UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ

Suelen Macedo Laurindo
Patrícia Bordignon André

UMA ARQUITETURA DE SERVIÇOS VOLTADA À RECUPERAÇÃO DE INFORMAÇÃO E RECOMENDAÇÃO DE CONTEÚDO

Trabalho de Conclusão de Curso submetido à Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharel em Tecnologias da Informação e Comunicação.
Orientador: Prof. Dr. Alexandre Leopoldo Gonçalves.

Araranguá
2014

André, Patrícia Bordignon

Uma arquitetura de serviços voltada à recuperação da informação e recomendação de conteúdo / Patrícia Bordignon André, Suelen Macedo Laurindo ; orientador Alexandre Leopoldo Gonçalves. – Araranguá, SC, 2014.
77 p.

Trabalho de Conclusão de Curso (graduação em Tecnologias da Informação e Comunicação) – Universidade Federal de Santa Catarina, Campus Araranguá.

1. Sistemas de Recuperação da Informação. 2. Recuperação da Informação. 3. Web 2.0. I. Laurindo, Suelen Macedo. II. Gonçalves, Alexandre Leopoldo. III. Universidade Federal de Santa Catarina. Campus Araranguá. IV. Título.

Catalogação na fonte elaborada por: Débora Maria Russiano Pereira, CRB-14/1125

Suelen Macedo Laurindo
Patrícia Bordignon André

UMA ARQUITETURA DE SERVIÇOS VOLTADA A RECUPERAÇÃO DE INFORMAÇÃO E RECOMENDAÇÃO DE CONTEÚDO

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de "Bacharel em Tecnologias da Informação e Comunicação", e aprovado em sua forma final pelo Curso de Graduação em Tecnologias da Informação e Comunicação.

Araranguá, 07 de Julho de 2014.

Prof. Vilson Gruber, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Alexandre Leopoldo Gonçalves, Dr. (Orientador)

Prof. Juarez Bento da Silva, Dr.

Prof. Olga Yevseyeva, Dra.

"A maioria dos homens prefere negar
uma verdade dura do que enfrentá-la."

George R.R. Martin

AGRADECIMENTOS

Primeiramente, a Deus por ter me dado fé e coragem que foram muito importantes durante toda esta caminhada.

Além de agradecer, dedico este trabalho aos meus pais José Justino Laurindo e Fátima Terezinha de Macedo Laurindo, pelo incentivo, pela confiança e pelos exemplos de força e dedicação que sempre me deram.

À todos os professores que durante a graduação me proporcionaram um aprendizado para a realização deste trabalho, principalmente ao nosso orientador Dr. Alexandre Leopoldo Gonçalves por estar sempre disposto a nos ajudar e guiar durante o desenvolvimento deste trabalho.

À Patrícia, minha amiga de todas as horas e parceira nesta jornada.

À meu namorado por seu amor e compreensão durante muitos momentos em que estive ausente.

À todos os amigos e familiares que contribuíram para pudéssemos concluir este trabalho.

AGRADECIMENTOS

Agradeço acima de tudo a Deus pela dádiva da vida, e por sempre me guiar durante toda esta jornada.

À minha mãe, pela compreensão, incentivo e apoio nos momentos de dificuldades.

Aos meus professores que durante a graduação me proporcionaram um aprendizado para a realização deste trabalho, principalmente ao nosso orientador Prof. Dr. Alexandre Leopoldo Gonçalves.

À minha amiga e fiel companheira desta jornada, Suelen e a sua família.

Agradeço também aos meus amigos e familiares que contribuíram no decorrer desta caminhada e compreenderam minha falta de tempo.

RESUMO

O avanço da Web tem promovido novas formas de interação por parte de usuários resultando no aumento do volume de informações, bem como em uma maior dificuldade na tomada de decisão. Os usuários passam a assumir um papel de produtores de conteúdo ao invés de simplesmente consumidores. Este cenário gera oportunidades na construção de sistemas capazes de auxiliar usuários em suas escolhas ou mesmo aprender mais sobre um domínio. Na área de Recuperação de Informação opiniões podem resultar em sistemas mais interativos e de maior valor agregado sugerindo documentos que sejam mais relacionados a determinado perfil de usuário. Este trabalho propõe uma arquitetura de serviços com foco na Recuperação de Informação e em Sistemas de Recomendação como forma de melhorar a interatividade e a localização de documentos que sejam de interesse de determinado usuário. A implementação e aplicação da arquitetura em um cenário de uso permitiu analisar a interconexão de todos os serviços e demonstrar, através de consultas as recomendações obtidas a partir da escolha de um documento em particular. Por fim, foi possível perceber que a elaboração de sistemas que levem em consideração informações obtidas a partir da interação de usuários tende a facilitar a localização de documentos de interesse além daqueles fornecidos em resposta a uma busca.

Palavras-chave: Sistemas de Recomendação; Recuperação de Informação; Web 2.0.

ABSTRACT

The Web has fostered new ways of interaction by users resulting in increased volume of information, as well as greater difficulty in making decisions. Users shall assume the role of content producers rather than simply consumers. This scenario creates opportunities to build systems capable of helping users in their choice or even learn more about a domain. In the area of Information Retrieval opinions can result in more interactive systems and with higher value-added suggesting documents that they are more related to a particular user profile. This paper proposes a service architecture with focus on Information Retrieval and Recommender Systems in order to improve the interactivity and location of documents that are of interest to a particular user. The implementation and application of the architecture in a usage scenario allowed us to analyze the interconnection of all services and demonstrate, through consultation the recommendations derived from the choice of a particular document. Finally, it was noted that the development of systems that take into account information obtained from the interaction of users tends to facilitate documents locating of interest beyond those provided in response to a search.

Keywords: Information Retrieval; Recommender Systems; Web 2.0.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface
AJAX - Asynchronous JavaScript and XML
BCPL - Basic Combined Programming Language
CERN - Conseil Européen pour la Recherche Nucléaire
FBC - Filtragem Baseada em Conteúdo
FC - Filtragem Colaborativa
GWT - Google Web Toolkit
HTML - HyperText Markup Language
HTTP - Hypertext Transfer Protocol
IDE - Integrated Development Environment
JDBC - Java Database Connectivity
JSON - JavaScript Object Notation
OSGI - Open Services Gateway Initiative
PDF - Portable Document Format
RI – Recuperação de Informação
RSS – Really Simple Syndication
RT – Relação de Termos
SGBD – Sistema Gerenciador de Banco de Dados
SMART – System for the Manipulation and Retrieval of Text
SR – Sistemas de Recomendação
SRI – Sistemas de Recuperação de Informação
TE - Termos Específicos
TG - Termos Genéricos
URL - Uniform Resource Locator
XML - eXtended Markup Language

SUMÁRIO

1 INTRODUÇÃO

A Web teve início na Organização Europeia para a Pesquisa Nuclear (*Conseil Européen pour la Recherche Nucléaire* - CERN), quando Tim Berners-Lee observou as dificuldades para o gerenciamento de informações que esta possuía. A organização das informações que o CERN apresentava se assemelhava a uma "teia de aranha" com conexões múltiplas, onde as interconexões progrediam com o tempo (BERNERS-LEE, 1989).

Diante desta situação Tim Berners-Lee idealizou uma organização dinâmica, onde as informações que eram geradas em lugares distintos poderiam ser conectadas. Desta maneira, caso uma pessoa deixasse o CERN, as informações geradas por esta não seriam perdidas. A solução que Berners-Lee propôs a este problema foi a criação de um sistema distribuído por meio de hipertexto (BERNERS-LEE, 1989).

Quando Tim Berners-Lee criou a *Web* a definiu como um universo de informação acessível pela rede global, local onde ocorre interação entre as pessoas, e atualmente é composta por hiperlink, imagens e vídeos (BERNERS - LEE, 1996).

A primeira arquitetura da *World Wide Web* foi proposta em 1989 por Tim Berners-Lee e em 1990 ele desenvolveu o primeiro navegador chamado *WorldWideWeb*, que mais tarde foi renomeado para Nexus a fim de evitar confusão (BERNERS-LEE, 1996).

Esta geração da *Web* é conhecida por *Web* 1.0 e segundo Primo (2007), os sites apenas eram trabalhados como unidades isoladas e estáticas. Berners-Lee criou a *Web* unindo as seguintes tecnologias: URL (*Uniform Resource Locator*), utilizado pra localizar recursos na *Web*, HTML (*Hypertext Markup Language*), utilizada para representar o conteúdo em páginas na *Web* e HTTP (*Hypertex Transfer Protocol)*, protocolo utilizado para transferência de dados na *Web* (BERNERS-LEE, 1996).

Com o passar dos anos a Web evoluiu para uma visão mais interativa, dando início ao que ficou conhecido como *Web* 2.0, onde o usuário passou a ser ativo não apenas consumindo a informação, mas também produzindo-a. Esse comportamento foi definido com o termo *Prosumer*, cunhado por Alvin Toffler em 1980 (TOFFLER, 1980).

Segundo Primo (2007), a *Web* 2.0 é a segunda geração de serviços online. O termo *Web* 2.0 foi definido por Dale Dougherty em 2004 (O'REILLY, 2007).

Além de mudanças conceituais, também ocorreram mudanças tecnológicas e adesão de novas ferramentas de desenvolvimento como: AJAX (*Asynchronous JavaScript and XML*), Flex (*Adobe Flex*), GWT (*Google Web Toolkit*), que permitem a construção de páginas dinâmicas (PRIMO, 2007).

O novo conceito de *Web* a torna bidirecional, ou seja, uma *Web* participativa com leitura e escrita de conteúdo, potencializando as formas de comunicação, onde o usuário deixa de ser apenas o receptor e passa a interagir contribuindo com conteúdo. Este conteúdo deixa de apresentar um padrão estático, passando a ser dinâmico (AGHAEI; NEMATBAKHSH; FARSANI, 2012).

A *Web* 2.0 trouxe consigo um novo paradigma e também repercussões sociais, difundindo a produção e circulação de informações, vista como uma nova plataforma que viabiliza funções online, aperfeiçoando a usabilidade e aprimorando o conceito de "arquitetura de participação". Essa arquitetura visa oferecer serviços, tais como: publicação em espaço de debate, gestão coletiva de trabalho, negociação coletiva, e interação social (PRIMO, 2007).

Dentre os principais serviços da *Web* 2.0 podemos destacar os *Blogs*, o RSS (*Really Simple Syndication*), os *Wikis*, os *Mashups*. Estes foram os mecanismos que facilitaram e induziram o crescimento de usuários ativos na *Web* (AGHAEI; NEMATBAKHSH; FARSANI, 2012). O crescimento exponencial de informação gerada pela *Web* 2.0 trouxe como consequência uma grande diversidade de conteúdo a disposição do usuário, tornando-se necessário recursos que ajudem usuários a realizarem consultas e escolhas. Entre as áreas que promovem suporte encontram-se a Recuperação de Informação (RI) e os Sistemas de Recomendação (SR).

Segundo Manning, Raghavan e Schütze (2009), a Recuperação de Informação (RI) consiste na localização de materiais de natureza não estruturada que satisfazem determinada necessidade por informação a partir de uma fonte de informação.

O processo de recuperação de informação possui como objetivo encontrar em uma coleção de documentos (*corpus*) quais satisfazem a busca de informação do usuário (FERNEDA, 2003). A Recuperação da Informação abrange tecnologias de consulta e indexação e está fundamentada na análise e disponibilização automática de conteúdo, normalmente documentos textuais (FOLTZ, 92; HERLOCKER, 2000).

Para que as buscas realizadas pelos usuários sejam mais eficientes são utilizadas algumas estratégias, estas se baseiam nos modelos de

recuperação de informação. Os modelos mais comuns são: booleano e vetorial.

O modelo booleano é o modelo mais simples, está baseado na teoria dos conjuntos, em que as consultas podem utilizar os conectores lógicos *AND*, *OR* e *NOT*. Entretanto, este modelo não possui ordenação para o resultado apresentado. Por outro lado, o modelo vetorial é um modelo estatístico em que o documento é representado através de uma lista de termo (um vetor). Diferentemente do modelo booleano, apresenta ordenação para o resultado baseada no grau de similaridade entre os documentos relevantes para a busca (SOUZA, 2006).

Além das estratégias de busca pode ser adequado um direcionamento para que o usuário saiba quais dos resultados listados melhor satisfazem a sua necessidade. Para tal, Sistemas de Recomendação podem proporcionar informações mais apuradas que complementam determinada consulta de usuário.

Os Sistemas de Recomendação são ferramentas de *software* e técnicas que proporcionam sugestões de itens para serem utilizados pelo usuário (RICCI et al., 2011). Têm por objetivo a redução da sobrecarga de conteúdo, através da seleção de informações baseada em prioridades do usuário (FIGUEIRA FILHO; GEUS; ALBUQUERQUE, 2008). As sugestões referem-se a vários processos de tomada de decisão, tais como os itens de compra, que tipo de música ouvir, ou que notícias ler.

O primeiro Sistema de Recomendação comercial foi chamado de *Tapestry* e utilizava o conceito de filtragem colaborativa (GOLDBERG et al., 1992).

