



**CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS**

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"  
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

Alquimar Sousa Almeida Filho

**DESENVOLVIMENTO DE UM SISTEMA DE RECOMENDAÇÃO DE  
PRODUTOS APOIADO PELA API DO BUSCAPÉ**

Palmas - TO

2014

Alquimar Sousa Almeida Filho

**DESENVOLVIMENTO DE UM SISTEMA DE RECOMENDAÇÃO DE  
PRODUTOS APOIADO PELA API DO BUSCAPÉ**

Trabalho de Conclusão de Curso (TCC) elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. Mestre Fernando Luiz Oliveira.

Palmas – TO

2014

Alquimar Sousa Almeida Filho

**DESENVOLVIMENTO DE UM SISTEMA DE RECOMENDAÇÃO DE  
PRODUTOS APOIADO PELA API DO BUSCAPÉ**

Trabalho de Conclusão de Curso (TCC) elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. Mestre Fernando Luiz Oliveira.

Aprovada em 27 de junho de 2014.

**BANCA EXAMINADORA**

---

Prof. M.Sc. Fernando Luiz Oliveira  
Centro Universitário Luterano de Palmas

---

Prof. M.Sc. Cristina D'Ornellas Filipakis  
Centro Universitário Luterano de Palmas

---

Prof. M.Sc. Parcilene Fernandes de Brito  
Centro Universitário Luterano de Palmas

Palmas – TO

2014

## **DEDICATÓRIA**

Dedico este trabalho aos meus avós, aos meus pais, a minha irmã e, aos meus amigos, por todo apoio, confiança e compreensão a mim direcionados.

## AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me dado força e motivação para que eu tivesse a capacidade de superar as dificuldades e desenvolver este trabalho, que representa a conclusão do curso.

Ao meu avô, por sempre ter confiado em mim e por ter me incentivado a fazer este curso.

Ao meu pai, que sempre me motivou a continuar, sempre me dando conselhos para nunca desistir e por ser um dos principais incentivadores para que eu concluísse o curso.

À minha avó, mãe e irmã, por me darem total suporte para que eu conseguisse me dedicar mais ao curso.

Aos meus amigos por estarem sempre ao meu lado me apoiando e me aconselhando nos momentos que mais precisei.

Ao meu professor e orientador Fernando Luiz, por ter acreditado em mim e por ter me dado a oportunidade de realizar este trabalho, além de ter sido muito importante na minha formação, assim como todos os demais professores.

A todos os professores do curso de Sistemas de Informação, ao qual tenho muito orgulho de ter sido aluno, pela qualidade e competência de toda equipe. Em especial ao professor Fabiano, que me aconselhou em muitos momentos importantes no decorrer de minha formação e também ao professor Edeilson por ter me apoiado e ajudado muito na realização deste trabalho.

Ao meu amigo Warley, por ter me incentivado e ajudado a realizar este trabalho.

## **RESUMO**

Os sistemas de recomendação baseiam-se nas informações a respeito dos usuários para poder recomendar produtos de acordo com suas preferências, fazendo com que os clientes encontrem os produtos que desejam com mais facilidade. Para atender melhor ainda as necessidades dos clientes, é interessante levar em conta o valor dos produtos, tendo em vista que a busca geralmente é pelos menores preços. Para gerar as recomendações, o Sistema de Recomendação precisa filtrar as informações a respeito dos usuários, o que é possível através da utilização das técnicas de filtragem. Assim, ao unir uma ferramenta de comparação de preços a um sistema de recomendação, espera-se obter um sistema de recomendação de produtos que atenda as necessidades dos clientes, recomendando novos produtos com os melhores preços de mercado. Neste contexto, o objetivo deste trabalho é apresentar os conceitos e as etapas envolvidas no desenvolvimento de uma ferramenta de recomendação de produtos, que irá gerar recomendações de produtos de acordo com o perfil do usuário e fazer uma busca que retornará os produtos recomendados com os menores preços do mercado.

**PALAVRAS-CHAVE:** Sistema de Recomendação, Recuperação da Informação, Filtragem

## LISTA DE FIGURAS

Figura 1 – Etapas de um Sistema de Recomendação .....	11
Figura 2 – Captura explícita. ....	22
Figura 3 – Captura implícita. ....	23
Figura 4 – Demonstração da criação da url (API do Buscapé). ....	25
Figura 5 – Arquitetura do Sistema de Recomendação. ....	34
Figura 6 – Modelo do perfil do usuário. ....	35
Figura 7 – Diagrama de Casos de Uso. ....	36
Figura 8 – Tela de avaliações. ....	37
Figura 9 – Formulário de avaliações. ....	38
Figura 10 – Formulário para gerar as recomendações. ....	39
Figura 11 – Lista de produtos recomendados. ....	39
Figura 12 – Classe responsável pelas recomendações. ....	40
Figura 13 – Classe responsável por ler arquivos no formato json. ....	42

## LISTA DE TABELAS

Tabela 1 – Matriz de avaliações usuário x item .....	16
Tabela 2 – Exemplo de aplicação da técnica <i>stemming</i> .....	19
Tabela 3 – Exemplo de aplicação da técnica <i>stopword</i> .....	20

## LISTA DE ABREVIATURAS

- API – *Application Programming Interfaces*
- CE – Comércio Eletrônico
- FBC – Filtragem Baseada em Conteúdo
- FC – Filtragem Colaborativa
- FH – Filtragem Híbrida
- IDE – *Integrated Development Environment*
- JDBC – *Java Database Connectivity*
- RI – Recuperação da Informação
- SR – Sistema de Recomendação

## SUMÁRIO

<b>INTRODUÇÃO</b> .....	<b>8</b>
<b>1 REFERENCIAL TEÓRICO</b> .....	<b>10</b>
<b>1.1 Sistema de Recomendação</b> .....	<b>10</b>
<b>1.2 Técnicas de Recomendação</b> .....	<b>13</b>
<b>1.2.1 Filtragem Baseada em Conteúdo (FBC)</b> .....	<b>13</b>
<b>1.2.2 Filtragem Colaborativa (FC)</b> .....	<b>15</b>
<b>1.2.3 Filtragem Híbrida (FH)</b> .....	<b>17</b>
<b>1.3 Recuperação da Informação (RI)</b> .....	<b>18</b>
<b>1.4 Perfil do Usuário</b> .....	<b>20</b>
<b>1.5 API do Buscapé</b> .....	<b>24</b>
<b>2 MATERIAIS E MÉTODOS</b> .....	<b>27</b>
<b>2.1 Materiais</b> .....	<b>27</b>
<b>2.1.1 Apache Mahout Taste</b> .....	<b>27</b>
<b>2.1.2 API do Buscapé</b> .....	<b>30</b>
<b>2.2 Metodologia</b> .....	<b>31</b>
<b>3 RESULTADOS E DISCUSSÃO</b> .....	<b>33</b>
<b>3.1 Arquitetura da ferramenta</b> .....	<b>33</b>
<b>3.2 Perfil do usuário</b> .....	<b>35</b>
<b>3.2.1 Diagrama de Casos de Uso</b> .....	<b>36</b>
<b>3.3 Implementação</b> .....	<b>37</b>
<b>3.3.1 Classe Recomendação</b> .....	<b>40</b>
<b>3.3.2 Classe LerJSON</b> .....	<b>42</b>
<b>CONSIDERAÇÕES FINAIS</b> .....	<b>44</b>
<b>Trabalhos Futuros</b> .....	<b>44</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>46</b>

## INTRODUÇÃO

Com o advento da internet surgiram muitos sites de comércio eletrônico, fazendo com que os clientes tenham inúmeras opções de produtos ofertados por estas lojas virtuais. Porém, se por um lado isto é vantajoso, pois os clientes têm várias opções para encontrar o produto desejado (além do preço que tende a diminuir por causa da concorrência), por outro, sem ferramentas apropriadas o usuário terá que acessar inúmeras páginas de *e-comerces* para achar e comparar os preços.

Atualmente, na internet, existem várias ferramentas de comparação de preços e produtos, sendo que uma das mais conhecidas nesse ramo é o Buscapé. A ferramenta Buscapé existe aproximadamente há 15 anos e atualmente é líder na área de comparação de preços de produtos na internet. Porém, apenas estas ferramentas não solucionam o problema, pois elas fazem apenas uma parte do processo, que no caso é a comparação do preço de produtos, elas não fazem recomendações, apenas apresentam produtos específicos com a variação de preços.

Existem ferramentas de recomendação que são responsáveis por identificar usuários, armazenar suas preferências e fazer recomendações, que podem ser produtos, serviços e/ou conteúdos, de acordo com as necessidades e interesses do usuário. No entanto, apenas a ferramenta de recomendação não resolve o problema por completo, já que ela apenas recomenda produtos de acordo com o interesse do usuário, porém sem o preço dos produtos.

Neste contexto, para preencher as lacunas deixadas pelas duas categorias de ferramentas citadas acima, foi proposta e desenvolvida uma ferramenta que ofereça estes serviços de forma integrada, ou seja, um software que realiza tanto a recomendação de produtos, o que foi possível através da utilização de uma ferramenta de recomendação de produtos, no caso, a *API Apache Mahout Taste*, quanto a comparação de preços destes produtos, que foi possível

através da utilização da API do Buscapé, que retorna uma lista de produtos com os menores preços. Esta ferramenta atende ao objetivo deste trabalho, que é desenvolver uma ferramenta capaz de construir o perfil de seus usuários e através deles identificar suas preferências, que em conjunto com a API do Buscapé, seja capaz de recomendar produtos de seu interesse com os menores preços.

Este trabalho está organizado da seguinte forma: na seção 1 são apresentados todos os conceitos pertinentes e necessários para o desenvolvimento deste trabalho, compondo o seu referencial teórico, onde são abordadas as técnicas de Filtragem, Recuperação da Informação (RI), Sistemas de Recomendação (SR) e API do Buscapé. A seção 2 descreve a metodologia deste trabalho, apresentando as tarefas realizadas, bem como os materiais/ferramentas utilizados no desenvolvimento do mesmo. A seção 3, além de descrever os resultados é responsável por apresentar os assuntos referentes a etapa de desenvolvimento, o funcionamento da ferramenta desenvolvida, bem como o detalhamento da implementação. Por fim, vêm as Considerações Finais, seção responsável por relatar as conclusões do trabalho e apresentar trabalhos futuros, que podem ser gerados a partir deste.

# 1 REFERENCIAL TEÓRICO

Nesta seção serão apresentados os conceitos e ferramentas necessárias para o desenvolvimento da ferramenta de recomendação personalizada de produtos proposta nesse trabalho. Para tanto, é necessário entender sobre Técnicas de Recomendação e Sistemas de Recomendação e seus demais conceitos.

