam sentido para o contexto do usuário. Existe uma troca entre qualidade e quantidade na escolha do tamanho de  $X$ . As atividades que não satisfazem esses requisitos de contexto são removidas do conjunto de atividades a ser recomendado.

### **6.3 Arquitetura do Sistema**

O sistema proposto por este trabalho é composto por três unidades: uma aplicação Web responsiva para fácil uso em smartphones, uma api Web escrita em Node JS que retorna recomendações a partir do terceiro componente do sistema, um banco de conhecimento. Este banco de conhecimento foi construído com a ferramenta Stardog, que

além de armazenar as triplas, possui um motor de inferência e expõe uma interface HTTP para consulta.

Figura 10 - Arquitetura do sistema recomendador



## 7 CONCLUSÕES

Durante o desenvolvimento da etapa teórica deste trabalho diversos algoritmos e abordagens relacionados à área de sistemas de recomendação sensível ao contexto foram estudadas e alguns trabalhos neste campo foram analisados com maior detalhe. No entanto para o domínio específico deste trabalho, recomendações sensíveis ao contexto de atividades de lazer, pouco foi encontrado e esta área ainda pode ser explorada muito mais a fundo.

Ainda é difícil integrar dados de variadas fontes RDF, essa integração é comumente feita carregando todo o conteúdo remoto para um repositório e fazendo consultas nesse repositório local. Em vários casos isso não será possível devido a restrições técnicas e legais.

Restrições técnicas incluem casos como tamanho da fonte remota de dados muito grande ou fonte remota se atualiza constantemente gerando outro desafio de manter a ontologia local atualizada. Restrições legais incluem casos como fontes de dados não permitirem cópias de bancos inteiros (QUILITZ; LESER, 2008).

Durante o desenvolvimento deste trabalho foi possível ver este problema na prática. A base de conhecimento DBPedia foi utilizada para enriquecer a ontologia desenvolvida

neste trabalho, no entanto se a integração fosse de forma remota, não seria possível realizar inferências que classificassem indivíduos situados na base DBPedia com as classes desenvolvidas localmente, o que inutilizaria os indivíduos remotos, já que o processo de pontuação de atividades é baseado na classificação dos indivíduos. Por esse motivo muitos dados do DBPedia foram extraídos e inseridos na base local. Caso atualizações sejam feitas no DBPedia, para refletir essas atualizações um processo manual de extração e inserção seria necessário.

Apesar da dificuldade supracitada, o uso da Web Semântica traz possibilidades extremamente interessantes, utilizar a base de conhecimento DBPedia permitiu que a base desenvolvida para este trabalho tivesse acesso a todas informações de boa parte das praias e museus registrados no Wikipedia por exemplo (DBPedia é atualizado periodicamente e por isso não é completamente atualizado com o Wikipedia).

O fato da Web Semântica ser uma tecnologia ainda recente e ser tecnológica e socialmente disruptiva (PASSANT, 2010) sua adoção ainda não é muito ampla. Isso causa algumas dificuldades como encontrar ontologias maduras para reuso. Para este trabalho, fora a ontologia FOAF, nenhuma outra ontologia teve seu vocabulário incorporado à ontologia desenvolvida.

Por outro lado, modelar o domínio com ontologia permitiu modelar de forma clara e flexível as atividades e o perfil de usuários, pontos centrais para o desenvolvimento do trabalho. O uso de ontologia também permitiu separação total do domínio com o código do sistema e isso permitiu o desenvolvimento das duas partes em paralelo sem uma grande dor na integração das duas partes. Outra vantagem de usar ontologia foi a capacidade de inferências oferecida. Parte importante da pontuação das atividades foi realizada através de classes inferidas para as atividades através de restrições de classes.

Para modelagem contextuais de período do dia e clima, foram realizadas generalizações de contexto para que esses valores fornecessem mais informações do que simplesmente horário e temperatura. No entanto a generalização para a classe clima foi feito de maneira demasiadamente simples, de forma que algumas atividades podem não possuir nenhum clima adequado para sua prática.

Na etapa prática do trabalho um modelo simples de pontuação para atividades foi proposto e implementado. Para possibilitar testes mais realistas do sistema, um protótipo de

aplicação para navegadores Web e Smartphones foi desenvolvido. Os resultados do sistema foram interessantes durante os testes realizados pelo autor e o objetivo de recomendar atividade de lazer de prática possível de acordo com o contexto do usuário foi atingido. No entanto não foram realizados testes efetivos com um número razoável de usuários de forma que, na verdade, a qualidade das recomendações realizadas por este sistema é desconhecida.

## 8 REFERÊNCIAS

BERNERS-LEE, T.; HENDLER, LASSILA, J.O.. **The Semantic Web** 17,2001

RENDLE, S.; FREUDENTHALER C.; SCHMIDT-THIEME, L. **Factorizing personalized markov chains for next-basket recommendation**. In: *Proceedings of the 19th International Conference on World Wide Web*. ACM, 2010. WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA. ACM 978-1-60558-799-8/10/04.

ABBAR, S; BOUZEGHOUB, B; LOPEZ, S. **Context-aware recommender systems: A service-oriented approach**. *VLDB PersDB workshop*. 2009. August 24-28, 2009, Lyon, France. Disponível <em <http://persdb09.stanford.edu/proceedings/persdb-6.pdf>>. Acesso em: 20 out. 2016.

PRIMO, T. T.. **Método de representação de conhecimento baseado em Ontologias para apoiar Sistemas de Recomendação Educacionais**. 2013. 120 f. Tese (Doutorado) - Programa de Pós-Graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.

PARK, D. H., et al. **A literature review and classification of recommender systems research**. *Expert Systems with Applications*, v. 39, n.11, p. 10059-10072, sept, 2012.

PANIELLO, U. **Experimental Comparison of Pre- vs. Post-Filtering Approaches in Context-Aware Recommender Systems**, 2009.

ADOMAVICIUS, G; TUZHILIN A. **Context-Aware Recommender Systems**. 2010.

SMITH, M.K ; WELTY,C. ; MCGUINESS, D. **OWL Web Ontology Language Guide, W3C Working Draft**, 31 March 2003. Disponivel em <http://www.w3.org/TR/2003/WD-owl-guide-20030331/>