Atualmente os SR têm sido classificados principalmente em três abordagens: sendo a primeira a baseada na comparação de conteúdo ou características de determinado item. A segunda é a abordagem colaborativa (CAZELLA; NUNES; REATEGUI, 2010). E a terceira abordagem é híbrida unindo as duas abordagens anteriores (FIGUEIRA FILHO; GEUS; ALBUQUERQUE, 2008).

1.1 PROBLEMÁTICA

A *Web* 2.0 permitiu que os usuários agregassem novos conteúdos e novos sites com facilidade, possibilitando a conexão destes conteúdos com outras páginas por meio de hiperlinks (O'REILLY, 2007). De modo geral, o atual cenário faz como que usuários assumam um papel proativo na disponibilização de informação, promovendo assim uma explosão de conteúdo na *Web*. O aumento na quantidade de informações afeta

principalmente a capacidade de escolha e, portanto, de tomada de decisão dos usuários.

Neste contexto, as áreas Recuperação de Informação e Sistemas de Recomendação são essenciais para prover sistemas capazes de lidar adequadamente com o volume de informações e com a recomendação de conteúdos de maior interesse por parte dos usuários. Enquanto que a Recuperação de Informação têm como objetivo encontrar em uma coleção de documentos, quais destes satisfazem a busca de informação do usuário (FERNEDA, 2003), os Sistemas de Recomendação possuem como objetivo selecionar o conteúdo com base nas prioridades do usuário (FIGUEIRA FILHO; GEUS; ALBUQUERQUE, 2008).

O desenvolvimento, considerando tais desafios, objetiva facilitar o processo de localização de informações, não apenas pela análise do texto completo, mas também pela sugestão de conteúdo de interesse levando em consideração a experiência de outros usuários.

A partir do exposto acima, este trabalho possui como pergunta de pesquisa "Como projetar uma arquitetura de *software* que auxilie o usuário na localização, análise e escolha de documentos textuais de seu interesse?".

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Este trabalho possui como objetivo geral a proposição de uma arquitetura de Recuperação de Informação incrementada pelo conceito de recomendação de conteúdo.

1.2.2 Objetivos Específicos

Visando atingir o objetivo principal, alguns objetivos específicos são requeridos, entre eles:

- Realizar um levantamento bibliográfico sobre as áreas de pesquisa envolvidas no trabalho, sendo estas, Recuperação de Informação e Sistemas de Recomendação;
- Propor uma arquitetura de *software* baseada em serviços que promova suporte a Recuperação de Informação e Sistemas de Recomendação;

- Desenvolver um protótipo que, através de serviços, possibilite a recuperação e sugestão de conteúdo textual;
- Realizar uma discussão dos resultados obtidos através da utilização do protótipo.

1.3 METODOLOGIA

O trabalho será realizado através de uma pesquisa exploratória e tecnológica com o desenvolvimento de um protótipo que permita a recuperação e sugestão de conteúdo textual. Para atingir os objetivos o trabalho foi dividido nas seguintes etapas:

- Levantamento da bibliografia com foco nas áreas de Sistemas de Recuperação de Informação e Sistemas de Recomendação;
- Análise das tecnologias que podem ser utilizadas no desenvolvimento da arquitetura proposta;
- Prototipação de um ambiente que torne possível a elaboração de cenários de utilização da arquitetura;
- Proposição de um cenário para uso e testes do protótipo;
- Análise dos resultados obtidos por meio da utilização da arquitetura de recuperação e recomendação de informação.

1.4 ORGANIZAÇÃO DO TEXTO

O documento está organizado em seis capítulos. Este primeiro capítulo realiza uma introdução e contextualização da problemática, dos objetivos e da metodologia utilizada.

No segundo capítulo é realizada uma revisão da área de Recuperação da Informação, apresentando uma síntese sobre a *Web*, o processo de indexação e modelos utilizados para a recuperação de informação.

O terceiro capítulo aborda a área de Sistemas de Recomendação, com um breve histórico, aplicações e as principais abordagens utilizadas.

No quarto capítulo é realizada a proposição de uma arquitetura de serviços, com foco na Recuperação de Informação incrementada pelo conceito de recomendação de conteúdo. Este capítulo divide-se em duas partes, uma descreve o modelo lógico da arquitetura e a segunda parte

descreve as tecnologias utilizadas e os serviços da arquitetura, ou seja, modelo físico.

No quinto capítulo são detalhados os serviços da arquitetura, apresentação de um cenário de uso e a discussão dos resultados obtidos. Finalizando, o sexto capítulo apresenta as considerações finais e os trabalhos futuros.

2 RECUPERAÇÃO DE INFORMAÇÃO

O termo Recuperação de Informação (*Information Retrieval*) foi definido por Calvin Mooers em 1951. Segundo Mooers (1951), a Recuperação de Informação se refere a características intelectivas da definição de informação e sua descrição para a busca, assim como técnicas ou máquinas que são utilizadas na operação.

Segundo Singhal (2001), o precursor dos Sistemas de Recuperação de Informação foi Gerard Salton, que em 1960 junto com seus alunos na Universidade de Harvard desenvolveu o sistema SMART. Posteriormente os estudos foram continuados na Universidade de Cornell.

De acordo com Manning, Raghavan e Schütze (2009), Recuperação de Informação (RI) consiste na localização de materiais de natureza não estruturada que satisfazem determinada necessidade por informação a partir de uma grande coleção.

A necessidade de recuperar informações textuais já existia muito antes da criação dos computadores, entretanto com o advento destes, foi possível o desenvolvimento da *Web* que nas últimas décadas obteve um crescimento exponencial. Diante disto, a tarefa de recuperar informações de uma maneira automatizada tornou-se uma necessidade (SINGHAL, 2001).

De modo geral, a *Web* se encontra dividida em duas categorias, sendo estas: *Web* Visível e *Web* Invisível (também denominadas de: *Web* Profunda, *Web* Oculta, *Dark Web* e *Deep Web*). A *Web* visível é aquela que permite a indexação de suas páginas, possibilitando a recuperação de informação. Já a *Web* invisível por sua vez não permite a indexação destas, devido a razões como: tecnologias utilizadas ou política (MONTEIRO; FIDENCIO, 2013).

Com o vasto número de páginas na *Web* surgiu a necessidade de organizar o conteúdo para permitir a recuperação deste. Um meio encontrado para realizar a organização desta foi através da indexação. A primeira forma de indexação foi a manual, se tornando ineficiente devido ao contínuo crescimento das páginas. Com uma visível necessidade de indexar de maneira eficiente, foram desenvolvidos mecanismos de busca (motores de busca ou buscadores) que criam os índices através de indexação mecanizada, utilizando algoritmos para a localização e indexação do conteúdo (MONTEIRO; FIDENCIO, 2013).

Os mecanismos de busca só realizam buscas na *Web* visível, porque os *crawlers* (indexadores automáticos, *web spiders* ou *Web robot*) não acessam páginas da *Web* invisível. Segundo Bergman (2001),

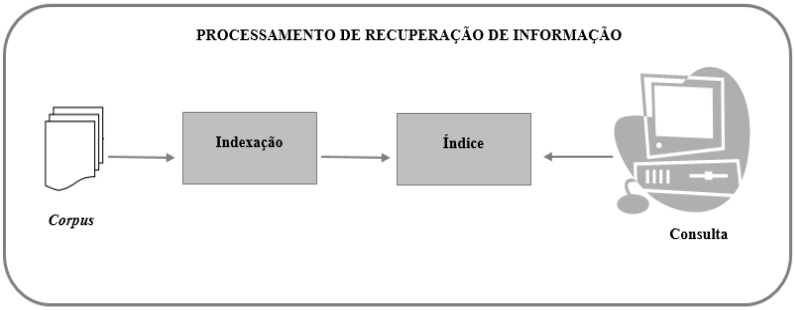
as informações públicas na *Deep Web* são 400 a 500 vezes maiores que as existentes na *web* visível.

Para que se possa buscar as informações que se encontram na *Web* visível foram desenvolvidos processos que realizam a recuperação de informação, denominados Sistemas de Recuperação de Informação (SRI).

O processo de recuperação de informação possui como objetivo encontrar, em uma coleção de documentos (*corpus*), quais satisfazem a busca de informação do usuário (FERNEDA, 2003, p.14). A Recuperação da Informação abrange tecnologias de consulta e indexação, e está fundamentada na análise automática de conteúdo, normalmente documentos textuais (HERLOCKER, 2000).

O método de recuperação de informação é executado através de uma consulta nas estruturas de dados (índice), estas que foram criadas por meio da indexação e retornam uma relação dos possíveis documentos que atendem a busca realizada, como ilustra a Figura 1.

Figura 1: Sistema de Recuperação de Informação.



Fonte: Adaptado de (CONCEIÇÃO, 2013).

2.1 PROCESSO DE INDEXAÇÃO

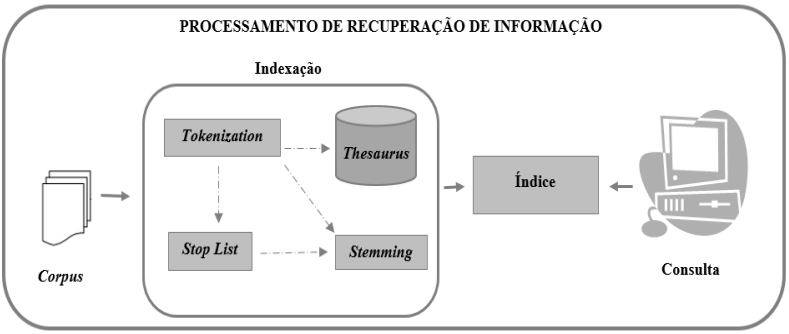
A indexação é um processo que analisa o conteúdo e sintetiza a informação relevante do documento, criando um vocábulo mediador entre o usuário e o documento (VIEIRA, 1988).

A indexação engloba o desenvolvimento de estruturas de dados relacionadas ao conjunto textual dos documentos, estas estruturas são compostas de dados referentes às características dos termos do *corpus*,

como por exemplo, a frequência com que cada termo ocorre (CARDOSO, 2000).

O processo de indexação é executado em etapas, sendo estas: extração de termos (*Tokenization*), lista de termos (*Stop List*), extração de radicais (*Stemming*) e aplicação de tesauro (*Thesaurus*), como ilustra a Figura 2.

Figura 2: Etapas da Indexação.



Fonte: Adaptado de (CONCEIÇÃO, 2013).

2.1.1 Extração de Termos

A etapa de extração de termos fundamenta-se em separar o texto em palavras (Termos/*Tokens*). Segundo Manning, Raghavan e Schütze (2009), dado uma sequência de caracteres e definido um documento, a extração de termos (*Tokenization*) é a tarefa de separar em *tokens* o documento textual, ignorando alguns caracteres, como a pontuação e o espaçamento. É necessário que o processo de extração de termos seja utilizado tanto no momento da indexação quanto ao realizar a busca (FOX, 2004).

Grande parte dos métodos de SRI utilizam algoritmos de busca de *Strings* para encontrar no índice os termos do documento que foram indexados. Partindo desde princípio, a extração de termos é uma etapa fundamental na recuperação de informação, visto que se um termo for buscado e este não estiver contido no índice, não ocorre a recuperação dos documentos relevantes para a busca. Um exemplo citado por Wu (2011), explica o funcionamento da seguinte forma, se o termo "*TradeOrPrice*" for indexado como um único *token* ao invés de "*Trade*

or Price", quando o usuário realizar a busca por "*Trade Price*" não terá como resposta um resultado satisfatório.

2.1.2 Lista de Termos

Esta etapa de Lista de Termos (*Stop List*) identifica um conjunto de palavras irrelevantes (*stop words*) constantes em uma lista de termos (*stop list*), na qual não alteram a essência do texto, que se repetem com frequência em um *corpus*, como artigos, conjunções, pronomes e preposições.

Uma pesquisa revelou que a utilização destas *stop words* em uma busca trará como retorno quase todos os itens de uma base de dados, sendo que essas palavras são responsáveis por 20 a 30 por cento dos *tokens* em um documento textual (SALTON; MCGILL, 1983; VAN RIJSBERGEN, 1975). Deixando estas palavras de fora durante a construção do índice o espaço utilizado por este será reduzido, além de tornar a busca mais eficiente (FOX, 1992). Por outro lado, esta estratégia pode reduzir a precisão na recuperação da informação (MANNING; RAGHAVAN; SCHÜTZE, 2009) uma vez que a semântica do documento é afetada de alguma maneira.

2.1.3 Raiz das Palavras

Esta etapa também é conhecida como *Stemming*, que segundo Manning, Raghavan e Schütze (2009), consiste em um processo heurístico onde é removido o sufixo e/ou prefixo para reduzir a palavra ao seu radical. Por exemplo, em um determinado documento textual constam as seguintes palavras: ESTUDANTE, ESTUDANTES, ESTUDAR, ESTUDANDO, ao aplicar o processo de extração de radicais nas palavras estas serão reduzidas ao seu radical: ESTUDA.