## 1.1 Sistema de Recomendação

As ferramentas de recomendação são utilizadas para facilitar a busca de produtos e/ou serviços para os usuários, pois estas têm o objetivo de reduzir a massa de informações, através da seleção de conteúdo baseada nos perfis existentes, assim como fazer recomendações. De acordo com Albuquerque (2008), essas ferramentas são classificadas em três categorias, referente ao método utilizado para recomendação:

- a primeira abordagem é baseada em conteúdo, onde são recomendados itens semelhantes aos que o usuário demonstrou preferência
- a segunda abordagem é chamada de colaborativa, onde recomenda-se itens escolhidos por pessoas com preferência semelhante ao usuário, exemplo: João e Pedro têm o mesmo perfil, o produto que João escolher pode ser recomendando para Pedro e vice-versa.
- Por fim, a terceira e última abordagem é chamada de híbrida, por fazer uma junção das duas abordagens apresentadas anteriormente e também utiliza técnicas que tentam solucionar possíveis problemas das abordagens baseada em conteúdo e abordagem colaborativa.

Segundo Venson (2002, apud BRESOLIN, 2008, p. 2): “Um sistema de recomendação

nada mais é que um mecanismo capaz de aprender através de iterações dos usuários com o sistema de recomendação, afim de obter experiências para poder recomendar, dentre os produtos disponíveis, ou seja, o que tiver uma maior relação com o usuário”. Sendo assim, uma sistema de recomendação não indica apenas o que as pessoas têm interesse direto, elas também podem indicar algo que uma pessoa que possui gostos semelhantes comprou, ou seja, os produtos que uma pessoa de perfil “X” comprou podem ser indicados para pessoas que possuem esse mesmo perfil “X”.

A figura 1, a seguir, demonstra através de figuras o funcionamento de um SR.



**Figura 1** – Etapas de um Sistema de Recomendação (MIRANDA, 2011, p. 12).

Como mostra a figura 1, um Sistema de Recomendação recebe dados com entrada (informações sobre o usuário e informações sobre a participação da comunidade/grupo), em seguida, os métodos de recomendação utilizam essas entradas para gerar a saída (recomendações) e por fim apresenta as recomendações de alguma forma. Mas para tudo isso

funcionar perfeitamente é imprescindível que o perfil do usuário tenha sido construído com relevância, pois é através destes perfis que a ferramenta de recomendação saberá quais tipos de produtos poderá recomendar a determinado usuário.

Conforme Schafer (2011 apud MIRANDA, 1999, p. 1), através da utilização de Sistemas de Recomendação (SR) é possível aumentas as vendas de um Comércio Eletrônico (CE) conforme os três itens a seguir:

1. Convertendo visitantes em compradores: são os usuários que navegam pelos sites de CE mas não compram. Neste caso, o SR pode ajudar o usuário a encontrar o produto que deseja, através de sugestão de produtos ou apresentação de uma lista de produtos mais vendidos;
2. Aumentando a venda adicional: o SR pode utilizar a técnica *cross-sell*, conhecida como “venda casada” de produtos, para melhorar as vendas adicionais. Para tanto o SR recomenda produtos adicionais no momento da compra de um produto, ou então se baseia nos itens existentes no carrinho de compra, onde os itens sugeridos (produtos adicionais) têm relação com o que o cliente está comprando;
3. Ganhando e aumentando a fidelidade: como o SR baseia-se nas preferências do usuário para fazer as recomendações, fica claro para o cliente que a empresa se preocupa em conhecer quem está comprando, para oferecer um atendimento diferenciado aos seus clientes. Portanto, quanto mais o cliente utiliza o *e-commerce*, mais o SR irá aprender sobre as preferências do cliente, possibilitando a recomendação de produtos cada vez melhor, assim o SR satisfaz e fideliza o cliente.

Os sistemas de recomendação são baseados em técnicas de recomendação, pois estas técnicas que são responsáveis pelo principal objetivo desses sistemas, que no caso, consiste na recomendação de produtos. Portanto, para desenvolver um sistema de recomendação é

imprescindível que seja feito um estudo sobre as técnicas de recomendação, que serão apresentadas na próxima sessão.

## **1.2 Técnicas de Recomendação**

São técnicas utilizadas para recomendar produtos, serviços, conteúdo, entre outros, por meio de dados gerados por um grupo de pessoas. Através de uma análise deste conjunto de dados é possível gerar informações relevantes que são utilizadas para definir os perfis das pessoas, com o objetivo de encontrar semelhanças entre elas, uma vez que, para recomendar algo a alguém é necessário saber quais são as suas preferências, interesses, gostos etc. As recomendações são feitas baseadas nos perfis gerados, pois estes perfis possuem um conjunto de interesses semelhantes.

Para auxiliar os consumidores a encontrar produtos de sua preferência, as empresas de Comércio Eletrônico (CE) utilizam Sistemas de Recomendação (SR), que são responsáveis por coletar informações dos usuários para descobrir suas preferências e a partir dessa descoberta oferecer produtos relevantes. Os SR utilizam técnicas de filtragem de informação para descobrir as relações e similaridades que existem entre os produtos, entre os consumidores e entre ambos, essas técnicas são: Filtragem Baseada em Conteúdo (FBC), Filtragem Colaborativa (FC) e Filtragem Híbrida (FH), as quais serão detalhadas nas sessões seguintes.

### **1.2.1 Filtragem Baseada em Conteúdo (FBC)**

A Filtragem Baseada em Conteúdo baseia-se nas características dos itens que um usuário consumiu para poder construir seu perfil. A recomendação é feita de acordo com a relação que existe entre o conteúdo do item e as preferências extraídas do perfil do usuário.

No entanto, o perfil do usuário de ser constantemente atualizado com dados obtidos

através de consultas realizadas por ele, ou através do conteúdo dos itens consumidos pelo próprio usuário. Para tanto, utiliza-se uma técnica conhecida como *Term Frequency Indexing* (Indexação de Frequência de Termos), onde cada usuário é representado por um vetor formado por um conjunto de dimensões, onde essas dimensões representam cada palavra que ocorre no documento ou base de dados e cada componente do vetor representa a quantidade de vezes (frequência) que uma determinada palavra ocorre em um documento ou consulta feita pelo usuário (CAZELLA, 2010, p. 3).

Para saber quais são os termos mais relevantes para um determinado usuário é preciso calcular o peso de cada termo, sendo que este peso é obtido através da frequência do termo (*Term Frequency*), representando pela seguinte equação:

$$tf_{i,j} = \frac{freq}{maxfreq}$$

O termo *freq*, da equação acima, corresponde à quantidade de vezes (frequência) que um termo ocorre em um determinado documento/consulta, esta frequência (*freq*) é dividida pela máxima frequência (*maxfreq*) do termo que mais ocorre no documento/consulta, resultando no peso do termo que foi calculado. Portanto, se existe um termo x cuja frequência é 10 e um termo y com frequência 8 e um termo z com frequência 4, para saber o peso do termo z basta dividir a frequência de z pela frequência de x, que no caso é o termo mais frequente.

A FBC parte do princípio que se o usuário consumiu um item anteriormente ele poderá consumir também outros itens similares. Dessa forma, para que se possa saber que itens são similares a outros, é necessário fazer uma análise do conteúdo desses produtos, ou seja, fazer uma correlação dos produtos que já foram consumidos pelo usuário com os novos produtos que estão na base (MIRANDA, 2011, p. 15).

Como relata Lázaro (2010), a FBC possui problemas que estão relacionados à super-especialização, pois os métodos dessa técnica permitem apenas a recomendação de itens similares ao perfil do usuário, ou seja, se um usuário comprou apenas livros de romance, então a ferramenta não irá lhe recomendar nenhum outro tipo de livro, portanto corre o risco

da recomendação ser semelhante às anteriores.

A FBC está ligada a área de Recuperação da Informação, responsável por recuperar informações para que os usuários tenham acesso às informações recuperadas, facilitando o acesso a informações necessárias ao usuário. A sessão 1.3 irá abordar com mais detalhe o referido conceito.

Na seção seguinte será apresentada a técnica de Filtragem Colaborativa (FC), que funciona de uma forma diferente da FBC.

### **1.2.2 Filtragem Colaborativa (FC)**

O objetivo da Filtragem Colaborativa (FC) é recomendar novos itens ao usuário, utilizando como base as preferências dos usuários que possuem o mesmo perfil, ou seja, é baseado na avaliação de produtos que foram comprados ou avaliados por usuários que possuem o mesmo perfil. Assim, é feita uma análise sobre do perfil dos usuários que possuem interesse em comum e em seguida recomenda-se novos produtos, onde a recomendação é baseada nas avaliações realizadas pelas pessoas que possuem interesses em comum.

Sendo assim, um Sistema de Recomendação que utiliza a técnica de FC depende muito da interação dos usuários com a ferramenta, pois é preciso que os usuários comprem produtos e avaliem para que através destas avaliações e dos perfis construídos o sistema possa recomendar produtos baseado nos perfis semelhantes.

A Filtragem Colaborativa busca prever a pontuação de um determinado item para um consumidor isoladamente, onde essa predição é baseada na nota dos produtos que foram avaliados por pessoas que possuem perfil semelhante ao usuário alvo, conforme Cazella (2006, p. 30). Na FC cada usuário contribui através de suas avaliações e essas avaliações impactam diretamente no desempenho do sistema, o que dá a ideia de comunidade.

A tabela 1, a seguir, demonstra uma matriz de produtos avaliados por vários usuários.

**Tabela 1** – Matriz de avaliações usuário x item (CAZELLA, 2006, p. 31).

Usuários	Itens			
	(A1) Fab: Content-Based, Collaborative Recommendation	(A2) Using collaborative filtering to weave an information Tapestry	(A3) Social information filtering: Algorithms for automating "word of mouth"	(A4) Grouplens: Applying Collaborative Filtering to Usenet News
João	4	4	5	2
Maria	-	3	-	5
José	4	5	5	?
Rosa	-	2	3	5

Na tabela 1, as linhas representam os usuários, as colunas representam os produtos e os dados da matriz representam as notas dos produtos que foram atribuídas pelos usuários através de sua avaliação. Os perfis de João e José são considerados semelhantes, pois as notas dos produtos avaliados por eles são semelhantes o que indica que o perfil de ambos é parecido, o que acontece também com Maria e Rosa, porém as avaliações de Maria e Rosa não são semelhantes as avaliações de João e José o que indica que elas possuem um perfil diferente do de João e José.

A tabela acima apresenta as notas de cada item que foi avaliado por um determinado usuário. O usuário “José” ainda não possui uma nota definida para o item “A4”, portanto é preciso saber quais são os vizinhos que possuem o perfil mais parecido com o de “José”. Para tanto, é preciso analisar a similaridade das avaliações dos usuários, ou seja, encontrar usuários com notas de avaliação dos itens semelhantes às notas de “José” referentes aos mesmos itens. A partir dessa análise é possível prever qual seria a nota que o usuário alvo (José) daria para o item “A4”, então, de acordo com esta nota o item “A4” poderia ou não ser recomendado ao usuário alvo (José).