Esta técnica permite reduzir o número de entradas de termos no índice, assim como aumentar o número de vezes que o radical aparece no documento. A estratégia tende a prover uma melhor definição de pesos para o termo, já que se um termo ocorre x vezes no documento e o mesmo termo no plural ocorre mais y vezes, sem a aplicação da técnica *Stemming* existiriam os dois termos no índice. Com a aplicação da etapa de *Stemming* o termo será inserido uma única vez com x + y ocorrências no documento, com a possibilidade de melhor classificar determinado documento como relevante para uma busca (SILVA, 2009).

Segundo Manning, Raghavan e Schütze (2009), o algoritmo mais comum para a técnica *Stemming* é o algoritmo de Porter (1980). O algoritmo de Porter foi desenvolvido na linguagem BCPL (*Basic Combined Programming Language*) e é considerado de implementação simples, porém com bom desempenho computacional.

O algoritmo consiste em cinco fases de extração de radicais para a língua inglesa, pois é dependente do idioma utilizado. As fases são: exclusão de sufixos comuns, sufixos verbais, sufixo "i" se estiver antes da consoante "c", sufixos residuais e sufixos "e", "é" e "ê" (XAVIER; SILVA; GOMES, 2013).

2.1.4 Tesauro

Segundo Srinivasan (1992), tesauros (*Thesauri*) são estruturas importantes para os SRI, pois fornecem um vocabulário preciso e controlado utilizado para coordenar a indexação e/ou a recuperação dos documentos.

Os tesauros realizam um controle na inserção de termos no índice, através dos *tokens* escolhidos para representarem os conceitos, ou seja, termos relevantes na indexação, denominados de descritores. Também existem os termos não descritores, os quais não são utilizados na indexação, mas que ajudam o usuário na busca pelo documento.

Segundo Miranda (1990) os tesauros estabelecem alguns tipos de relações, tais como:

- Relação de Equivalência: ocorrem entre termos sinônimos, mas apenas um termo será o descritor.
- Relação Hierárquica: apresenta os níveis hierárquicos de superordenação e subordinação entre os conceitos, onde superordenação é o conceito mais abrangente (TG - Termos Genérico) e o subordinado é o conceito mais especifico (TE - Termos Especifico).
- Relação Associativa: ocorre quando o termo não é equivalente e nem hierárquico, mas existe uma associação mental e é necessário que exista essa relação de termos no Tesauro (RT - Relação de Termos).

2.2 MODELOS DE RECUPERAÇÃO DE INFORMAÇÃO

Os modelos de recuperação de informação são utilizados como estratégias para que as buscas realizadas pelos usuários sejam mais eficientes. Estes se dividem em duas categorias: clássicos e estruturados (SOUZA, 2006). Este trabalho abordará os dois modelos mais comuns dentro da categoria clássicos, que são: booleano e vetorial.

2.2.1 Modelo Booleano

O modelo booleano é o modelo mais simples, entretanto é a base de outros modelos como o vetorial. O modelo booleano está baseado na teoria dos conjuntos e na álgebra booleana, em que ao realizar uma consulta utilizando este modelo o usuário informa uma expressão lógica composta por operadores booleanos como: AND, OR e NOT. Através desta expressão o sistema retornará um conjunto de documentos que são relevantes para a busca (SOUZA, 2006). Deste modo, o sistema apenas classifica os documentos em dois conjuntos, aqueles que cumprem os requisitos da expressão realizada na consulta, que no modelo booleano é representado pelo valor "1" e aqueles que não cumprem os requisitos que são representados pelo valor "0", não apresentando nenhum tipo de ordenação.

O modelo booleano não registra a quantidade de vezes que cada termo foi referenciado no documento. Segundo Salton (1989), o funcionamento do modelo booleano ocorre da seguinte forma: existe uma lista invertida (índice) onde cada inserção refere-se a um termo da indexação, portanto, a inserção T_i é um ponteiro para uma lista de documentos em que o termo T_i existe. Os resultados serão obtidos através da intersecção dos índices dos documentos, ou seja, apenas termos que atendam aos critérios da consulta poderão ser recuperados.

O funcionamento do modelo booleano pode ser simulado na Tabela 1, que representa um índice invertido, onde as entradas de T_1, T_2, T_3,..., T_n representam os termos indexados e a coleção de documentos é representada por D_1, D_2,..., D_n. Ao realizar uma busca que tenha o objetivo de buscar documentos que estejam relacionados com os termos *Information Retrieval*, mas que não tenham relação com o termo *System*, será utilizada a seguinte expressão:

Information **AND** *Retrieval* **AND NOT** *System*

Considerando que T_1, representa o termo *Information*, T_2 representa *Retrieval* e T_3 corresponde a *System* o documento que satisfaz a busca será o D_1.

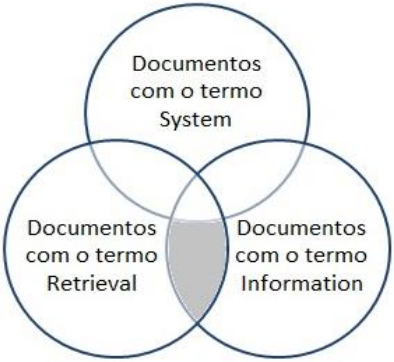
Tabela 1: Modelo Booleano.

	D_1,	D_2	...	D_n.
T_1	1	0	...	1
T_2	1	1	...	0
T_3	0	0	...	0
...	...	...	...	...
T_n	0	1	...	1

Fonte: Autores.

A expressão de busca pode ser representada na teoria dos conjuntos como ilustra a Figura 3, onde a intersecção está em cor cinza simbolizando os documentos que apresentam os termos *Information* e *Retrieval.*

Figura 3: Representação do Modelo Booleano.



Fonte: Autores.

De acordo com Souza (2006) o modelo booleano apresenta desvantagens por trabalhar com o sistema binário, ou seja, é relevante ou não relevante. Além disso, não é criado nenhum tipo de ordenação para listar os resultados por relevância para o usuário.

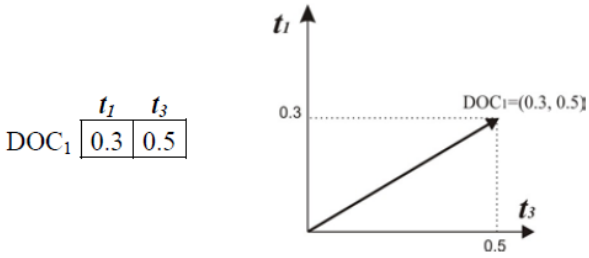
2.2.2 Modelo Vetorial

O modelo vetorial também é conhecido como espaço vetorial, e é caracterizado como um modelo estatístico, onde o documento é representado através de um vetor (Lista de Termos) no espaço *n*-dimensional, em que cada entrada do vetor representa um termo que possui um peso, ou relevância em um determinado documento (SOUZA, 2006).

Diferentemente do modelo booleano onde os pesos são binários identificando somente a presença (representado pelo valor "1") ou ausência (representado pelo valor "0") dos termos, o modelo vetorial atribui pesos que possui uma relevância entre a consulta do usuário e os documentos, classificada entre "0" e "1", sendo que quanto mais próxima de "1" maior é sua relevância. Essa forma de classificação é denominada de comparação parcial (KURAMOTO, 2002).

A Figura 4 ilustra um vetor representando um documento DOC_1 com dois termos T_1 e T_3, que possuem os respectivos pesos 0,3 e 0,5.

Figura 4: Representação vetorial de um documento com dois termos.

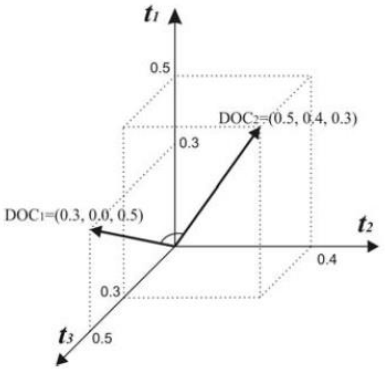


Fonte: FERNEDA, 2003.

A Figura 5 ilustra a representação vetorial de dois documentos DOC_1 e DOC_2 por três termos T_1, T_2 e T_3, sendo que para o DOC_1 foram atribuídos os pesos 0.3, 0.0 e 0.5, e para o DOC_2 foram atribuídos os pesos 0.5, 0.4 e 0.3.

Figura 5: Representação vetorial de uma busca em dois documentos com três termos.

	t_1	t_2	t_3
DOC_1	0.3	0.0	0.5
DOC_2	0.5	0.4	0.3



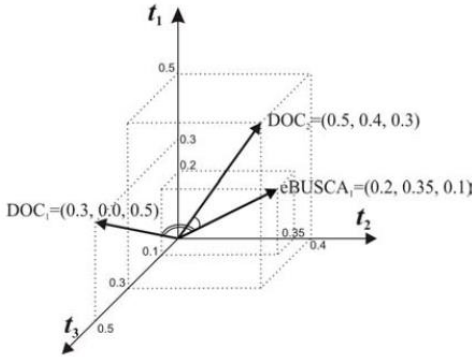
Fonte: FERNEDA, 2003.

Assim como os documentos, a busca também apresenta um peso de relevância atribuído aos termos da consulta (SALTON, 1988). A Figura 6 demonstra uma consulta com pesos tanto nos termos da busca quanto nos termos do documento. Os pesos atribuídos aos termos da busca foram (0.2, 0.35, 0.1) para os termos do DOC_1, foram atribuídos os pesos (0.3, 0.0, 0.5) e os pesos para o DOC_2 foram (0.5, 0.4, 0.3).

Figura 6: Representação vetorial de uma expressão de busca.

	t_1	t_2	t_3
$eBUSCA_1$	0.2	0.35	0.1

	t_1	t_2	t_3
DOC_1	0.3	0.0	0.5
DOC_2	0.5	0.4	0.3



Fonte: FERNEDA, 2003.

Para realizar o cálculo dos pesos dos termos da consulta (W_{iq}) e dos pesos dos termos dos documentos (W_{id}), é necessário que primeiro seja realizado o cálculo da frequência (*term frequency*) e o cálculo da frequência inversa (*inverse document frequency*).

A frequência é a quantidades de vezes que um termo *t* é encontrado em um documento *d* e esta pode ser obtida por meio da fórmula abaixo:

$$tf_{t,d} = frec_{t,d}$$

A frequência inversa calcula a relevância do termo *t* como fator de redução do *corpus*. Esta é encontrada a partir da seguinte fórmula:

$$idf_i = \text{Log}\ \frac{N}{n_i}$$

onde, *N* se refere ao número de documentos presente no *corpus* e n_i se refere a quantidade de documentos que contém o termo.

A fórmula para realizar o cálculo dos pesos utiliza os resultados obtidos a partir das fórmulas anteriores, ou seja, é o produto da frequência do termo em um documento pela frequência inversa do termo na coleção de documentos (CARDOSO, 2000) como mostra a fórmula abaixo:

$$W_{id} = frec(t_i, d) * idf_i$$

Após obter os pesos, estes podem ser aplicados para encontrar o grau de similaridade por meio do cosseno θ, do ângulo formado pelo vetor de termos no documento, representado pela variável *x* e pelo vetor de termos da busca, representado pela variável *y* (FERNEDA, 2003; JONES; FURNAS, 1987). Como mostra a fórmula abaixo:

$$\text{Sim}(x,y) = \frac{\sum_{i=1}^{t}(W_{i,x} \times W_{i,y})}{\sqrt{\sum_{i=1}^{t}(W_{i,x})^2} \times \sqrt{\sum_{i=1}^{t}(W_{i,y})^2}}$$

O modelo vetorial diferente do modelo booleano apresenta ordenação determinada através do grau de similaridade, ou seja, realiza uma ordenação por relevância entre os documentos e os termos de busca. Além disso, este modelo possui um bom desempenho e apresenta uma implementação simples (SOUZA, 2006).

3 SISTEMAS DE RECOMENDAÇÃO

A Web proporcionou o crescimento exponencial da informação, o que trouxe como consequência uma grande diversidade de conteúdo a disposição do usuário, tornando-se necessário a presença de recursos que auxiliem o usuário inexperiente a realizar suas escolhas.

O usuário, quando recebe uma recomendação sobre um item que está buscando, sente-se seguro para realizar sua escolha com eficácia (CAZELLA; NUNES; REATEGUI, 2010). Os Sistemas de Recomendação (SR) surgiram para sanar essa necessidade por meio de *softwares* e técnicas que proporcionam sugestões de itens para serem utilizados pelo usuário (RICCI et al.,2011).

O primeiro Sistema de Recomendação surgiu nos anos 90 e foi chamado de *Tapestry*. Este é um sistema que visava auxiliar o processo de recomendações de mensagens eletrônicas e grupos de notícias desenvolvido no *Xerox Palo Alto Research Center*. *Tapestry* utilizava o conceito de filtragem colaborativa e filtragem baseada em conteúdo (GOLDBERG et al., 1992).

Os Sistemas de Recomendação têm por objetivo reduzir a sobrecarga de conteúdo, por meio da seleção de informações baseadas nas prioridades do usuário (FIGUEIRA FILHO; GEUS; ALBUQUERQUE, 2008).

Atualmente, o mercado de comércio eletrônico faz grande uso dos Sistemas de Recomendação oferecendo recursos que ajudam seus clientes na compra de produtos que melhor satisfazem suas necessidades (SCHAFER; KONSTAN; RIEDL, 1999). Segundo Santini (2010), podem ser citados alguns exemplos de casos bem sucedidos que utilizam sistemas de recomendação como: Youtube®, Last.Fm®, Amazon®, Pandora®, NetFlix®, Google Books®, Google News®, Yahoo Music®, MovieLens®, e ChoiceStream®.

Sistemas de Recomendação são muito utilizados em sites de comércio eletrônico para sugerir itens aos seus clientes. Estes itens são recomendados com base em algumas abordagens: principais itens vendidos no site, região demográfica do cliente, uma análise do comportamento de compra que o cliente apresentou no passado como previsão para o futuro comportamento de compra. Estas técnicas são parte da personalização do perfil do cliente (SCHAFER; KONSTAN; RIEDL, 1999).

3.1 ABORDAGENS UTILIZADAS

Atualmente os SR têm sido classificados em três categorias/abordagens principais: a primeira abordagem é a baseada em conteúdo, ou seja, se utiliza das características dos itens a serem recomendados. Este tipo de abordagem é geralmente indicado para prover documentos textuais, visto que estes se baseiam em termos comuns (CAZELLA; NUNES; REATEGUI, 2010). A segunda é a abordagem de filtragem colaborativa em que a indicação de itens se baseia na troca de experiência entre os usuários, por exemplo, através de avaliações passadas realizadas sobre os itens (CAZELLA; NUNES; REATEGUI, 2010). A terceira abordagem é a híbrida unindo as duas abordagens anteriores (FIGUEIRA FILHO; GEUS; ALBUQUERQUE, 2008).

3.1.1 Recomendação Baseada Em Conteúdo

A filtragem realizada através de análises dos conteúdos dos itens é denominada filtragem baseada em conteúdo (*Content-based Filtering*) (LOPES, 2007).

A filtragem baseada em conteúdo (FBC) baseia-se na disponibilidade das descrições de itens que podem ser criados manualmente ou extraídos automaticamente, e um perfil que atribui importância a essas características. Os perfis de usuários também podem ser derivados automaticamente, ou seja, aprendidos por meio de interação do usuário com o sistema, ou podem ser obtidos por meio de informações que foram fornecidas previamente pelo próprio usuário (JANNACH et al, 2010).

Para descrever o perfil de itens é necessário manter uma lista com as características de cada item. Por exemplo, para recomendar um livro utiliza-se como características: o nome do autor, editor, título, gênero, tipo, preço, palavras-chave ou outras características que descrevam o item e que possam ser armazenadas em um sistema de banco de dados (JANNACH et al., 2010).

Este tipo de filtragem busca recomendar itens que possuam descrições similares a dos itens que anteriormente foram avaliados positivamente pelo usuário. Utiliza-se o conceito de que se possuem descrições similares serão avaliados de maneira semelhante (BOBADILLA et al., 2013).

Para recomendar um livro qualquer, o sistema poderia simplesmente verificar se o gênero do livro está na lista de gêneros

preferidos do usuário. A similaridade neste caso é 0 ou 1, ou seja, é semelhante ou não. Outra opção é calcular a semelhança ou sobreposição das palavras-chave (*Keywords*) envolvidas através do coeficiente de Dice que é um cálculo estatístico utilizado para comparar a similaridade entre características de vários valores. Se todos os livros *B* são descritos por um conjunto de palavras-chave (*bi*), o coeficiente de Dice mede a semelhança entre livros *bi* e *bj* (JANNACH et al., 2010), utilizando a fórmula abaixo:

$$\frac{2 \times |\text{keywords}\,(b_i) \cap \text{keywords}\,(b_j)|}{|\text{keywords}\,(b_i)| + |\,\text{keywords}\,(b_j)|}$$

onde é realizado o produto dos termos semelhantes (palavras-chaves) dos livros b_i e b_j por 2 e divide pelo somatório dos termos dos livros b_i e b_j.

Podem-se citar outros métodos para determinar a similaridade entre itens, por exemplo, o método do cosseno já descrito na seção sobre o modelo vetorial (seção 2.2.2). Na essência a filtragem baseada em conteúdo pode ser vista como uma recuperação de itens suportada pelo modelo vetorial.

3.1.2 Recomendação por meio de Filtragem Colaborativa

Durante o desenvolvimento do *Tapestry* surgiu o conceito de Filtragem Colaborativa para conceituar um sistema de filtragem de informação que utilizava a colaboração de dados de interesse entre usuários, por meio da interação humana no sistema (GOLDBERG et al., 1992; CAZELLA; NUNES; REATEGUI, 2010).

Segundo Cazella, Nunes e Reategui (2010), a abordagem Colaborativa ou Filtragem Colaborativa (FC) é a indicação de itens a usuários baseada na troca de experiências de outros usuários que possuem interesses similares. A indicação ocorre através de avaliações que estes realizaram para os itens.

Devido a grande quantidade de informações geradas através da interação de usuários em um determinado sistema (ex: site de compras da *Amazon*®), as ações como compras realizadas, itens visualizados e buscas feitas são armazenadas e processadas através de métodos computacionais, capazes de obter resultados para realizar sugestões de itens similares por meio das experiências de outros usuários (JANNACH et al., 2010).

A Tabela 2 ilustra um exemplo de como é o funcionamento de um Sistema de Recomendação na prática. Para recomendar um produto a um usuário é analisado o perfil de outros usuários que possuem interesses semelhantes. Por exemplo, para sugerir itens para o *User 6* serão analisadas as interações de usuários com hábitos semelhantes. Se o *User1*, *User 2* e o *User 6* adquiriram o mesmo produto, ou seja, o *Prod. 2*, então será recomendado outros produtos que os dois usuários já adquiriram e que o *User 6* ainda não possui. Na tabela abaixo o produto de maior relevância para a sugestão será o *Prod.1* e o de menor relevância será o *Prod.5* (REATEGUI; CAZELLA, 2005).

Tabela 2: Matriz Usuários X Produtos.

Usuários	*Prod.1*	*Prod.2*	*Prod.3*	*Prod.4*	*Prod.5*	*Prod.6*
User1	X	X			X	
User2	X	X				
User3			X	X	X	
User4			X			X
User5	X			X		
User6		X				

Fonte: Autores.

Os métodos mais utilizados e de maior relevância para calcular a similaridade entre itens são: *K*-Vizinhos mais Próximos e modelos de correlação.

3.1.2.1 K-Vizinhos mais Próximos

O algoritmo K-Vizinhos mais Próximos (*K-Nearest Neighbor*) pode ser dividido em duas abordagens: *K*-Vizinhos mais Próximos baseado no Usuário (*User-based Nearest Neighbor Recommendation*) e *K*-Vizinhos mais Próximos baseados no Item (*Item-based Nearest Neighbor Recommendation*) (SCHAFER et al., 2007).

O *K*-Vizinhos mais Próximos baseado no Usuário utiliza a função de similaridade para calcular a similaridade entre usuários com interesses comuns (MORAIS, 2012). A similaridade pode ser encontrada através de vários coeficientes, sendo a fórmula de coeficiente de Correlação de Pearson a mais utilizada (HERLOCK, 2000).

$$W(a,u) = \frac{\sum_{u=1}^{m}[(r_{a,i} - \bar{r}_a)\times(r_{u,i} - \bar{r}_u]}{\sqrt{\sum_{i=1}^{m}(r_{a,i} - \bar{r}_a)^2 \times \sum_{i=1}^{m}(r_{u,i} - \bar{r}_u)^2}}$$

De acordo com a fórmula de coeficiente de Correlação de Pearson ilustrada acima, *W(a,u)* é a correlação do usuário ativo *a* com um usuário *u*, $r_{a,i}$ é a avaliação que o item *i* recebeu do usuário, $\bar{r}_a$ é a média das avaliações do usuário *a, e* $r_{u,i}$ é o conjunto de avaliações do vizinho. Os resultados podem variar de 1 a -1, sendo 1 similaridade total e -1 para ausência de similaridade (CAZELLA, 2006).

A previsão dos itens para o usuário ocorre através da fórmula de predição, ilustrada abaixo:

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n}(r_{u,i} - \bar{r}_u) \times W_{a,u}}{\sum_{u=1}^{n} |W_{a,u}|}$$

onde $p_{a,i}$ é a média ponderada das avaliações realizadas ao item *i* pelos *n* vizinhos do usuário *a* (HERLOCK, 2000).

A abordagem *K*-Vizinhos mais Próximos baseado no Item é mais vantajosa do que a abordagem *K*-Vizinhos mais Próximos baseado no usuário quando o número de usuários é muito elevado. Nesta situação a recomendação se torna lenta, pois a cada recomendação é realizada a comparação de todos os usuários e de todos os itens que cada usuário avaliou. Esta avaliação (baseada no usuário) é realizada em tempo real diferentemente da abordagem baseada no item que realiza esse processamento *offline* o que a torna mais vantajosa (MORAIS, 2012).

Na abordagem baseada em itens a similaridade é calculada entre os itens, não no contexto de conteúdo, mas em termos de avaliações (pesos) realizadas pelos usuários sobre os itens. Para encontrar a similaridade de itens é utilizada a fórmula abaixo (SCHAFER et al., 2007):

$$Wi,j = \frac{\sum u \in U(r_{u,i} - \bar{r}_i) \times (r_{u,j} - \bar{r}_j)}{\sqrt{\sum u \in U(r_{u,i} - \bar{r}_i)^2} \times \sqrt{\sum u \in U(r_{u,j} - \bar{r}_j)^2}}$$

onde *U* representa o conjunto de usuários que avaliaram os itens, $r_{u,i}$ é a avaliação que o item *i* recebeu do usuário *u* e $\bar{r}_i$ é a média das avaliações que o item *i* recebeu.

A previsão dos itens para o usuário ocorre através da fórmula de predição, ilustrada abaixo:

$$p_{a,i} = \frac{\sum j \in K \; r_{a,j} \, w_{i,j}}{\sum j \in K \; |w_{i,j}|}$$

onde *K* representa os vizinhos do usuário *a* sob o item *i* (MORAIS, 2012).

3.1.2.2 Modelos Baseados em Coocorrência

Os modelos baseados em coocorrência buscam encontrar uma relação entre um conjunto de termos por meio de uma análise das coocorrências destes (GONÇALVES, 2006). Como exemplo, podem-se utilizar dois termos *Y* e *Z*, onde a coocorrência indica o quão relacionado estes termos estão.

A Tabela 3 demostra um conjunto de coeficientes de correlação, que pode ser compreendida como uma matriz de correlação ITEM X ITEM, em que cada célula representa a coocorrência entre dois itens, representado pelo símbolo $W(P_i, P_j)$.

Para melhor entendimento, o processo de recomendação foi exemplificado através de matrizes (tabelas), que aplicando o método de coocorrência na Tabela 2 (disponível na seção 3.1.2) obtêm-se a Tabela 3. Na Tabela 2 é possível analisar que o *User 1* adquiriu o *prod.1* e o *prod.2*, dessa forma ocorreu a correlação $(w_{(p1,p2)})$ destes itens, ou seja, implica que o *prod.1* está relacionado com o *prod.2*.

Tabela 3: Matriz de correlação ITEM X ITEM.

ITENS	p1	p2	p3	p4	p5	p6
p1		$w_{(p1,p2)}$		$w_{(p1,p4)}$	$w_{(p1,p5)}$	
p2	$w_{(p2,p1)}$				$w_{(p2,p5)}$	
p3				$w_{(p3,p4)}$	$w_{(p3,p5)}$	$w_{(p3,p6)}$
p4	$w_{(p4,p1)}$		$w_{(p4,p3)}$		$w_{(p4,p5)}$	$w_{(p4,p6)}$
p5	$w_{(p5,p1)}$	$w_{(p5,p2)}$	$w_{(p5,p3)}$	$w_{(p5,p4)}$		
p6			$w_{(p6,p3)}$	$w_{(p6,p4)}$		

Fonte: Autores.

Para obter o grau de relação entre os produtos são utilizados cálculos estatísticos (STEVENSON, 2001). Existem vários métodos para realizar estes cálculos, entre eles o *Phi-squared* que se baseia na frequência individual e conjunta para estabelecer determinado grau de correlação.

A frequência conjunta é um método simples para encontrar a relação entre dois elementos (GONÇALVES, 2006). Esta considera a quantidade de vezes que o elemento ocorre (SCHIESSL, 2007). Quando aplicado este método em documentos textuais é necessário que ocorra um tratamento para evitar que seja calculada a frequência conjunta de

elementos irrelevantes como artigos e preposições, como técnica para tratar este problema utiliza-se *Stop List* (seção 2.1.2).