Neste exemplo, o usuário “José” é o alvo. Portanto, percebe-se que o usuário que possui uma avaliação dos itens mais semelhante à avaliação de “José” é o usuário “João”, pelo fato das notas dos itens que foram avaliados por “João” serem parecidas com as notas avaliadas nos mesmos itens pelo “José”. Observa-se que “João” avaliou os itens (A1, A2 e A3) respectivamente com as notas (4, 4 e 5) e “José” avaliou os mesmos itens com as respectivas notas (4, 5 e 5). Portanto, como o perfil de “João” é semelhante ao de “José”, é grande a probabilidade de sucesso na recomendação do item “A4” para “José”, baseado na semelhança dos perfis.

### **1.2.3 Filtragem Híbrida (FH)**

O objetivo da Filtragem Híbrida é fazer uma junção das duas técnicas citadas nos parágrafos anteriores, que são, Filtragem Baseada em Conteúdo (FBC) e Filtragem Colaborativa (FC), pois assim é possível que algumas limitações existentes nessas técnicas sejam solucionadas e/ou melhorados. A FH agrega características das duas técnicas (FBC e FC) para atender melhor as necessidades dos usuários, pois dessa forma é possível obter os benefícios da FBC, que no caso seria a predição para os usuário e itens, independente da quantidade de usuários e avaliações dos produtos e também a maior probabilidade de acerto nas predições feitas pela FC, que é melhorada de acordo com o crescimento da quantidade de usuários no sistema e das recomendações realizadas. Segundo Lázaro (2010, p. 25) existem algumas formas possíveis de se combinar FBC com FC, que são:

- Ponderada: implementar as técnicas de Filtragem Baseada em Conteúdo e Filtragem Colaborativa separadamente, porém, combinando suas predições, ou seja, ambas podem conter um mesmo peso inicial e de acordo com as recomendações realizadas esses pesos são atualizados;

- Mista: a recomendação é feita através da combinação das recomendações feitas pelas técnicas FBC e FC;
- Combinação seqüencial: os perfis dos usuários são criados de acordo com os itens acessados por eles, através da FBC e em seguida aplica a FC para realizar as predições baseadas nos perfis gerados pela FBC;
- Comutação: o sistema utiliza de algum critério para combinar as técnicas FBC e FC. Podendo alterná-las, ou seja, primeiramente aplica a FBC e caso as recomendações feitas por esta técnica não sejam satisfatórias aplica-se a FC, ou vice-versa.

### 1.3 Recuperação da Informação

Recuperação da Informação (RI) é uma área da computação que trabalha com o armazenamento, manipulação e recuperação automática de um conjunto dados em formatos distintos. Seu objetivo é gerar conhecimentos através de informações recuperadas de documentos, bancos de dados, ou qualquer outra fonte, ou seja, auxiliar o usuário na busca de informações que o interessam (RODRIGUES, 2009, p. 19).

O processo de Recuperação da Informação é classificado em algumas outras áreas, que são: recuperação de dados, recuperação de documentos, recuperação de informações ou recuperação de textos.

O processo de recuperação de informação consiste em identificar, no conjunto de documentos (*corpus*) de um sistema, quais atendem à necessidade de informação do usuário. O usuário de um sistema de recuperação de informação está, portanto, interessado em recuperar “informação” sobre um determinado assunto e não em recuperar dados que satisfazem sua expressão de busca, nem tampouco documentos, embora seja nestes que a informação estará registrada (FERNEDA, 2003, p. 14,15).

A primeira etapa da do processo de Recuperação da Informação é chamada de Pré-Processamento. Esta etapa é responsável por analisar os dados dos documentos para melhorar o resultado da recuperação da informação na busca por dados e/ou informações semelhantes,

de forma que realize uma “limpeza”, filtrando o que é relevante, removendo as redundâncias e irrelevância, reduzindo a base de dados. Para realizar essa etapa é preciso aplicar as técnicas de *stemming* e *stopwords*. A técnica de *Stemming* é a responsável por encontrar os termos que possuem significados semelhantes e reduzi-los ao seu radical. Portanto, cada variação morfológica de um termo é eliminada e considera-se apenas a sua raiz, isso é possível através da eliminação dos prefixos e/ou sufixos. Assim, os termos que poderiam ser considerados distintos, por causa da forma como foram escritos, são unificados.

A Tabela 2, a seguir, demonstra um exemplo de aplicação da técnica *Stemming*.

**Tabela 2** – Exemplo de Aplicação da Técnica *Stemming*.

<b>Termos</b>	<b><i>Stemming</i></b>
Pensar	Pensa
Pensativo	Pensa
Pensador	Pensa
Pensamento	Pensa

A Tabela 2 mostra um exemplo de aplicação da técnica *stemming*, na qual as palavras “Pensar”, “Pensativo”, “Pensador” e “Pensamento” tiveram seus respectivos sufixos “r”, “tivo”, “dor” e “mento” removidos, resultando em um mesmo radical “Pensa”.

Já a técnica de *Stopwords* é a responsável por filtrar os termos do texto de forma que permaneçam apenas os termos relevantes, ou seja, ela remove do texto os termos irrelevantes, como pronomes, artigos, preposições e conjunções, pois esses termos são repetidos várias vezes e servem apenas para dar sentido ao texto. Um exemplo de termos irrelevantes seria: a, o, na, no, que, qualquer, ainda, seja, mas, entre outros.

Portanto, caso exista um documento para ser analisado e supondo que esse documento contenha o seguinte texto: “Metade das favelas brasileiras tem acesso à internet, diz

pesquisa”. Então, esse documento seria dividido conforme a Tabela 2, a seguir:

**Tabela 3** – Exemplo de Aplicação da Técnica *Stopword*.

<b>Texto do Documento</b>	<b>Stopwords</b>	<b>Termos Relevantes</b>
<b>Metade das favelas brasileiras tem acesso à internet, diz pesquisa</b>	das	Metade
	tem	favelas
	à	brasileiras
		acesso
	diz	internet
pesquisa		

No exemplo de aplicação da técnica *stopword*, demonstrado na Tabela 3, foram removidos do texto os termos irrelevantes (*stopwords*), respectivamente “das”, “tem”, “à” e “diz” mantendo somente os termos relevantes “Metade”, “favelas”, “brasileiras”, “acesso”, “internet” e “pesquisa”.

Após o processo de Recuperação da Informação (RI), em conjunto com as técnicas de *stemming* e *stopword*, é possível obter um documento mais seletivo, contendo apenas os termos mais relevantes do documento analisado. A próxima sessão falará sobre o perfil dos usuários.

#### **1.4 Perfil do Usuário**

Para que um sistema de recomendação consiga alcançar o seu principal objetivo, que no caso consiste na recomendação de produtos, é necessário entender como os perfis dos usuários serão criados, afinal, é com base nos perfis que o sistema irá recomendar os produtos. Esta seção explicará o processo de criação dos perfis de usuários.

Conforme Dudev et al. (2008 apud CASTRO & BARBOSA, 2009, p.16), para construir o perfil, que servirá como base para a recomendação personalizada de produtos, é necessário

seguir três etapas importantes, que são:

- **Modelagem do usuário:** fase em que é feita uma análise que avalia que tipo de características do usuário são importantes para a personalização, ou seja, quais os interesses, habilidades e preferências do usuário. Estas características são identificadas e classificadas como críticas ou não críticas para o sistema de recomendação e são divididas em grupos.
- **Perfil do usuário:** é composto pelo conjunto de características do usuário, onde este conjunto é formado por meio dos dados obtidos através da observação do comportamento do usuário e/ou de informações inseridas por ele de forma direta ou indireta, formando assim o perfil do usuário.
- **Personalização:** é a etapa final, onde o perfil do usuário é utilizado para fazer as recomendações personalizadas.

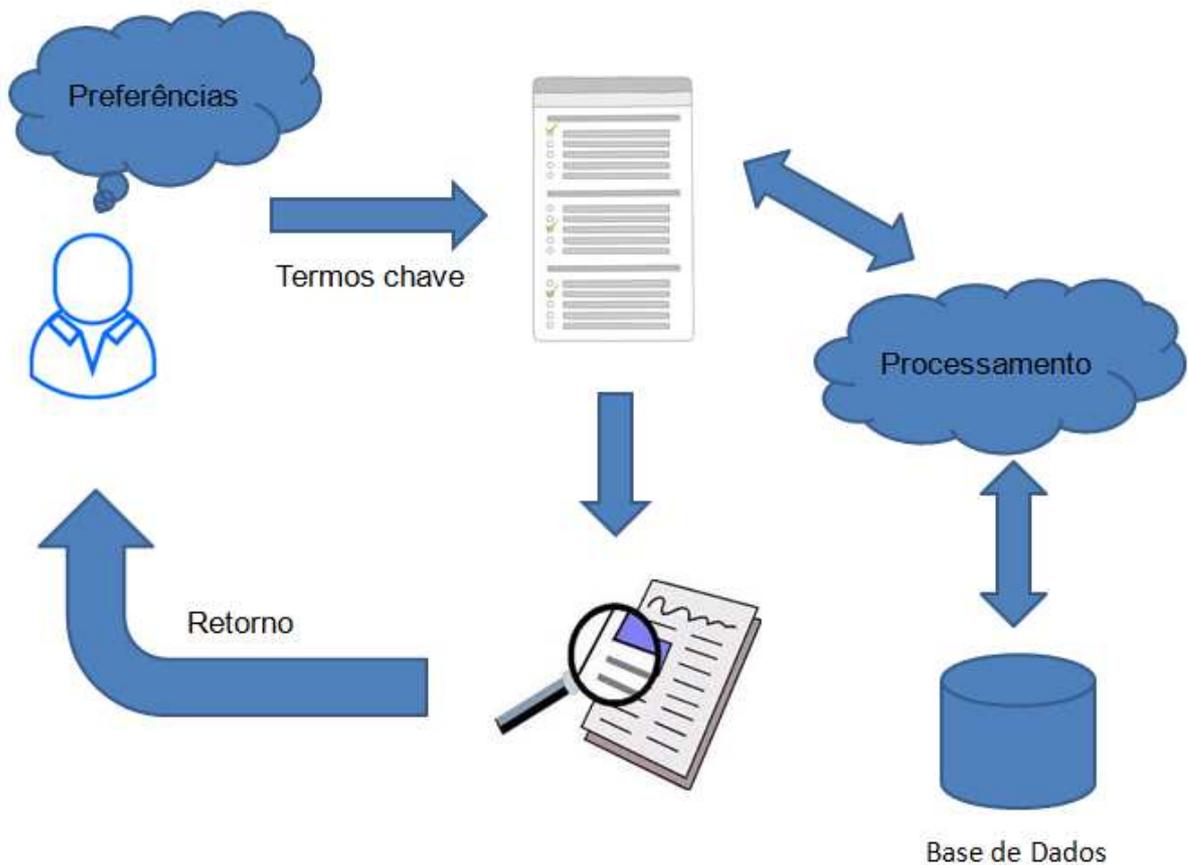
Então, para construir o perfil de um usuário é preciso primeiramente coletar informações a respeito das suas preferências, para que estas informações sejam armazenadas, processadas e analisadas, gerando assim um perfil que represente o usuário no sistema. Por fim, as recomendações são geradas com base nos perfis e em seguida o sistema recomenda produtos associados ao perfil do usuário.

Existem duas formas para realizar a coleta de informações necessárias para a construção dos perfis, que são:

- **Explícita:** nessa abordagem as informações são capturadas diretamente, ou seja, o usuário informa suas preferências ao sistema. Para captar informações de forma explícita, geralmente, os sistemas utilizam formulários com algumas perguntas, que em conjunto caracterizam as preferências do usuário, podendo também ser através de avaliação e/ou classificação de produtos (através de notas que classificam os produtos como bom ou ruim). Essa abordagem é muito interessante

pelo fato de ser mais confiável, pois os próprios usuários que são responsáveis por informar suas preferências para o sistema, diminuindo bastante a chance de ocorrer uma recomendação falha. Porém, essa abordagem necessita da boa vontade por parte do usuário, o que gera um possível problema, pois nem sempre o usuário tem um tempo disponível ou o interesse para responder os formulários e fazer as avaliações/classificações dos produtos.

A figura 2, a seguir, demonstra a coleta explícita, onde o usuário é responsável por informar as suas preferências ao sistema. Isso é possível através de formulários disponibilizados no sistema.



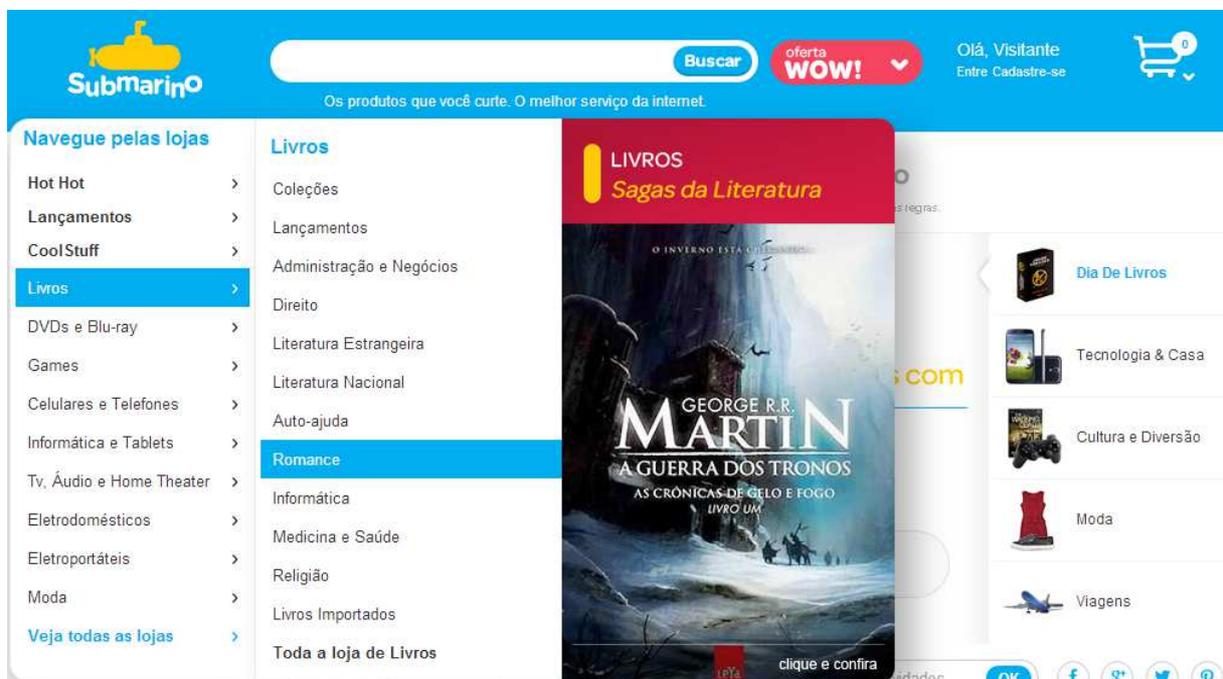
**Figura 2** – Captura Explícita (MATOS LIMA et al., 2010, p. 4).

A figura 2 demonstra o fluxo do processo de captura de informação de forma explícita, onde o usuário é responsável por informar suas preferências.

- Implícita: nessa abordagem as informações são captadas sem que o usuário

perceba, ou seja, é de forma indireta, sem a necessidade do usuário informar suas preferências para o sistema. Geralmente, essa coleta é feita através do monitoramento do comportamento e da interação do usuário com o site. Alguns exemplos deste tipo de coleta são os históricos de palavras-chave buscadas pelo usuário, históricos de compras e/ou avaliação/classificação de produtos e outros.

A figura 3, a seguir, representa a captura de informações de forma implícita, ou seja, capta as informações sobre as preferências do usuário sem que o mesmo perceba. Isso é possível através das interações do usuário com o sistema.



**Figura 3** – Captura Implícita (Foto da página do site Submarino).

A figura 3 representa a coleta de informações de forma implícita, ou seja, as informações são captadas sem que o usuário perceba, apenas através das interações do usuário com o site, de acordo com os itens que ele procura.

Fazendo uma comparação entre as duas abordagens apresentadas acima, é possível perceber que a segunda abordagem (implícita) é mais ideal quando se têm usuários com pouca disponibilidade de tempo ou que não têm interesse em informar suas preferências, porém, os

perfis criados podem não ser tão confiáveis, pelo fato de que é o próprio sistema que decide quais informações são relevantes ou não para o usuário. Na primeira abordagem (explícita) não ocorre este problema, pois como foi explicado em sua definição, são os próprios usuários que informam suas preferências, porém, os usuários nem sempre têm disponibilidade de tempo e/ou interesse de informar suas preferências, o que também caracteriza um possível problema dessa abordagem.

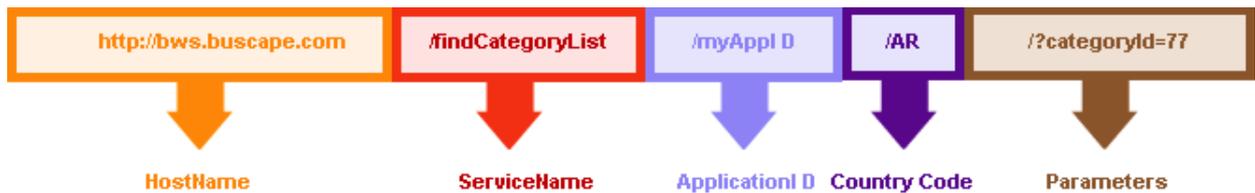
## 1.5 API do Buscapé

*Application Programming Interface* (API), ou Interface de Programação de Aplicativos, em português, são métodos desenvolvidos por uma ferramenta que forma um conjunto de funções que são disponibilizados para que outros softwares possam utilizá-las. O software que utiliza essa API não precisa envolver-se com detalhes da implementação da mesma, mas apenas utilizar suas funcionalidades. Um exemplo de API é o próprio Sistema Operacional (SO), pois este disponibiliza muitas funções, que permitem ao programador criar janelas, criptografar dados etc, facilitando o desenvolvimento de aplicações.

“A API (Interface de Programação de Aplicativos) do Buscapé é um conjunto de funções e padrões estabelecidos para criação de aplicativos customizados, em que há necessidade de uso da base de dados de produtos, ofertas e serviços oferecidos pelo Buscapé” (BUSCAPE, 2013).

Para utilizar a API do Buscapé é necessário, primeiramente, obter um ID para sua aplicação, que será utilizado em todas as requisições dos métodos solicitados.

A figura 4, a seguir, demonstra como a URL é construída:



**Figura 4** – Demonstração da criação da URL. Foto retirada da página oficial (API do Buscapé).

- **HostName:** URL principal do webservice;
- **ServiceName:** nome do serviço;
- **ApplicationId:** ID da aplicação;
- **CountryCode:** Código do país;
- **Parameters:** Parâmetros do serviço.

O formato padrão de resposta é XML, mas o formato JSON pode ser utilizado, facilitando a integração com JavaScript e Flash de maneira muito ágil.

A API do Buscapé é uma biblioteca que fornece uma base de dados que contém os produtos, ofertas e categorias de várias lojas virtuais. Através da utilização dessa API é possível obter a lista de categorias, produtos e ofertas existentes, essa consulta é realizada através da passagem de parâmetros como: id (categoria, produto), palavra-chave etc.

Os métodos disponibilizados por esta API são:

- **Lista de ofertas:** retorna uma lista de ofertas;
- **Lista de categorias:** retorna uma lista de categorias;
- **Lista de produtos:** retorna uma lista de produtos.

Todos esses métodos possuem a seguinte opção de busca:

- **Por ID da Categoria:** retorna uma lista de ofertas de uma determinada categoria;
- **Por ID de produto:** retorna uma lista de oferta de determinado produto;
- **Por código de barras:** retorna uma lista de ofertas de um determinado código de barras;

- Por palavra-chave: retorna uma lista de ofertas baseada em uma palavra-chave;
- ID da categoria com palavra-chave: retorna uma lista de ofertas baseada em uma categoria e na palavra-chave.

## 2 MATERIAIS E MÉTODOS

Neste tópico será apresentada a metodologia utilizada, assim como os procedimentos que foram utilizados durante o desenvolvimento deste trabalho. É importante ressaltar que o objetivo deste trabalho é realizar um estudo aprofundado sobre Sistemas de Recomendação para obter um conhecimento necessário para desenvolver uma ferramenta que possa, além de utilizar das Técnicas de Recomendações existentes, interagir com a API do Buscapé para fazer uma recomendação personalizada de produtos com os melhores preços.

### 2.1 Materiais

Os materiais utilizados no desenvolvimento deste trabalho podem ser divididos em duas partes, que são:

- Fontes bibliográficas: materiais utilizados no desenvolvimento deste trabalho, como: dissertações, artigos, teses, monografias e publicações científicas;
- Softwares: para o desenvolvimento da ferramenta de recomendação foi utilizado o ambiente de desenvolvimento Eclipse Kepler, juntamente com a API *Apache Mahout Taste* e a API do Buscapé. A linguagem de programação utilizada foi Java. As tecnologias e ferramentas utilizadas neste trabalho serão melhor apresentadas nas seções seguintes.