O método *Phi-squared* utiliza uma tabela de contingência, ou seja, uma matriz que apresenta a distribuição de frequência das variáveis. Segundo Church e Gale (1991) este método favorece as associações que apresentam frequência elevada. O *Phi-squared* é representado da seguinte forma:

$$ø^2 = \frac{(ad-bc)^2}{(a+b)(c+d)(a+c)(b+d)}, \text{ onde } 0 \leq ø^2 \leq 1.$$

3.1.3 Analise entre as Abordagens Recomendação Baseada Em Conteúdo e Recomendação por meio de Filtragem Colaborativa

Segundo Garcia e Frozza (2013) a filtragem baseada em conteúdo possui maior aplicabilidade em recomendações textuais, devido à facilidade que existe em verificar o quão similar é o interesse do usuário com o texto, pois pode ser utilizadas palavras-chave para verificar esta similaridade.

Uma vantagem que este tipo de filtragem apresenta é que não necessita que existam muitos usuários, ou registro de interações no sistema, visto que a recomendação é baseada na semelhança de conteúdo entre os itens e o interesse do usuário, tornando possível a recomendação mesmo que exista apenas um usuário ativo no sistema (JANNACH et al., 2010).

A filtragem colaborativa possui aplicabilidade na recomendação de produtos, pois não exige descrição dos atributos dos produtos, ou seja, não existe a necessidade de obter conhecimento a respeito do produto, apenas utiliza a troca de experiência entre os usuários com interesses comuns (JANNACH et al., 2010).

3.1.4 Híbrida

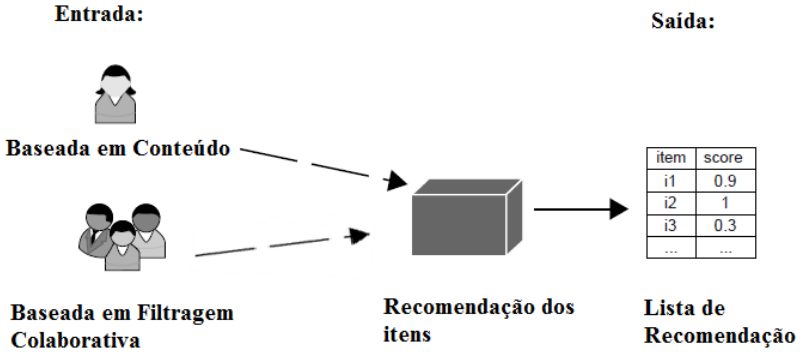
Os Sistemas de Recomendação Híbridos realizam a combinação dos pontos fortes de diferentes modelos de recomendação com o objetivo de criar estratégias mais eficientes (JANNACH et al., 2010).

Existem diversas maneiras de combinar os tipos de filtragem para potencializar suas vantagens e minimizar as desvantagens, uma dessas maneiras é por meio da aplicação de dois tipos de filtragem separadamente, fazendo a combinação das recomendações. Outra forma bastante utilizada é agregar características da Filtragem Baseada em

Conteúdo na Filtragem Colaborativa ou incorporar características da Filtragem Colaborativa na Filtragem Baseada em Conteúdo (MORAES, 2012).

A Figura 7 ilustra um Sistema de Recomendação Híbrido como uma caixa preta, onde transforma os dados de entrada em uma lista ordenada de itens como saída. Neste exemplo é utilizado os Modelos Baseado em Conteúdo e Filtragem colaborativa.

Figura 7: Sistema de Recomendação Híbrido.



Fonte: Adaptado de (JANNACH et al., 2010).

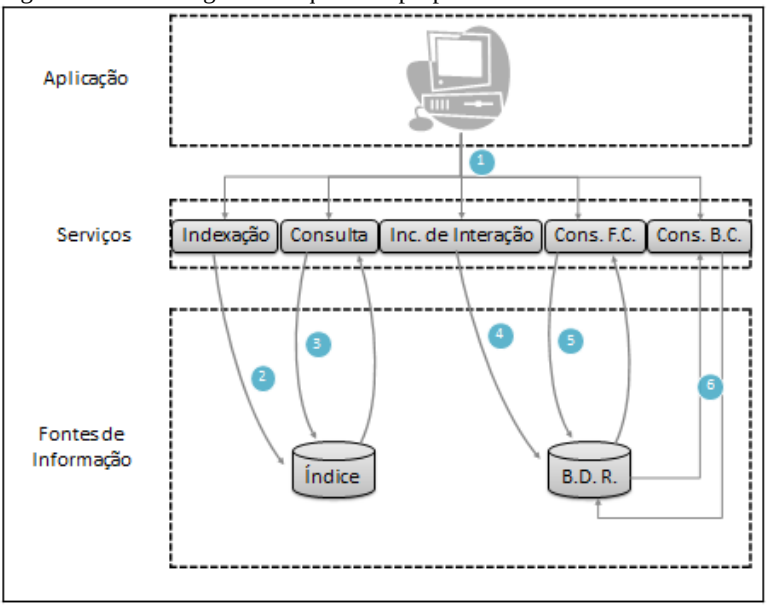
4 ARQUITETURA PROPOSTA

Este capítulo apresentará a arquitetura proposta dividindo-a em duas etapas. Sendo a primeira etapa referente ao modelo lógico detalhando a interação entre as camadas que compõem a arquitetura. A segunda etapa aborda o modelo físico, expondo os componentes tecnológicos, assim como, a justificativa da utilização destes.

4.1 MODELO LÓGICO

O modelo lógico ilustrado na Figura 8, apresenta conceitualmente as camadas da arquitetura proposta permitindo a Recuperação e a Recomendação de conteúdo textual.

Figura 8: Modelo lógico da arquitetura proposta.



Fonte: Autores.

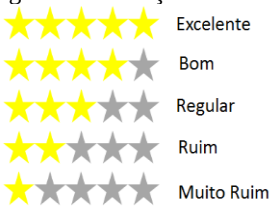
4.1.1 Camada de Aplicação

A Camada de Aplicação permite a interação entre o usuário e o sistema, possibilitando que o usuário realize uma busca e/ou avalie um documento. A interação ocorre através de uma página web e esta, além

de recuperar o conteúdo resultante da pesquisa realizada pelo usuário, proporciona a recomendação de novos conteúdos, podendo estes serem semelhantes a um documento de interesse previamente recuperado ou documentos que são recomendados por meio das experiências de outros usuários que possuem perfis semelhantes.

A avaliação dos documentos se baseia nos critérios Excelente, Bom, Regular, Ruim e Muito Ruim, como ilustra a Figura 9.

Figura 9: Avaliação do conteúdo.



Fonte: Autores.

No entanto esta camada não está definida no escopo da arquitetura. Para simular esta funcionalidade foi desenvolvido um programa (*software*) que realiza as interações de busca e avaliação dos documentos de maneira automática.

4.1.2 Camada de Serviço

A Camada de Serviço realiza a interconexão entre a camada de aplicação e a camada de fontes de informação (na Figura 8 está representado pela seta número 1). Os serviços executados por esta camada são: Indexação, Consulta, Inclusão de Interação, Consulta Baseada em Filtragem Colaborativa e Consulta Baseada em Conteúdo.

4.1.2.1 Indexação

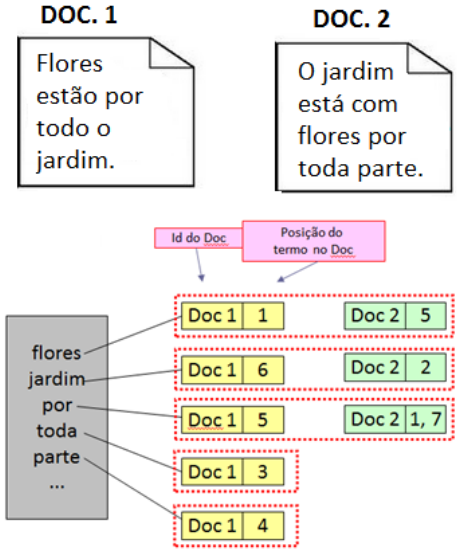
As estruturas de dados tradicionalmente utilizadas para realizar buscas lexicográficas são índices invertidos. Estes permitem encontrar documentos que são correspondentes aos termos de busca de maneira eficiente (HARMAN et al.,1992).

O processo de indexação consiste em converter dados textuais em listas invertidas, mas antes da criação do índice o texto precisa ser pré-processado, ou seja, tratado em etapas de: extração de termos, lista de termos, extração de radicais e tesauro como mencionado na seção 2.1 (HARMAN et al.,1992).

Após a extração dos termos relevantes cria-se os índices conforme ilustra a Figura 10, onde todas as palavras dos textos são inseridas em uma lista (índice), cada uma destas palavras referenciam os documentos onde estão presentes, além de armazenarem a posição onde se encontram no documento.

O índice invertido é dividido em duas partes: o vocabulário e as ocorrências. O vocabulário forma uma lista com todas as palavras distintas do texto e para cada palavra no vocabulário tem-se uma lista de ocorrências relacionando todos os documentos onde o termo ocorre com suas respectivas posições dentro do texto (BAEZA-YATES; RIBEIRO-NETO, 1999).

Figura 10: Índice Invertido.



Fonte: Autores.

4.1.2.2 Consulta

A Camada de Serviço receberá uma requisição oriunda da interação realizada pelo usuário na camada de aplicação. O usuário informa o termo que deseja buscar, o serviço de consulta localiza os termos no índice invertido, encontrando os documentos onde essas palavras ocorrem e retorna para a camada de aplicação os documentos resultantes da busca.

4.1.2.3 Inclusão de Interação

As interações realizadas pelo usuário podem ser de dois tipos: Consulta e/ou Avaliação dos documentos. Cada interação será armazenada vinculando-se a uma sessão e esta poderá armazenar várias interações. Os dados de cada sessão são persistidos em um banco de dados e são utilizados para realizar recomendações em futuras buscas.

4.1.2.4 Consulta Baseada em Filtragem Colaborativa

O serviço de consulta baseada em filtragem colaborativa recomenda conteúdos baseado na experiência de outros usuários que possuem perfis semelhantes, por meio de avaliações realizadas nos documentos visualizados (representado na Figura 8 pela seta número 5).

4.1.2.5 Filtragem Baseada em Conteúdo

O serviço de filtragem baseada em conteúdo recomenda documentos que possuam descrições similares a determinado item (documento) de interesse (representado na Figura 8 pela seta número 6). Este tipo de filtragem é indicado para recomendações que envolvam conteúdos textuais.

4.1.3 Camada de Fontes de Informação

A Camada de Fontes de Informação é responsável pelo processo de armazenamento e pelo processo de Recuperação da Informação. O armazenamento pode acontecer nas seguintes situações: quando um documento é indexado (na Figura 8 pode ser visualizado na seta seguida do número 2) ou quando são armazenadas as interações de um usuário (na Figura 8 é visualizado na seta número 4). As interações são armazenadas em um banco de dados relacional.

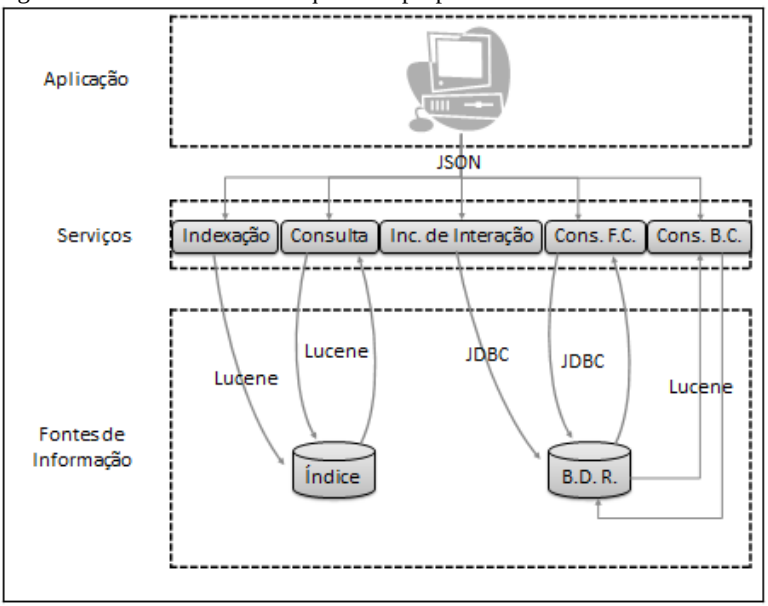
O processo de Recuperação de Informação é desencadeado após a consulta realizada pelo usuário (na Figura 8 estando representado pela seta de número 3).

4.2 MODELO FÍSICO

O modelo físico apresentado na Figura 11 demostra as funcionalidades e tecnologias utilizadas para o desenvolvimento da

arquitetura proposta nas seguintes camadas: Aplicação, Serviços e Fontes de Informação. Este modelo possui o objetivo de especificar as fases do processo de Recuperação e Recomendação de conteúdo textual.

Figura 11: Modelo físico da arquitetura proposta.



Fonte: Autores.

4.2.1 Camada de Aplicação

A camada de Aplicação permite ao usuário interagir com o sistema realizando buscas e avaliações no *corpus* (coleção de documentos).

No presente trabalho foi simulada a camada de aplicação através de um programa implementado pela tecnologia Java® e desenvolvido no ambiente de desenvolvimento Eclipse®. O programa simula de maneira randômica as interações de buscas e avalições realizadas pelo usuário.

4.2.2 Camada de Serviço

Esta camada é responsável por fazer a interconexão entre a Camada de Aplicação e a Camada de Fontes de Informação. Os serviços definidos nesta camada são: Indexação, Consulta, Inclusão de Interação,

Consulta Baseada em Filtragem Colaborativa e Consulta Baseada em Conteúdo. Para a implementação da camada de serviço foi utilizando o conceito de *Servlet* da linguagem Java®.

Após o recebimento de uma requisição JSON (*JavaScript Object Notation*) proveniente da camada de aplicação, a camada de serviço executará o serviço correspondente à requisição.

4.2.2.1 Indexação

Para a realização da indexação utilizou-se a API Lucene™, esta é um projeto da fundação Apache. Possui como objetivo a indexação e a busca de conteúdo textual por meio de um *framework open source* desenvolvido na linguagem Java. A API Lucene™ realiza a indexação através de índice invertido (HATCHER; GOSPODNETIC; MCCANDLESS, 2010).

Para realizar a indexação são necessários alguns procedimentos. Primeiramente o documento textual deve ser extraído dos formatos HTML, XML, Microsoft® Word, arquivos em PDF, entre outros, para disponibilizar os dados no formato de texto simples. A extração pode ser realizada por meio de conversores de dados, chamado de *parsers*.

Após a extração é necessário realizar o pré-processamento do texto, ou seja, a extração de termos (*Tokenization*), podendo envolver também a eliminação de termos a partir de uma lista (*Stop List*), a extração de radicais (*Stemming*) e o uso de tesauro (*Thesauru*). O Lucene realiza este procedimento através de analisadores como: *SimpleAnalyzer*, *StandardAnalyzer*, *StopAnalyzer*, *SnowballAnalyzer*, *BrazilianAnalyzer* entre outros. Neste trabalho foi utilizado o analisador *BrazilianAnalyzer*.

O Lucene realiza a indexação por meio da classe *indexWriter*, através de uma requisição JSON contendo as informações para a indexação. Na Figura 12 é apresentado um exemplo do objeto JSON enviando os dados para indexação.

Na requisição é necessário informar a operação que será realizada, que neste caso é a indexação (*operation: index*) sendo a estrutura deste objeto formada por quatro campos (*text, title, type, key*), onde o campo *text* representa o corpo do documento, o campo *title* representa o título do documento, o campo *type* representa o tipo do documento que esta sendo indexado, por exemplo, pdf, xml ou doc, e o campo *key* representa o identificador único do documento. Cada um destes campos possui quatro atributos (*analyzed, content, field, stored*); o atributo *analyzed* indica se será aplicado o analisador para a

tokenização das palavras; o atributo *content* indica o conteúdo que poderá ser analisado (de acordo com a configuração realizada no *analyzed*) de acordo com a configuração realizada no *stored*; o atributo *stored* indica se o conteúdo será armazenado no índice ou não e o atributo *field* indica o nome que o campo terá no índice.

Figura 12: Objeto JSON para Indexação.

Fonte: Autores.

4.2.2.2 Consulta

Para realizar a busca utilizou-se a API Lucene que efetua a busca no índice criado pelo serviço anterior. O Lucene utiliza a classe *IndexSearcher* para realizar as buscas, que podem ocorrer de duas maneiras: busca por termo ou busca por sentença respectivamente implementadas por *TermQuery* e *PhraseQuery*.

Através desta camada realiza-se uma requisição JSON contendo um termo de busca, possibilitando a troca de informações com a camada de serviço. Na Figura 13 pode ser visualizado um exemplo desta requisição.

A requisição JSON é composta pelos campos: *project*, *operation*, *hits*, *fragment_field*, *page* e *query*. O campo *project* indica os campos que deverão retornar, o campo *operation* indica a operação que será realizada, neste caso busca (*search*), o campo *hits* indica quantos itens serão retornados, o campo *fragment_ field* indica as sentenças relevantes do documento, o campo *page* indica a quantidade de páginas que apresentarão os resultados e o campo *query* indica o termo que está sendo buscado.

Figura 13: Objeto JSON para Consulta.

Fonte: Autores.

Como resposta, o servidor envia para a Camada de Aplicação um objeto JSON conforme ilustra a Figura 14. O objeto é composto por três elementos cada um representando um documento com os devidos atributos: o identificador do documento (*id*), o título do documento (*title*) e o peso indicando o grau de relevância do documento com o termo ou sentença buscada (*score*).

Figura 14: Resposta para a requisição JSON de Consulta.

Fonte: Autores.

4.2.2.3 Inclusão de Interação

Este serviço insere as interações realizadas pelo usuário em uma tabela que irá registrar a data em que ocorreu a interação, a sessão, o tipo de interação que está sendo realizado (0: acesso/leitura do documento e 1 – avaliações como apresentado na Figura 9). No caso da interação realizada do tipo avaliação será armazenado o identificador do documento que recebeu a avaliação e a avaliação, podendo variar de 1 à 5.

4.2.2.4 Consulta Baseada em Conteúdo

Para a realização deste serviço serão utilizados dados provenientes do trabalho desenvolvido por Geronimo e Anacleto (2014). Os dados se encontram em uma tabela que armazena as recomendações geradas. A tabela juntamente com o modelo lógico utilizado para compor o banco de dados relacional desta arquitetura será apresentado na Figura 17.

Os dados que compõe a tabela de recomendação são resultantes da aplicação do método do cosseno (já descrito na seção 2.2.2), onde foi gerado o grau de similaridade de cada documento em relação a todos os documentos da base. Na

Tabela 4 apresenta uma matriz simulando o grau de similaridade entre todos os documentos.

Tabela 4: Matriz Grau de Similaridade.

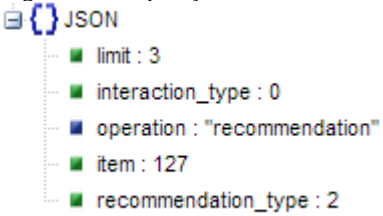
Documentos	Doc. 1	Doc. 2	Doc. 3	Doc. n
Doc. 1	0	0.3	0.4	0.6
Doc. 2	0.3	0	0.6	0.2
Doc. 3	0.4	0.6	0	0.1
Doc. n	0.6	0.2	0.1	0

Fonte: Autores.

Para a realização da recomendação o sistema apresenta vários documentos que correspondem a busca realizada pelo o usuário, sendo que para cada documento o sistema recomendará outros documentos que possuam similaridade.

Na Figura 15 é apresentada uma requisição JSON, enviando o identificador do documento (*item: 127*) oriundo da consulta realizada anteriormente (Figura 14), a quantidade máxima de documentos que poderão ser retornados (*limit: 3*), o tipo da interação (*interaction_type: 0*) conforme seção 4.2.2.3, o tipo da operação, neste caso é recomendação (*operation: "recommendation"*), e o tipo da recomendação (*recommendation_type: 2*), em que 1 se refere a recomendação por filtragem colaborativa e 2 para recomendação baseada em conteúdo.

Figura 15: Requisição JSON de FBC.

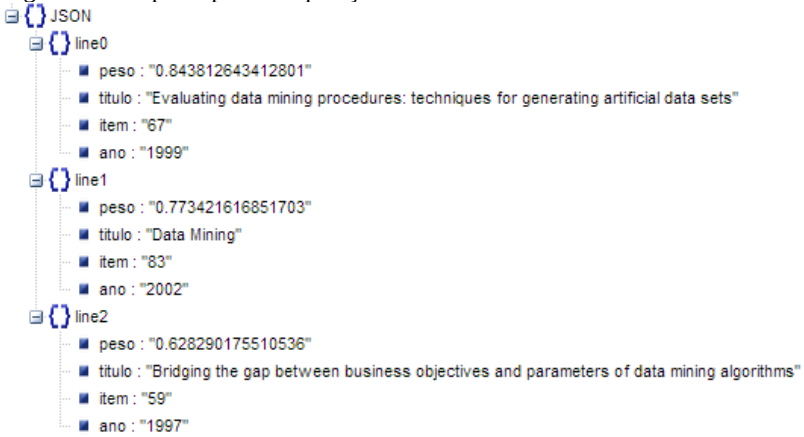


Fonte: Autores.

Como resposta, o servidor envia para a Camada de Aplicação um objeto JSON conforme ilustra a Figura 16. O objeto é composto por cinco elementos cada um representando um documento com os devidos atributos: *peso*, indicando o grau de similaridade do documento com o

termo ou sentença buscada, título do documento (*titulo*), o identificador do documento (*item*) e o ano do documento (*ano*).

Figura 16: Resposta para a requisição JSON de FBC.



Fonte: Autores.

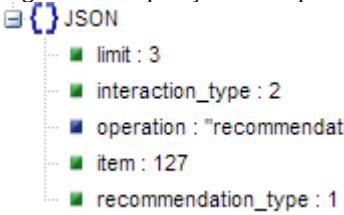
4.2.2.5 Consulta Baseada em Filtragem Colaborativa

Para a realização deste serviço também são utilizado os dados da tabela de recomendação. Para recomendar um documento utiliza-se o peso resultante do cálculo da similaridade, utilizando o método de correlação *Phi-squared* (já descrito na seção 3.1.2.2).

Esse tipo de recomendação utiliza o padrão de comportamento do usuário para realizar a recomendação. Assim quando o usuário 1 avaliar o Doc. 1, Doc. 2 e Doc. 3 entende-se que os documentos estão correlacionados e caso o usuário 2 avalie o Doc. 1 e Doc. 2 poderá ser recomendado o Doc. 3, já que estes apresentaram interesses comum.

Para melhor entendimento do funcionamento da Recomendação Baseada em Filtragem Colaborativa a Figura 17 apresenta uma requisição JSON em que consta o identificador do documento (*item*: *127*), a quantidade máxima de documentos que poderá ser exibida (*limit:3*), tipo de interação (*interaction_type*: *2*, ou seja, interação do tipo Avaliação), a operação realizada (*operation*:"*recommendation*"), o tipo da recomendação é (*recommendation_type*:1), indicando filtragem colaborativa.

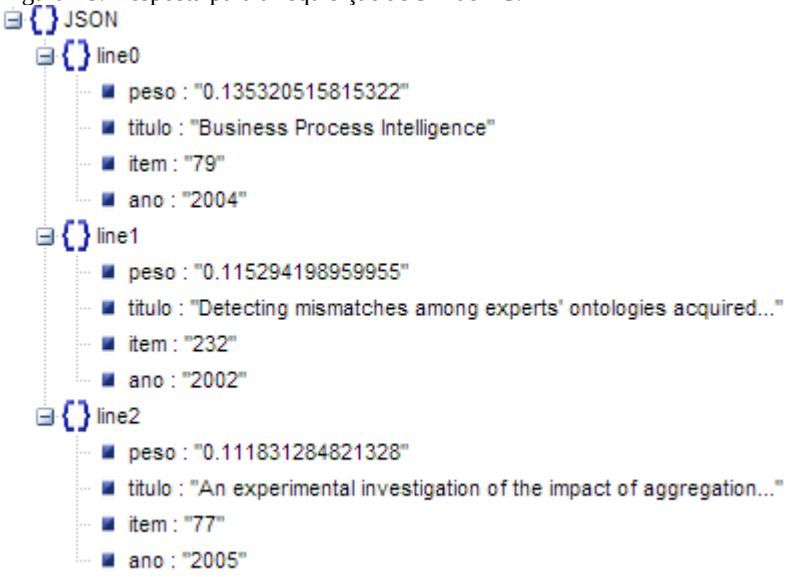
Figura 17: Requisição JSON para consulta baseada em FC.



Fonte: Autores.

O retorno esperado pela Camada de Aplicação é um objeto JSON contendo as informações sobre os documentos resultantes da busca realizada. A Figura 18 apresenta o objeto JSON composto por cinco elementos, ou seja, cinco documentos com os devidos atributos: *peso*, indicando o grau de similaridade do documento com o termo ou sentença buscada, o titulo do documento (*titulo*), o identificador do documento (*item*) e o ano do documento (*ano*).

Figura 18: Resposta para a requisição JSON de FC.



Fonte: Autores.