#### 2.1.1 Apache Mahout Taste

Esta API é responsável por gerar as recomendações. Estas recomendações são geradas através do uso da técnica de Filtragem Colaborativa (FC). Esta API é um projeto de código aberto, desenvolvido pela *Apache Software Foundation*, que fornece uma biblioteca Java que possibilita o uso de alguns algoritmos, sendo os principais: *cluster*, classificação e filtragem colaborativa (INGERSOLL, 2009, ONLINE).

Para utilizar essa API é necessário que a base de dados, que pode ser um banco de dados ou um arquivo que contenha os dados necessários, possua uma tabela no seguinte formato: uma coluna que representa a chave primária do usuário (tipo: *long*), uma coluna que representa a chave primária do item (tipo: *long*) e uma coluna que representa a preferência (tipo: *float*). Com este padrão é possível utilizar a API para gerar as recomendações.

A versão utilizada neste trabalho foi a mahout-core 0.3, que pode ser obtida no site: <http://grepcod.com/snapshot/repo1.maven.org/maven2/org.apache.mahout/mahout-core/0.3/>.

Para utilizar esta API é necessário fazer o download dos seguintes pacotes:

- *mahout-core-0.3.jar*, que pode ser obtido através do link: <http://grepcod.com/snapshot/repo1.maven.org/maven2/org.apache.mahout/mahout-core/0.3/>;
- *uncommons-math-1.0.2.jar*, que pode ser obtido através do link: <http://mirrors.ibiblio.org/pub/mirrors/maven2/org/apache/mahout/uncommons-math/uncommons-math/1.0.2/uncommons-math-1.0.2.jar>;
- *slf4-nop-1.6.1.jar*, que pode ser obtido através do link: <http://grepcod.com/snapshot/repo1.maven.org/maven2/org.slf4j/slf4j-nop/1.6.1>;
- *slf4j-api-1.6.1.jar*, que pode ser obtido através do link: <http://grepcod.com/snapshot/repo1.maven.org/maven2/org.slf4j/slf4j-api/1.6.1>.

Posteriormente, é necessário utilizar a ferramenta Eclipse Kepler para criar um projeto do tipo *WEB Project*. Depois de ter criado o projeto é necessário criar uma pasta dentro do projeto e, em seguida, adicionar os pacotes ao projeto. Para adicionar os pacotes, basta clicar com o botão direito na pasta criada e escolher a opção “adicionar JAR”.

A biblioteca mahout-core 0.3 possui várias classes implementadas, sendo que neste trabalho foram utilizadas:

- *DataModel* – interface que pode ser implementada pelo *MySQLJDBCDataModel* ou *FileDataModel*, que são responsáveis por carregar as informações que serão utilizadas para gerar as recomendações, podendo esta ser uma base de dados ou um arquivo txt, csv etc. Para que seja possível gerar as recomendações, é necessário que a base de dados siga o padrão especificado no início desta seção. Assim é possível utilizar uma base de dados *MySQL*, com a utilização da classe *MySQLJDCDataModel*, ou então, utilizar arquivos como base de dados através da classe *FileDataModel*, desde que este arquivo esteja no formato txt, csv, dat, entre outros.
- *UserSimilarity* – interface que pode ser implementada pela classe *PearsonCorrelationSimilarity*, que utiliza o algoritmo que calcula quais os usuários possuem perfil semelhante ao usuário alvo, retornando valores entre -1.0 e 1.0, sendo que o valor 1.0 representa total semelhança e, quanto mais próximo, maior a semelhança; e o inverso com relação ao valor -1.0. Para utilizar essa classe é preciso indicar a base de dados que contém as informações dos usuários.
- *UserNeighborhood* – interface que pode ser implementada pela classe *NearestNUserNeighborhood*, que é utilizada para identificar quais vizinhos possuem perfil semelhante ao usuário alvo, pois as recomendações serão baseadas nesse conjunto de usuários que possuem perfis semelhantes. Para utilizar essa classe, é preciso indicar a quantidade limite de vizinhos, que são os usuários que possuem perfil semelhante ao usuário alvo, para que a ferramenta tome de base para realizar as recomendações, como também o segundo parâmetro que consiste no algoritmo que será utilizado para calcular a similaridade entre os usuários, e por fim a base de dados que contém as informações necessárias para a realização das recomendações.

- *Recommender* – interface que pode ser implementada pela classe *GenericUserBasedRecommender*, que é responsável por gerar as recomendações. Essa classe necessita de três parâmetros, que são: *DataModel*, *UserNeighborhood* e *UserSimilarity*. Essa classe possui o método *recommend*, que recebe a chave primária do usuário que deseja fazer as recomendações, como também a quantidade máxima de recomendações que deseja e retorna uma lista de itens do tipo *RecommenderItem*.
- *RecommenderItem* – objeto retornado através da chamada do método *recommend()* da classe *Recommender*. Para acessar o conteúdo desse objeto, são disponibilizados dois métodos: *getItemId()*, que retorna a chave primária do objeto e o *getValue()*, que retorna o valor da preferência, que no caso, é o possível valor que o usuário alvo poderia avaliar o item, sendo que este valor é predito na geração da recomendação.

### 2.1.2 API do Buscapé

Após a obtenção das recomendações geradas é possível utilizar a API do buscapé para que esta, de acordo com os itens retornados na recomendação, retorne uma lista de produtos com os menores preços. Para tanto, é preciso realizar as seguintes etapas: gerar as recomendações, recuperar os produtos através da API do Buscapé, de acordo com os itens que foram recomendados e em seguida apresentar os produtos para o usuário alvo.

A API do Buscapé disponibiliza vários métodos de busca, sendo que cada um dos métodos possui uma forma diferente de filtragem. Os métodos mais utilizados são:

- Busca por id da categoria: retorna os produtos de acordo com o id da categoria;
- Busca por palavra chave: retorna os produtos que contenham a palavra chave na sua descrição;

- Busca por palavra chave e id da categoria: retorna os produtos de acordo com o id da categoria e que contenham a palavra chave na sua descrição.

A API do buscapé retorna a lista de produtos de duas formas, podendo ser em xml ou json. Por padrão, os métodos já retornam as lista no formato xml, porém, é possível passar como parâmetro o formato de retorno desejado. Para tanto, é necessário informar o formato de retorno, bastando apenas acrescentar o parâmetro *format* com o valor *json*, da seguinte forma: `format=json`.

## 2.2 Metodologia

O desenvolvimento deste trabalho envolveu, basicamente, duas etapas. A primeira etapa foi primordial para que a subsequente acontecesse, pois nela foi realizado um estudo aprofundado sobre os conceitos básicos de um Sistema de Recomendação (definição, aplicação, contexto aplicado), e somente com este conhecimento foi possível identificar e entender as Técnicas de Recomendação, e em seguida, compreender o funcionamento de um SR. Como resultado desta etapa foi desenvolvida a revisão de literatura, que foi dividida da seguinte forma:

- Sistemas de Recomendação (SR): seção em que estão contidos os conceitos de um SR. Nesta seção é possível entender o que são SR's, qual a utilidade desses sistemas, como chegar a um sistema desses, enfim, é uma abordagem geral sobre Sistemas de Recomendação (SR);
- Técnicas de Recomendação: nesta seção são apresentadas as técnicas de recomendação mais utilizadas. A partir disso é possível entender o que são técnicas de recomendação, quais são elas, para que servem e como utilizá-las;
- Recuperação da Informação (RI): seção responsável por explicar o conceito de RI. Através dessa sessão é possível entender o processo de recuperação da informação,

como é feito esse processo, qual sua importância para um sistema de recomendação;

- Perfil de usuário: seção responsável por definir os conceitos envolvidos na criação do perfil do usuário e as técnicas aplicadas para a criação do mesmo. Por meio desta seção é possível entender o que é o perfil do usuário, o que ele contém e o que é preciso para construí-lo;
- API do Buscapé: seção responsável pela apresentação da ferramenta de busca disponibilizada pelo Buscapé. Esta seção faz uma apresentação geral sobre a API do Buscapé, ou seja, explica o que é essa API, como utilizá-la, para que serve e qual a importância de utilizá-la juntamente com o sistema de recomendação que será desenvolvido.

A segunda etapa consiste no desenvolvimento da ferramenta. No entanto, para que o sistema fosse desenvolvido foi necessário:

- Analisar a ferramenta de comparação de preços do Buscapé, pois foi necessário entender como ela funciona e qual a importância de utilizar o mesmo modelo no sistema que será desenvolvido;
- Estudar a API do Buscapé, pois foi através dessa API que os dados dos produtos serão recuperados para que a recomendação seja feita, ou seja, a API do Buscapé está integrada ao sistema desenvolvido neste trabalho;
- Analisar a API *Apache Mahout Taste*, pois esta disponibiliza classes e métodos que são responsáveis por gerar as recomendações de acordo com a base de dados disponibilizada. Para tanto, foi necessário entender o seu funcionamento, bem como suas classes e métodos;
- Definir, modelar e implementar o modelo de perfil do usuário;
- Por fim, implementar o protótipo funcional da ferramenta propriamente dita.

### **3 RESULTADOS E DISCUSSÃO**

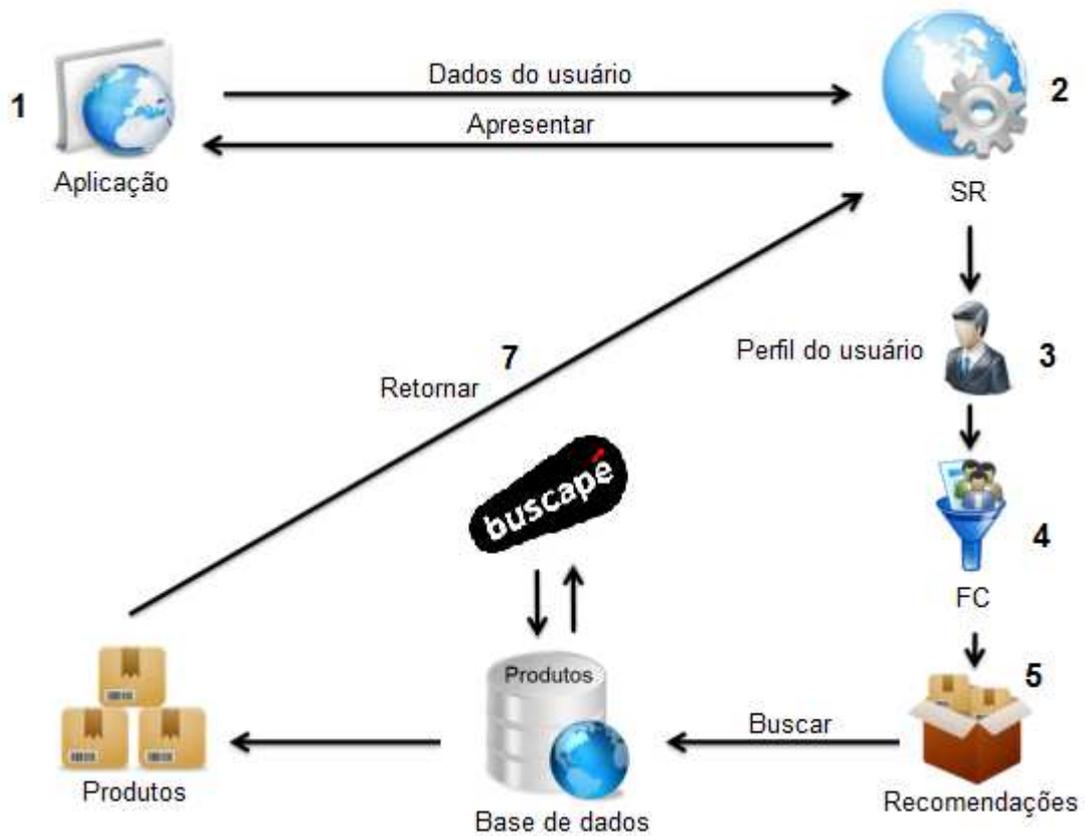
O objetivo desta seção é apresentar os resultados obtidos no desenvolvimento do sistema proposto neste trabalho, que permitirá a recomendação de produtos de acordo com o interesse e realidade do usuário. O sistema permitirá que os usuários avaliem os produtos de seu interesse e estas avaliações serão utilizadas como base para a construção dos perfis. O sistema também permitirá que o usuário solicite recomendações de produtos. Essas recomendações serão geradas de acordo com as avaliações, que são utilizadas para detectar as semelhanças existentes entre os usuários.

As próximas seções apresentarão a arquitetura do Sistema de Recomendação (SR), o modelo do usuário e a implementação da ferramenta.

#### **3.1 Arquitetura da Ferramenta**

Para gerar as recomendações, a ferramenta desenvolvida permite que os usuários avaliem os produtos de seu interesse, isso é possível através da interação do usuário com a ferramenta, na qual o usuário pode selecionar os produtos de seu interesse. Esses registros de avaliações são utilizados como base para a identificação dos perfis semelhantes. Tendo como base estas avaliações, é possível gerar uma quantidade de recomendações para um usuário alvo.

A figura 5 representa a arquitetura do Sistema de Recomendação proposto neste trabalho.



**Figura 5** – Arquitetura do Sistema de Recomendação.

Como pode ser observado na Figura 5, a arquitetura do Sistema de Recomendação (SR) foi dividida em 7 fases. A primeira fase é responsável pela chamada do SR através da aplicação, que é responsável por enviar os dados do cliente para a recomendação. A segunda fase é onde começa a recomendação propriamente dita, pois o SR irá solicitar a fase 3, que consiste na consulta ao perfil do usuário, pois é através do perfil que o SR irá gerar as recomendações. A quarta fase consiste em analisar o perfil do usuário alvo, detectando na base de dados, quais usuários possuem perfil semelhante ao usuário alvo, isso é possível através da Filtragem Colaborativa (FC), técnica utilizada na API *Mahout Apache Taste*. A quinta fase é responsável por retornar os produtos de acordo com o perfil do usuário que foi identificado nas fases 3 e 4, onde os produtos serão obtidos através da API do Buscapé. A sexta fase é responsável por fazer uma busca na base de dados do Buscapé e retornar os produtos de acordo com as recomendações geradas na quinta fase. A sétima e última fase é a responsável por retornar as recomendações geradas ao site, para que este apresente as recomendações ao cliente.

### 3.2 Perfil do usuário

Para gerar as recomendações é necessário que no banco de dados da aplicação contenha uma tabela de avaliações. Esta tabela deve seguir o modelo solicitado pela ferramenta de recomendação *Apache Mahout Taste*. A Figura 6 representa o modelo do perfil do usuário.



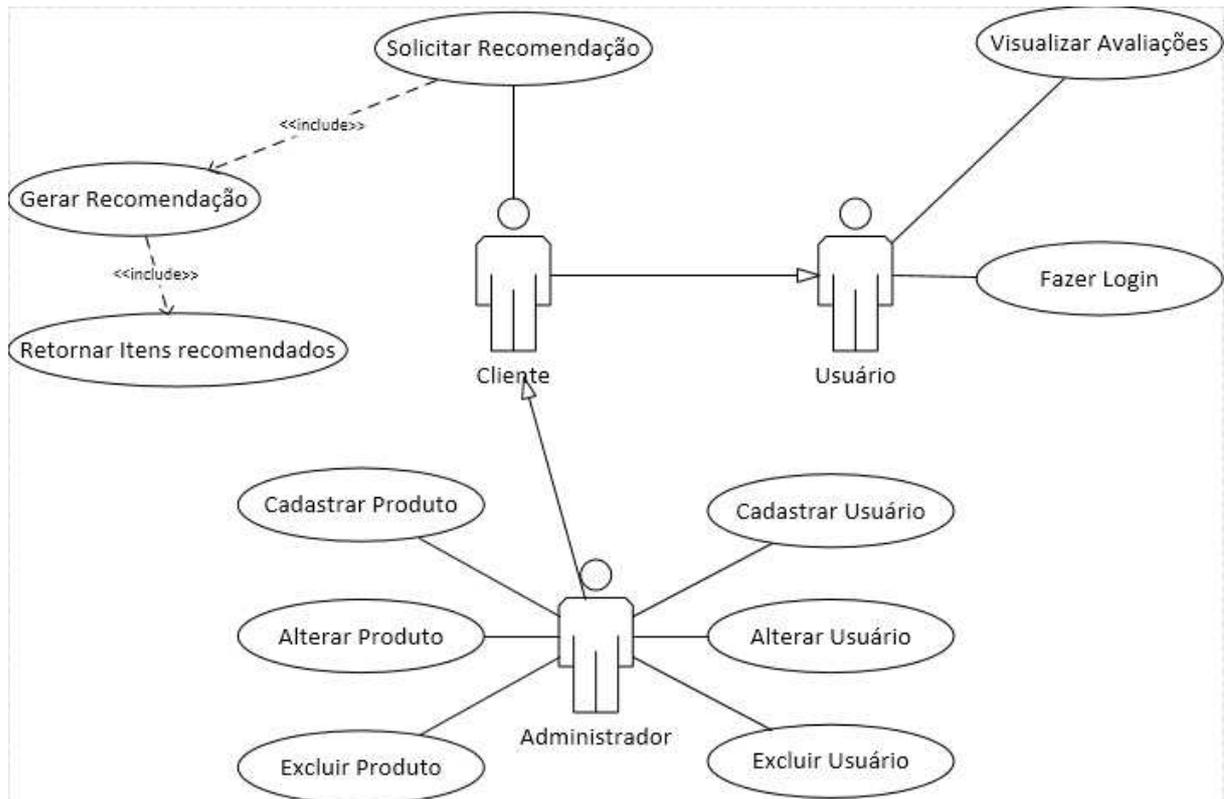
**Figura 6** – Modelo do Perfil do Usuário.

Como pode ser observado na Figura 6, o modelo do perfil do usuário é composto pela seguinte estrutura:

- **Usuários:** tabela responsável por armazenar os dados a respeito do usuário;
- **Produtos:** tabela responsável por armazenar os dados a respeito dos produtos;
- **Avaliações:** tabela responsável por armazenar os dados a respeito de todas as avaliações feitas pelo usuário;

### 3.2.1 Diagrama de Casos de Uso

O objetivo desta seção é apresentar o Diagrama de Casos de Uso da ferramenta desenvolvida. Através deste diagrama é possível visualizar o fluxo das ações que podem ser realizadas através da ferramenta. A Figura 7 demonstra o diagrama.



**Figura 7** – Diagrama de Casos de Uso.

Como pode ser observado na Figura 7, o diagrama contém três atores, que são: Administrador, Cliente e Usuário. Cada um dos atores pode executar tarefas distintas, sendo que o “Cliente”, além de suas tarefas, também pode executar as tarefas do “Usuário”, pois ele herda suas características. O “Administrador” pode executar todas as tarefas além das que são exclusivas a ele, pois ele herda as características do “Cliente”, que por sua vez herda as características do “Usuário”, como mostra a figura. Através deste diagrama é possível visualizar as funcionalidades existentes na ferramenta desenvolvida neste trabalho.

### 3.3 Implementação

O objetivo desta seção é apresentar os passos e artefatos gerados no desenvolvimento da ferramenta, bem como suas funcionalidades. A ferramenta desenvolvida neste trabalho, permite que um usuário que esteja logado no sistema avalie os produtos de seu interesse, o que é possível através do menu “Avaliações”, onde o usuário visualiza a lista de produtos avaliados.

Através das avaliações, é possível gerar recomendações de produtos para um determinado usuário. Para tanto, é necessário que o usuário navegue até o menu “Recomendação”, selecione a opção “Gerar recomendação”. Em seguida deve ser informado qual o cliente que deseja obter as recomendações e qual a quantidade máxima de recomendações que deve ser gerada, por fim, deve-se confirmar a solicitação. Essas funcionalidades serão apresentadas a seguir, através de imagens capturadas da tela da ferramenta.

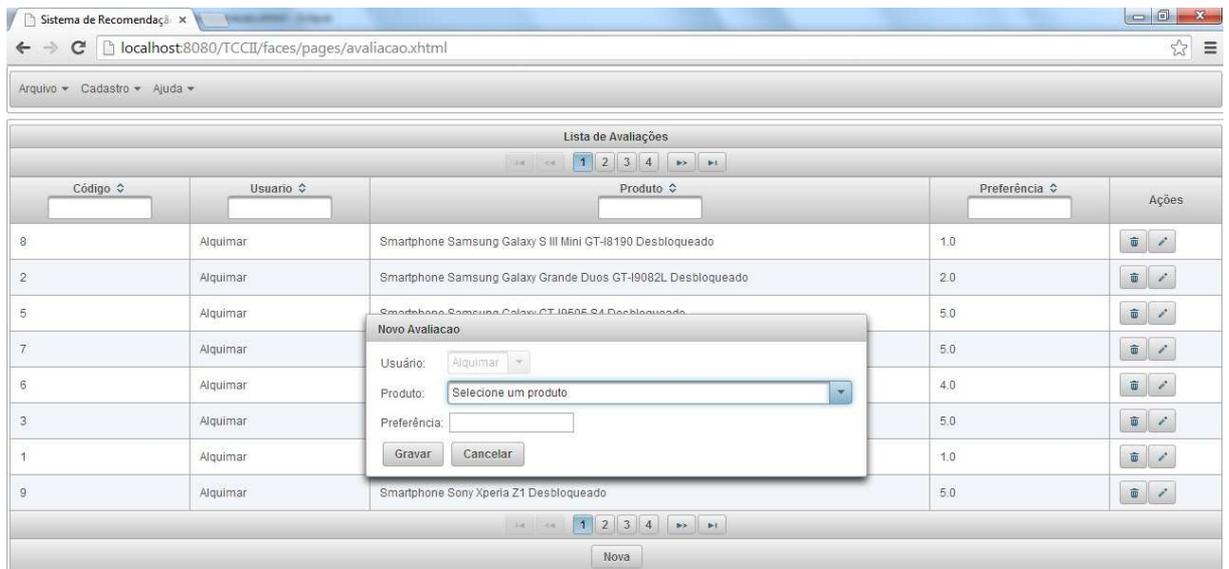
A Figura 8 demonstra a tela de avaliações, onde os usuários, além de visualizar as avaliações, podem avaliar os produtos de seu interesse.