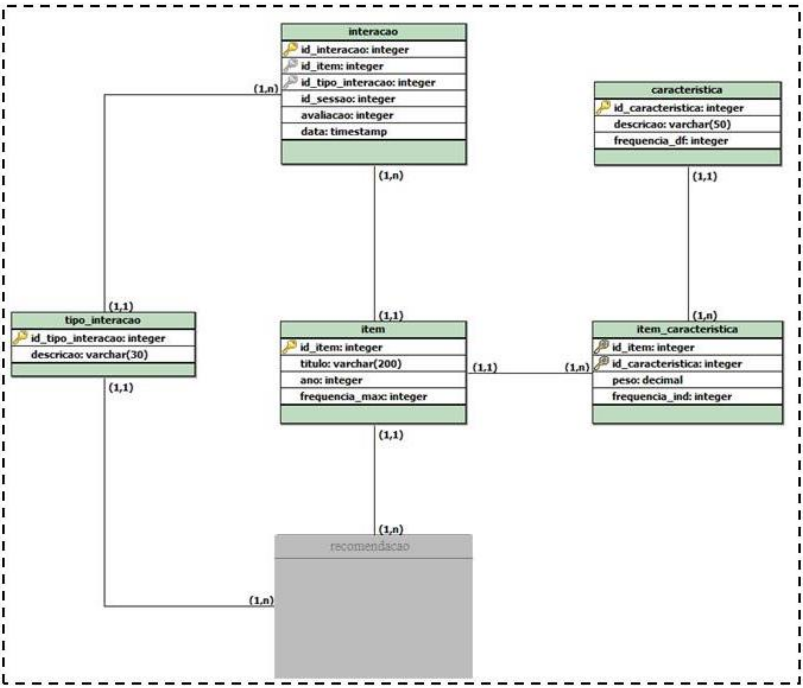
4.2.3 Camada de Fontes de Informação

A Camada de Fontes de Informação realiza o armazenamento e possibilita a recuperação dos dados através da Camada de Serviço. O armazenamento pode ocorrer de duas formas: uma referente à indexação dos documentos no índice (seção 4.2.2.1) e a outra a partir da interação do usuário com o sistema, persistindo os dados em um Banco de Dados Relacional.

O banco de dados relacional foi desenvolvido utilizando o Sistema Gerenciador de Banco de Dados (SGBD) PostgreeSQL® por se tratar de um SGBD *open source.*

O banco de dados foi modelado para comportar a arquitetura desenvolvida conforme a Figura 19. Para isto foram criadas as seguintes tabelas: "*tipo_interacao*", "*interacao*", "*item*", "*item_caracteristica*" e "*característica*". A tabela "*recomendacao*" é auxiliar a esta arquitetura e foi desenvolvida no trabalho de Geromino e Anacleto (2014).

Figura 19: Modelo lógico do banco de dados.



Fonte: Autores.

Os tipos de interações que serão realizadas pelo usuário são armazenados na tabela "*tipo_interacao*" (Figura 20). As interações possíveis nesta arquitetura são seleção/leitura e avaliação de documentos. O atributo "*id_tipo_interacao*" armazena o identificador do tipo da interação, sendo 0 para seleção/leitura de um documento em uma sessão, e 1 para a avaliação do documento também em determinada sessão. Já o atributo "*descrição*" armazena a descrição do tipo da interação.

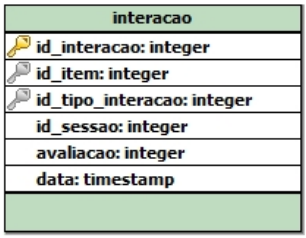
Figura 20: Tabela tipo_interaçao.

Fonte: Autores.

Todas as interações realizadas pelo usuário acontecem dentro de uma sessão, neste caso, cada consulta realizada pelo usuário é tratada

como uma sessão. Cada interação realizada pelo usuário promove um registro na tabela "*interacao*" (Figura 21). As informações coletadas das interações serão utilizadas para realizar o processamento das recomendações. O atributo "*id_interacao*" representa o identificador único de cada interação, o "*id_item*" identifica o item que o usuário interagiu, o "*id_tipo_interaçao*" indica o tipo da interação efetuada pelo usuário. Já o atributo "*id_sessao*" é a identificação da sessão do usuário e o atributo "*avaliacao*" é responsável por armazenar o valor da avaliação feita pelo usuário em determinado documento, podendo variar de 1 à 5. A data em que as interações ocorreram é armazenada no atributo "data".

Figura 21: Tabela interação.



Fonte: Autores.

As informações sobre os documentos estarão armazenadas na tabela "*item*", como ilustra a

Figura 22. O atributo "*id_item*" é o identificador único de cada documento, e o título e o ano dos documentos são armazenados respectivamente nos atributos "*titulo*" e "*ano*". Outros atributos como autores, editora são importantes, mas não neste momento, sendo desnecessário o seu armazenamento. Por fim, o atributo "*frequencia_max*" armazena a frequência do atributo que possui a máxima frequência no item (documento).

Figura 22: Tabela item.

Fonte: Autores.

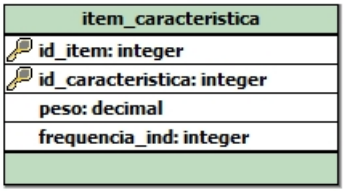
Cada documento possui um conjunto de palavras-chave que descrevem o documento sendo estas armazenadas na tabela "*caracteristica*" (Figura 23). O atributo "*id_caracteristica*", representa o identificador de cada palavra-chave, a descrição das palavras são armazenadas em "*descricao*", e a frequência com que a palavra ocorre no conjunto de documentos é armazenada em "*frequência_df*".

Figura 23: Tabela característica.

Fonte: Autores.

A tabela "*item_caracteristica*" (Figura 24) estabelece o relacionamento entre as tabelas "item" e "*caracteristica*". O atributo "*id_item*" representa o identificador do documento e o atributo "*id_caracteristica*" o identificador da palavra-chave relacionada. Cada palavra-chave possui um peso que indica a importância da palavra no documento ("*peso*") e a frequência em que a palavra ocorre no documento, pelo atributo "*frequência_ind*".

Figura 24: Tabela item_caracteristica.



Fonte: Autores.

Com as informações contidas nas tabelas "*item*", "*característica*", e "*item_caracteristica*" é possível determinar o peso de cada característica de um item (documento) através do cálculo do *tf-idf*, para posteriormente permitir o cálculo de similaridade através da equação do cosseno.

5 DESENVOLVIMENTO E APRESENTAÇÃO DE RESULTADOS

Este capítulo apresentará os resultados, objetivando uma visão da camada de serviços bem como a utilização de um sistema de recuperação e recomendação de informação textual. Para executar o serviço de recuperação e recomendação foi construída uma base de dados com artigos coletados na revista *ScienceDirect*[1].

5.1 INTRODUÇÃO

As discussões a respeito deste capítulo foram divididas em três partes, sendo estas:

- **Fluxo De Execução Dos Serviços:** apresenta de forma mais detalhada o protótipo desenvolvido para os serviços de Indexação, Consulta, Inclusão de interações, Recomendação baseada em Filtragem Colaborativa e Recomendação Baseada em Conteúdo.
- **Cenário de aplicação**: demonstra o cenário de aplicação do protótipo de maneira geral.
- **Apresentação dos resultados**: apresenta algumas consultas visando demonstrar os resultados obtidos através dos serviços de recomendação aplicados ao cenário.

5.2 FLUXO DE EXECUÇÃO DOS SERVIÇOS

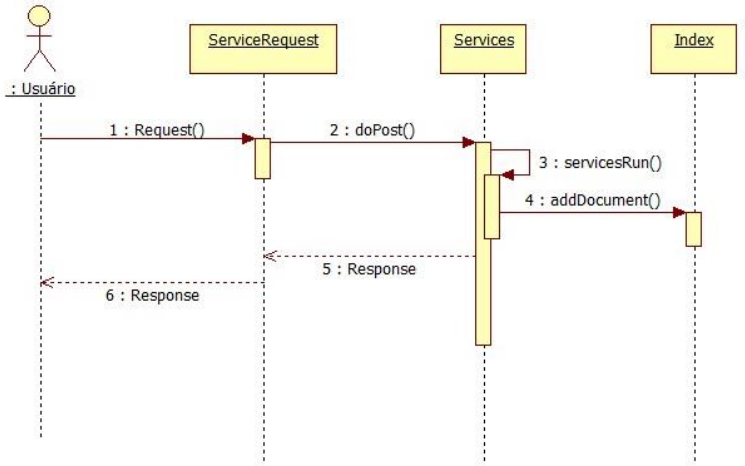
Nesta sessão será discutido o protótipo de forma detalhada para fornecer um melhor entendimento dos serviços que compõem a arquitetura proposta.

5.2.1 Indexação

Este serviço permite indexar qualquer tipo de documento textual. Para melhor apresentar o comportamento deste serviço pode ser visualizado o diagrama de sequência (Figura 25), que demostra os objetos do sistema e os devidos comportamentos na execução.

[1] http://www.sciencedirect.com/

Figura 25: Diagrama de sequência do serviço de indexação.



Fonte: Autores.

O usuário envia uma requisição ao *ServiceRequest* criando um canal de conexão para converter a requisição em uma sequência de *bytes* para ser enviado a classe *Services*, que receberá a mensagem por meio do método *doPost()*. Este método recebe a mensagem de requisição repassando o tipo de serviço a ser realizado e os parâmetros a serem indexados pelo método s*erviceRun()*, que neste caso, o tipo de serviço é indexação.

O método *serviceRun()* recebe a mensagem de requisição no formato de um *InputStream*, realiza a conversão para *String* e gera um objeto JSON. O *serviceRun()* verifica também o serviço a ser realizado e chama o método *addDocument*() para indexar o conteúdo enviado.

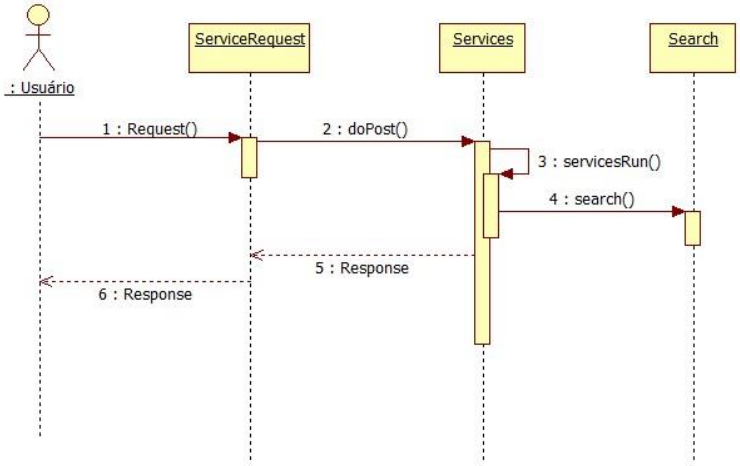
O método *addDocument()* recebe o objeto JSON e realiza a análise do conteúdo deste extraindo-o para uma lista. Esta lista é composta por campos que se referem aos atributos do documento que serão indexados. No índice o documento é uma instância da classe *Document* e para cada campo (*field*) há três propriedades: *content*, *stored* e *analyzed*.

Após isso o método *serviceRun()* retorna ao usuário se a indexação ocorreu com sucesso ou não. Em caso positivo o documento será adicionado a base de índices, caso contrário, o usuário receberá uma mensagem com a possível causa do erro.

5.2.2 Consulta

Este serviço permite realizar buscas e recuperar documentos textuais. Para melhor apresentar o comportamento deste serviço pode ser visualizado o diagrama de sequência (Figura 26), que demostra os objetos do sistema e os devidos comportamentos na execução.

Figura 26: Diagrama de sequência do serviço de consulta.



Fonte: Autores.

O usuário envia uma requisição ao *ServiceRequest* através de um objeto JSON estabelecendo um canal de comunicação com a classe *Services* por meio do protocolo HTTP. O método *doPost()* da classe *Services* recebe uma mensagem de requisição repassando o tipo de serviço a ser realizado pelo método s*erviceRun()*.

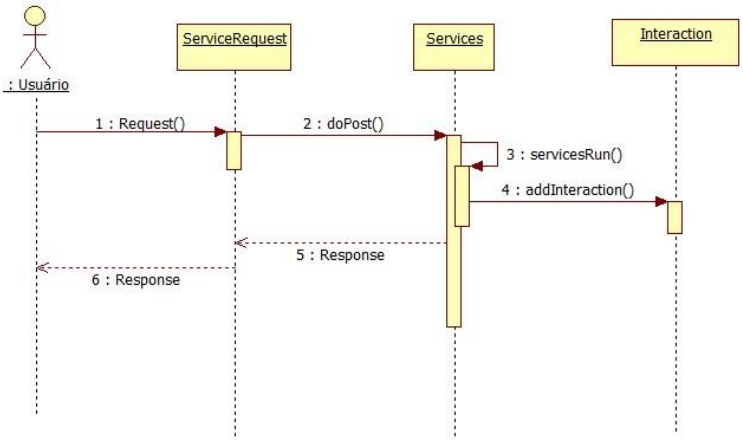
O *serviceRun()* verifica o serviço a ser realizado que neste caso é busca (*search*) e chama o método *search*() da classe *SearcherLucene* para realizar a busca do conteúdo requisitado. Este método acessa o índice, realiza a busca e responde ao usuário com os documentos que apresentam similaridade ao conteúdo buscado.

5.2.3 Inclusão de Interação

Este serviço permite armazenar as interações que o usuário realiza no sistema. A demonstração do seu comportamento pode ser

visualizada no diagrama de sequência (Figura 27), que demostra os objetos do sistema e os devidos comportamentos na execução.

Figura 27: Diagrama de sequência do serviço de inclusão de interação.



Fonte: Autores.