Código	Usuario	Produto	Preferência	Ações
8	Alquimar	Smartphone Samsung Galaxy S III Mini GT-I8190 Desbloqueado	1.0	[Delete] [Edit]
2	Alquimar	Smartphone Samsung Galaxy Grande Duos GT-I9082L Desbloqueado	2.0	[Delete] [Edit]
5	Alquimar	Smartphone Samsung Galaxy GT-I9505 S4 Desbloqueado	5.0	[Delete] [Edit]
7	Alquimar	Smartphone Samsung Galaxy Win Duos GT-I8552 Desbloqueado	5.0	[Delete] [Edit]
6	Alquimar	Smartphone Samsung Galaxy S4 Mini GT-I9192 Desbloqueado	4.0	[Delete] [Edit]
3	Alquimar	Smartphone LG G2 D805 Desbloqueado	5.0	[Delete] [Edit]
1	Alquimar	Smartphone Samsung Galaxy Note III GT-N9005 Desbloqueado	1.0	[Delete] [Edit]
9	Alquimar	Smartphone Sony Xperia Z1 Desbloqueado	5.0	[Delete] [Edit]

**Figura 8** – Tela de avaliações.

Como pode ser observado na Figura 8, a ferramenta permite que o usuário tenha acesso às avaliações existentes. Além de visualizar as avaliações, o usuário também pode avaliar novos produtos, alterar avaliações existentes e excluir avaliações.

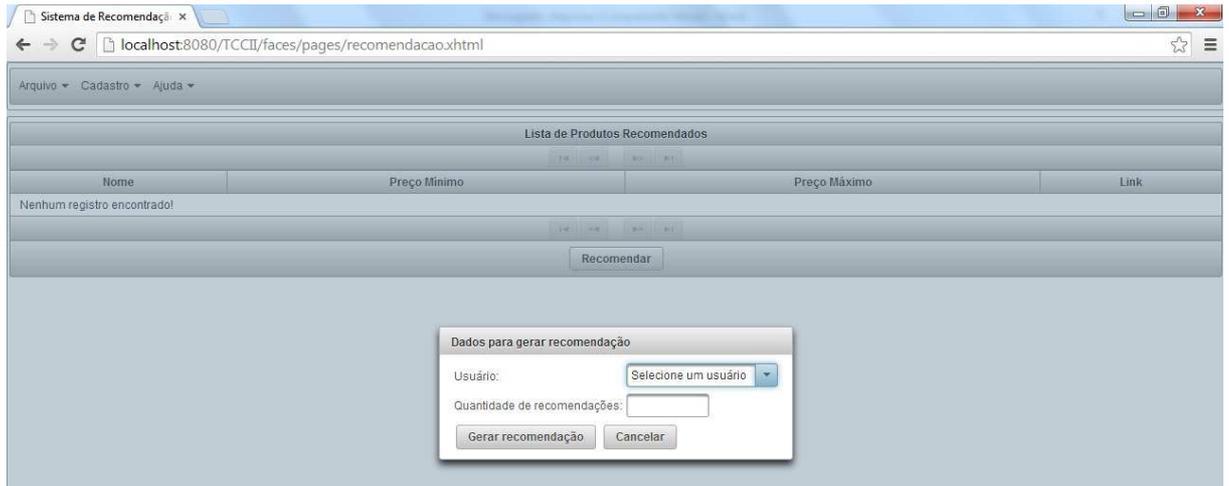
A Figura 9 demonstra o formulário utilizado para realizar a avaliação de um produto.



**Figura 9** – Formulário de avaliação.

Como pode ser observado na Figura 9, o formulário utilizado para avaliar os produtos necessita de três informações: o usuário que está avaliando (o sistema detecta o usuário da sessão), o usuário seleciona o produto que deseja avaliar e digita um número (inteiro) que representa sua preferência pelo produto que está sendo avaliado. Após informar os dados necessários, o usuário confirma a ação clicando no botão gravar.

A Figura 10 demonstra o formulário responsável por gerar as recomendações.



**Figura 10** – Formulário para gerar as recomendações.

Como pode ser observado na Figura 10, o formulário utilizado para gerar as recomendações necessita de duas informações: o usuário alvo das recomendações e a quantidade máxima de recomendações que se deseja obter.

A Figura 11 demonstra uma lista de produtos recomendados.

Nome	Preço Mínimo	Preço Máximo	Link
Smartphone Samsung Galaxy S III Mini GT-I8190 Desbloqueado	548,61	849,00	<a href="http://compare.buscape.com.br/prod_unico?idu=477908&amp;mdsrc=26380199&amp;mdapp=100&amp;mddtn=0">http://compare.buscape.com.br/prod_unico?idu=477908&amp;mdsrc=26380199&amp;mdapp=100&amp;mddtn=0</a>
Smartphone Samsung Galaxy Grande Duos GT-I9082L Desbloqueado	764,10	1199,00	<a href="http://compare.buscape.com.br/prod_unico?idu=492103&amp;mdsrc=26380199&amp;mdapp=100&amp;mddtn=0">http://compare.buscape.com.br/prod_unico?idu=492103&amp;mdsrc=26380199&amp;mdapp=100&amp;mddtn=0</a>
Smartphone Samsung Galaxy GT-I9505 S4 Desbloqueado	1399,90	2365,52	<a href="http://compare.buscape.com.br/prod_unico?idu=516076&amp;mdsrc=26380199&amp;mdapp=100&amp;mddtn=0">http://compare.buscape.com.br/prod_unico?idu=516076&amp;mdsrc=26380199&amp;mdapp=100&amp;mddtn=0</a>

**Figura 11** – Lista de produtos recomendados.

Como pode ser observado na Figura 11, após solicitar uma recomendação e informar os dados necessários, o sistema exibe a lista de produtos recomendados para o usuário alvo. As recomendações são geradas pela classe Recomendação, apresentada na seção seguinte.

As seções seguintes são responsáveis por mostrar e detalhar os artefatos gerados no desenvolvimento da ferramenta.

### 3.3.1 Classe Recomendação

Esta é a classe principal da ferramenta, pois é através dela que o sistema consegue analisar os dados a respeito dos usuários, definindo o perfil de cada um, para em seguida, calcular a similaridade entre um usuário e seus vizinhos (através de um algoritmo de correlação de usuário) e, por fim, gerar as recomendações para um usuário específico. A Figura 12 demonstra a classe utilizada para realizar as recomendações, de acordo com os padrões definidos pela API *Apache Mahout Taste*.

```

16 import java.io.File;
17 public class Recomendacao {
18     public static void main(String[] args) {
19         //Interface implementada pelas classes MySQLJDBCDataModel e FileDataModel
20         DataModel base = null;
21         lerJSON teste = new lerJSON();
22         Item item = new Item();
23         try {
24             base = new FileDataModel(new File(
25                 "C:\\Desenvolvimento\\Workspace\\TCC\\ml-100k\\base.csv"));
26
27             //Interface implementada pela classe PearsonCorrelationSimilarity
28             //Algoritmo de correlação de usuário, identifica semelhança entre os perfis dos usuários
29             UserSimilarity usuariosSemelhantes = new PearsonCorrelationSimilarity(base);
30
31             //Interface implementada pela classe NearestUserNeighborhood
32             //Determina os perfis semelhantes na vizinhança, recebe a quantidade limite e o algoritmo
33             UserNeighborhood vizinhanca = new NearestUserNeighborhood(3, usuariosSemelhantes, base);
34
35             //Interface implementada pela classe GenericUserBasedRecommender
36             //Recebe como parâmetro a base de dados, a vizinhança e o algoritmo de correlação
37             Recommender recomendacao = new GenericUserBasedRecommender(base, vizinhanca, usuariosSemelhantes);
38
39             //Método responsável por gerar as recomendações, recebe o usuário alvo e a quantidade máxima de recomendações
40             List<RecommendedItem> recomendacoes = recomendacao.recommend(2, 10);
41             for (RecommendedItem recommendedItem : recomendacoes) {
42                 System.out.println("Item recomendado: " + recommendedItem.getItemID() + " - Avaliação: " + recommendedItem.getValue());
43                 item = teste.getProduto(recommendedItem.getItemID());
44             }
45             System.out.println("Quantidade de recomendações: " + recomendacoes.size());
46         } catch (IOException ex) {
47             ex.printStackTrace();
48         } catch (TasteException e) {
49             e.printStackTrace();
50         }
51     }
52 }

```

Figura 12 – Classe responsável pelas recomendações.

Como pode ser observado na Figura 12, a classe responsável por gerar as recomendações utiliza 4 interfaces: *DataModel*, *UserSimilarity*, *UserNeighborhood*, *Recommender*, onde estas utilizam classes que as implementam bem como métodos que são utilizados para que as recomendações sejam geradas. Estas interfaces, métodos e classes serão detalhadas em seguida:

- A interface *DataModel* foi implementada pela classe *MySQLJDBCDataModel*, que é utilizada no caso de utilização de um banco de dados como fonte. Para tanto, é

necessário informar quais são as tabelas que serão utilizadas para gerar as recomendações, que no caso são: tabela de usuários, tabela de produtos e tabela de avaliação, como pode ser observado no final da linha 24.

- A interface *UserSimilarity* foi implementada pela classe *PearsonCorrelationSimilarity*, classe que implementa um algoritmo de correlação, responsável por identificar quais são os usuários que possuem correlação. Para tanto, é preciso especificar uma base de dados do tipo *DataModel*, como pode ser observado no final da linha 29.
- A interface *UserNeighborhood* foi implementada pela classe *NearestNUserNeighborhood*, classe responsável por identificar, na vizinhança, os usuários que possuem perfil semelhante ao usuário alvo. Para tanto, é preciso identificar a quantidade limite de vizinhos, especificar o algoritmo de correlação de usuário (do tipo *UserSimilarity*) e a base de dados do tipo *DataModel*, como pode ser observado no final da linha 33.
- A interface *Recommender* foi implementada pela classe *GenericUserBasedRecommender*, classe responsável por preparar a recomendação. Para tanto é preciso especificar a base de dados (tipo *DataModel*), os perfis semelhantes na vizinhança (do tipo *UserNeighborhood*) e o algoritmo de correlação (do tipo *UserSimilarity*), como pode ser observado no final da linha 37.
- Por fim, o método *recommend*, que é disponibilizado através da interface *Recommender*. Este método é responsável por gerar uma lista de recomendações. Para que as recomendações sejam geradas, é necessário especificar o id do usuário alvo e a quantidade máxima de recomendações que se deseja obter para o usuário alvo, como pode ser observado no final da linha 40.

### 3.3.1 Classe LerJSON

Esta classe é importante para que a ferramenta funcione conforme a proposta deste trabalho, pois através desta classe é possível ler um arquivo do tipo *json* que contém uma lista de produtos da base de dados do Buscapé. Através deste arquivo é possível percorrer a lista de produtos retornados do buscapé e recomendar os produtos que estão de acordo com os itens selecionados na recomendação. Este arquivo é obtido através da API do Buscapé, que disponibiliza métodos para retornar os produtos de acordo a necessidade da aplicação. A Figura 13 demonstra a utilização da classe *LerJSON*, que permite ao sistema, reconhecer e percorrer uma lista de produtos em um arquivo no formato *json*.



```

lerJSON.java | Item.java | Recomendacao.java
1 import java.io.FileNotFoundException;
11
12 public class LerJSON {
13     public Item getProduto(Long id) {
14         JSONObject jsonObject;
15         JSONParser parser = new JSONParser();
16         Item item = new Item();
17         try {
18             jsonObject = (JSONObject) parser
19                 .parse(new FileReader(
20                     "C:\\Desenvolvimento\\Workspace\\TCC\\ml-100k\\dados.json"));
21
22             JSONArray completo = (JSONArray) jsonObject.get("product");
23             for (int i = 0; i < completo.size(); i++) {
24                 JSONObject temp = (JSONObject) completo.get(i);
25                 JSONObject produto = (JSONObject) temp.get("product");
26
27                 if (produto.get("id").toString().equals(id.toString())) {
28                     item.setId(Long.parseLong(produto.get("id").toString()));
29                     item.setNome((String) produto.get("productname"));
30                     item.setPrecoMin(Double.parseDouble((String) produto
31                         .get("pricemin")));
32                     item.setPrecoMax(Double.parseDouble((String) produto
33                         .get("pricemax")));
34
35                     System.out.println(item);
36                 }
37             }
38         } catch (FileNotFoundException e) {
39             e.printStackTrace();
40         } catch (IOException e) {
41             e.printStackTrace();
42         } catch (ParseException e) {
43             e.printStackTrace();
44         }
45         return item;
46     }
47 }

```

**Figura 13** – Classe responsável por ler arquivos no formato *json*

Como pode ser observado na Figura 13, a classe responsável por ler arquivos no formato *json* utiliza duas classes: *JSONObject* (recebe objetos do tipo *json*) e *JSONParser* (transforma o arquivo *json* em *JSONObject*). Estas classes serão detalhadas em seguida:

- Na linha 18, a variável *jsonObject* está recebendo um objeto do tipo *JSONObject*. Porém, é preciso que o arquivo passe por um conversor, que no caso é o parser, que instancia um objeto do tipo *FileReader* e o transforma em *JSONObject*.
- Na linha 22, é criado um objeto do tipo *JSONArray*, que é uma lista de *JSONObject*, que armazena todos os produtos retornados do arquivo que contém a lista de produtos do Buscapé, através do método *get()*, que recebe como parâmetro a chave desejada, que no caso foi “product”.
- As linhas 23 – 36 são responsáveis por percorrer toda lista de objetos json e verificar se o id do produto local é igual ao id do produto recomendado, em caso positivo os dados desse produto são armazenados em uma variável “item”, para que seja usada posteriormente para apresentação. Tudo isso faz parte do método *getProduto()*, da classe *LerJSON*, que recebe como parâmetro o id de um produto da lista de produtos recomendados pela classe de recomendação.

## CONSIDERAÇÕES FINAIS

Neste trabalho foram abordados os conceitos sobre Sistemas de Recomendação, Recuperação da Informação, Técnicas de Recomendação e Técnicas de Filtragem. O estudo e compreensão destes conceitos foram fundamentais para que fosse possível realizar a definição e desenvolvimento da ferramenta proposta neste trabalho. Em relação às técnicas de recomendação, neste trabalho foi utilizada a técnica de Filtragem Colaborativa (FC). Em relação aos produtos utilizados para avaliação e recomendação, neste trabalho foi utilizada a base de produtos do Buscapé, o que foi possível através da utilização da API do Buscapé.

Para desenvolver a ferramenta foi necessária a implementação da técnica de Filtragem Colaborativa (FC). Esta técnica foi implementada com o apoio da API *Apache Mahout Taste*, que possibilitou que essa técnica fosse implementada de forma mais rápida e prática, porém, foi necessário um estudo aprofundado sobre a funcionalidade e utilização desta API.

A ferramenta desenvolvida recomenda ao usuário alvo novos produtos, sendo que estes são produtos que foram consumidos por seus vizinhos, os quais possuem perfil semelhante. A lista de produtos retornados possui um diferencial na ordenação, onde se utiliza o preço como base para ordenar a lista, ou seja, os produtos são apresentados do menor para o maior preço.

Como trabalho futuro, seria interessante, além da técnica de Filtragem Colaborativa (FC), utilizar mais uma técnica de filtragem, que no caso poderia ser a técnica de Filtragem Baseada em Conteúdo (FBC), pois esta permite que os itens recomendados estejam de acordo com as preferências do usuário alvo. Assim, através da técnica FC seria possível obter novos produtos e com a utilização da FBC, verificar quais desses produtos estão de acordo com as preferências do usuário alvo. Outra proposta seria implantar a ferramenta em um

ambiente real, para verificar o seu funcionamento e analisar as recomendações feitas por ela, do ponto de vista qualitativo e eficiente.

## REFERÊNCIAS BIBLIOGRÁFICAS

BORGES, D. M. **Desenvolvimento de uma ferramenta de recomendação automática de produtos**. 2011. 89p. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) – Centro Universitário Luterano de Palmas, Palmas – TO.

BRESOLIN, A. **Desenvolvimento de um Sistema de Recomendação Fazendo uso de Técnicas de Mineração de Dados**. Disponível em: <<http://www.unochapeco.edu.br/static/files/trabalhos-anais/Pesquisa/Ci%C3%A2ncia%20e%20Tecnologia/Ademir%20Bresolin.pdf>>. Acesso em: 08 de ago. 2013.

BUSCAPE, Página oficial. Disponível em: <<http://develAoper.buscape.com/api/>>. Acesso em: 22 de ago. 2013.

CARDOSO, O. N. P. **Recuperação de Informação**. 6p. Universidade Federal de Lavras – MG. Disponível em: <<http://www.dcc.ufla.br/infocomp/artigos/v2.1/art07.pdf>>. Acesso em: 15 de out. 2013.

CASTRO, S. S. G. F; BARBOSA, T. M. **RAPIDos: Recomendação Automática de Produtos Interessantes em Dispositivos Móveis**. 2009. 61p. TCC (Graduação em Ciência da Computação) – Universidade de Brasília. Disponível em: <[http://monografias.cic.unb.br/dspace/bitstream/123456789/191/1/Monografia\\_Saulo\\_Tiago\\_FINAL.pdf](http://monografias.cic.unb.br/dspace/bitstream/123456789/191/1/Monografia_Saulo_Tiago_FINAL.pdf)>. Acesso em: 05 de nov. 2013.

CAZELLA, S. C. **Aplicando a Relevância da Opinião de Usuários em Sistema de Recomendação para Pesquisadores**. 2006. 180p. Tese (Doutorado em Ciências da Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/8424/000575704.pdf?sequence=1>>. Acesso em: 23 de set. 2013.

CAZELLA, S. C.; DRUMM, J. V; BARBOSA, J. L. V. **Um serviço para recomendação de artigos científicos baseado em filtragem de conteúdo aplicado a dispositivos móveis.** *Novas Tecnologias na Educação*, v. 8, n. 3, p. 1-11, dez. 2010.

FERNEDA, E. **Recuperação de Informação: Análise sobre a contribuição da Ciência da Computação para a Ciência da Informação.** 2003. 147p. São Paulo - SP Disponível em: <[http://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CEgQFjAC&url=http%3A%2F%2Fwww.teses.usp.br%2Fteses%2Fdisponiveis%2F27%2F27143%2Ftde-15032004-130230%2Fpublico%2FTese.pdf&ei=X98MUtfHLW44AP61ICQBw&usg=AFQjCNG0CJp\\_yV8WJf\\_KhoeaHBMm\\_happw&sig2=TR0wRJ0BMm0E5kRcKn-8uQ&bvm=bv.50723672,d.dmg&cad=rja](http://www.google.com.br/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CEgQFjAC&url=http%3A%2F%2Fwww.teses.usp.br%2Fteses%2Fdisponiveis%2F27%2F27143%2Ftde-15032004-130230%2Fpublico%2FTese.pdf&ei=X98MUtfHLW44AP61ICQBw&usg=AFQjCNG0CJp_yV8WJf_KhoeaHBMm_happw&sig2=TR0wRJ0BMm0E5kRcKn-8uQ&bvm=bv.50723672,d.dmg&cad=rja)>. Acesso em: 15 de ago. 2013.

LÁZARO, A. S. **Análise e seleção de algoritmos de filtragem de informação para solução do problema cold-start item.** 2010. 65p. Monografia (Graduação em Sistemas de Informação) – Universidade Federal de Lavras, Lavras - MG. Disponível em: <[http://www.bsi.ufla.br/wp-content/uploads/2013/07/Monografia\\_Alexsandra.pdf](http://www.bsi.ufla.br/wp-content/uploads/2013/07/Monografia_Alexsandra.pdf)>. Acesso em: 23 de set. 2013.

LIMA, M. R.; MATOS LIMA, E. H.; SILVA, E. C. **Reflexos da cibercultura: a democratização do acesso à memória histórica documental.** 2010. 11p. Universidade Federal do Rio de Janeiro – UFRJ, Rio de Janeiro – RJ. Disponível em: <<http://www.marcinholima.com.br/artigos/abciber2010x.pdf>>. Acesso em: 21 de nov. 2013.