Toda interação de acesso/leitura ou avaliação de documentos que o usuário realiza através do sistema é enviada por meio de uma requisição ao *ServiceRequest* utilizando um objeto JSON. Para tal, primeiro se estabelece um canal de comunicação com a classe *Services* utilizando o protocolo HTTP. O método *doPost()* da classe *Services* recebe uma mensagem de requisição repassando o tipo de serviço a ser realizado pelo método s*erviceRun()*.

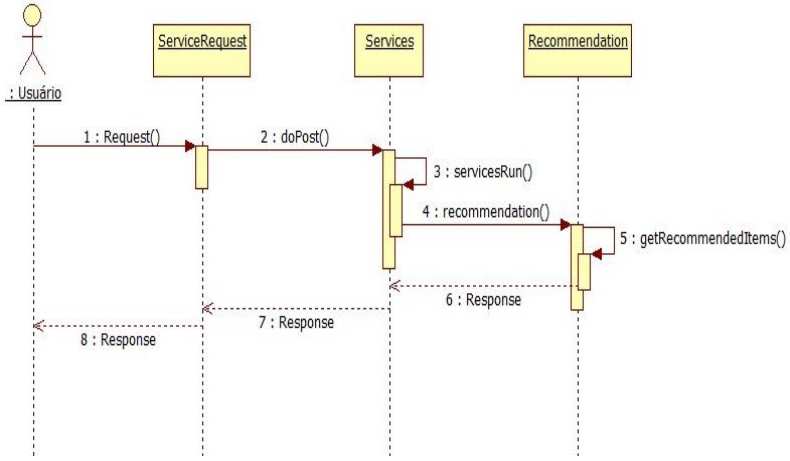
O *serviceRun()* verifica o serviço a ser realizado que neste caso é inclusão de interação (*addInteraction*) e chama o método *addInteraction*() para armazenar a interação realizada. Este método estabelece conexão com o banco de dados e armazena as interações na tabela interação (conforme modelo de dados apresentação no Capítulo 3).

5.2.4 Serviços de Recomendações

Este serviço realiza a recomendação de documentos textuais por meio da Recomendação Baseada em Filtragem Colaborativa e Recomendação Baseada em Conteúdo. Para melhor apresentar o comportamento destes serviços pode ser visualizado o diagrama de

sequencia (Figura 28), que demostra os objetos do sistema e os devidos comportamentos na execução.

Figura 28: Diagrama de sequência dos serviços de recomendação.



Fonte: Autores.

O usuário envia uma requisição ao *ServiceRequest* através de um objeto JSON estabelecendo um canal de comunicação com a classe *Services* por meio do protocolo HTTP. O método *doPost()* da classe *Services* recebe uma mensagem de requisição repassando o tipo de serviço a ser realizado pelo método s*erviceRun().*

O *serviceRun()* verifica o serviço a ser realizado que neste caso é Recomendação (*recommendation*) e invoca o método *getRecommendedItems().* Este método recebe um objeto JSON com o identificador do item buscado e o tipo de recomendação que será realizada (1-Filtragem Colaborativa, 2-Baseada em Conteúdo). O próximo passo é estabelecer uma conexão JDBC (*Java Database Connectivity*) com o banco de dados, pois este possui armazenada a tabela "*recomendacao*", que é utilizada para realizar a recomendação dos documentos.

5.3 CENÁRIO DE APLICAÇÃO

Para compor a base de dados do protótipo utilizou-se informações oriunda da base de dados da ScienceDirect®, disponíveis via seu web site através de acesso restrito (Periódicos da Capes). Ao todo foram coletados 305 documentos disponibilizados como texto completo em

formato PDF. Para permitir dados mais precisos na apresentação para cada documento foi elaborado um arquivo de metadados com identificador, título, palavras-chave e ano de publicação. A partir do metadados e o texto completo foi produzida a base de índices.

Os 305 documentos representam os itens possíveis de serem consultados ou avaliados durante a interação de usuários com a camada de aplicação. Com o objetivo de criar uma base com um número adequado de interações elaborou-se uma aplicação que artificialmente popula a base de interações. Foram geradas 100.000 sessões em que cada sessão possuía no mínimo 5 e no máximo 30 interações distintas (sem duplicação de itens) distribuídas aleatoriamente entre acesso ou avaliação de documento. Ao todo foram geradas 149.000 linhas na base de interações. O processo de geração de recomendações produziu ao todo 16.131 registros. O número reduzido em relação ao número de interações se justifica uma vez que as recomendações são baseadas em itens.

5.4 APRESENTAÇÃO DOS RESULTADOS

O funcionamento do protótipo ocorre em duas etapas, na primeira é realizada a busca que ocorre por meio da inserção de um termo a ser buscado e a partir deste termo serão listados os resultados encontrados. Na segunda etapa ocorrem as recomendações, que utilizarão como base o "*id*" de cada documento retornado pela busca, realizando as recomendações para cada um dos documentos listados. O fator que determina que documentos serão retornados é o peso agregado ao documento, ou seja, o peso que o documento recomendado apresenta em relação ao documento listado.

Para apresentar o funcionamento do protótipo foram realizadas três buscas, a primeira pelo termo "*Data Mining*" (Figura 29), a segunda pelo termo "*Biotechnology*" (Figura 30) e a terceira pelo termo "*Knowledge*" (Figura 31), sendo apresentados como resultado da consulta os três primeiros documentos.

Para cada documento o usuário possui três possibilidades de interação, sendo, o acesso ao texto completo do documento, a avaliação do documento pela indicação de uma quantidade de estrelas (*rating*), e a verificação das recomendações através da opção "*More Like This*". As duas primeiras possibilidades promovem o armazenamento da interação através do serviço de registro de interações.

O usuário ao selecionar a opção "*More Like This*" em qualquer documento, obtém um conjunto de documentos relacionados. São

apresentados seis documentos, sendo três referentes a recomendação baseada em conteúdo (serviço de consulta baseado em conteúdo) e três referentes a recomendação colaborativa (serviço de consulta de filtragem colaborativa).

Figura 29: Simulação com o termo de busca "*Data Mining*".

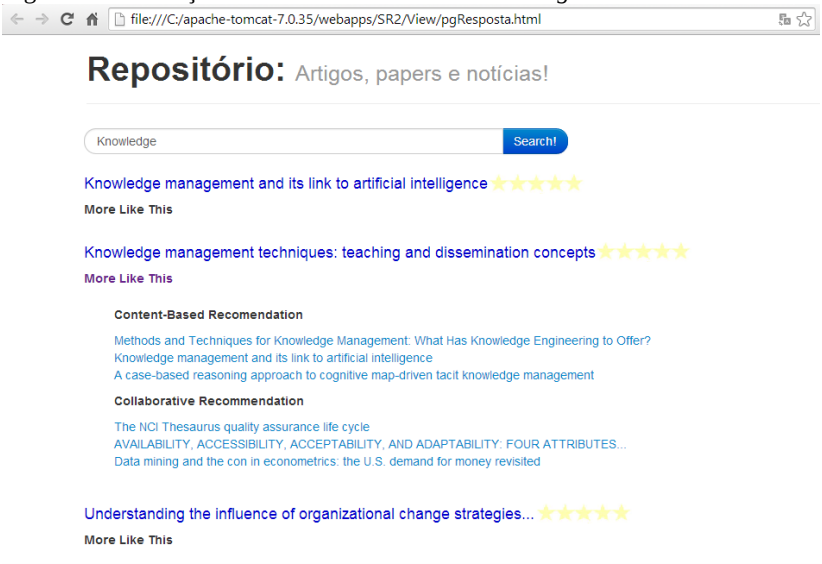
Fonte: Autores.

Abaixo estão apresentados a Figura 30 e Figura 31 referentes aos termos "*Biotechnology*" e "*Knowledge*".

Figura 30: Simulação com o termo de busca "*Biotechnology*".

Fonte: Autores.

Figura 31: Simulação com o termo de busca "*Knowledge*".



Fonte: Autores.

6 CONSIDERAÇÕES FINAIS

O objetivo geral deste trabalho consistiu proposição de uma arquitetura baseada em serviços para Recuperação de Informação e Sistemas de Recomendação.

A partir da revisão bibliográfica das áreas Recuperação de Informação e Sistemas de Recomendação, foi possível adquirir conhecimento para o desenvolvimento de um protótipo que comportasse a arquitetura proposta neste trabalho.

O foco da arquitetura proposta foi elaborar um conjunto de serviços capaz de armazenar as informações (documentos e interações) e prover respostas a partir da necessidade de consulta de determinado usuário.

Para permitir a avaliação da arquitetura foi elaborado um cenário de uso composto de 305 documentos, oriundos da *ScienceDirect*® e um conjunto de registros gerados artificialmente, totalizando 149.000 linhas na base de interações.

Também visando uma melhor apresentação dos serviços foi simulada uma *interface* que primariamente apresenta os resultados oriundos de uma busca, os dados apresentados nesta *interface* foram obtidos a partir da execução dos serviços em uma IDE Eclipse. Como resultado da busca, são retornados os documentos mais similares e para cada documento são possíveis dois tipos de interação, a leitura (acesso ao documento completo) ou a avaliação da relevância. As interações representam os dados necessários para a geração das recomendações, estas que na interface são acessadas através da opção "*More Like This*". Nesta opção, considerando determinado documento escolhido, são sugeridos três documentos através da abordagem baseada em conteúdo e três documentos por meio da abordagem colaborativa.

Os resultados obtidos demonstram que a arquitetura proposta é capaz de realizar adequadamente recomendações de documentos textuais nas duas abordagens de recomendação apresentadas, a colaborativa e a baseada em conteúdo. Também foi possível perceber que as recomendações tendem a facilitar a localização de documentos de interesse que vai além das respostas providas pelo serviço de busca textual.

No desenvolvimento deste trabalho foram observadas, algumas possibilidades de trabalhos futuros, como o desenvolvimento de uma interface gráfica que permita a interação do usuário com o protótipo, a qual daria suporte a uma interação amigável e intuitiva seguindo os padrões de usabilidade.

Adaptar a camada de serviços que atualmente está em *Servlet* para a arquitetura OSGI (*Open Services Gateway Initiative*), pois esta tecnologia possibilita a criação dos serviços em módulos dinâmicos, ou seja, caso um serviço precise ser atualizado não será necessário interromper o funcionamento de toda aplicação por que cada serviço possui seu próprio ciclo de vida.

Além disso, vislumbrasse o aprimoramento do modelo de dados proposto visando suportar outras formas de recomendação, principalmente aquelas relacionadas a dimensão tempo. Esta dimensão torna-se particularmente importante por criar meios de se avaliar a evolução do comportamento/perfil de um usuário, evitando assim sugerir itens que se tornam irrelevantes em função do tempo. Embora estas melhorias não tenham sido desenvolvidas, o protótipo foi planejado visando futuras melhorias.

REFERÊNCIAS

AGHAEI, Sareh; NEMATBAKHSH, Mohammad Ali; FARSANI, Hadi Khosravi. EVOLUTION OF THE WORLD WIDE WEB: FROM WEB 1.0 TO WEB 4.0. **International Journal Of Web & Semantic Technology**, v. 3, n. 1, jan. 2012.

BAEZA-YATES, Ricardo, RIBEIRO-NETO, Berthier. **Modern information retrieval**. Vol. 463. New York: ACM press, 1999.

BERGMAN, Michael K.. The Deep Web: Surfacing Hidden Value. **The Journal of Electronic Publishing**, v. 7, n. 1, set. 2001. Disponível em: <http://grids.ucs.indiana.edu/courses/xinformatics/searchindik/deepweb whitepaper.pdf>. Acesso em: 20 nov. 2013.

BERNERS-LEE, Tim. **The World Wide Web**: Past, Present and Future. 1996. Disponível em: http://www.w3.org/People/Berners-Lee/1996/ppf.html>. Acesso em: 09 out. 2013.

BOBADILLA, Jesús et al. Recommender Systems Survey. **Journal Knowledge-based Systems**, Amsterdam, v. 46, p.109-132, jul. 2013.

CARDOSO, Olinda Nogueira Paes. Recuperação de Informação. INFOCOMP - **Journal of Computer Science.** v. 2, p. 27-32, 2000.

CAZELLA, Sílvio César. **Aplicando a Relevância da Opinião de Usuários em Sistemas de Recomendação para Pesquisadores**. 2006. 180 f. Tese (Doutorado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006.

CAZELLA, Sílvio César; NUNES, Maria Augusta S. N. ; REATEGUI, Eliseu. **A Ciência da Opinião:** Estado da Arte em Sistemas de Recomendação. In: André Ponce de Leon F. de Carvalho; Tomasz Kowaltowski. (Org.). Jornada de Atualização de Informática-JAI 2010-CSBC2010. Rio de Janeiro: PucRIO, 2010, v. 1, p. 161-216.

CONCEIÇÃO, Álvaro William da. **Um sistema voltado ao armazenamento e recuperação de conteúdo textual de diferentes contextos**. 2013. 61 f. Monografia (Graduação) - Curso de Tecnologias

da Informação e Comunicação, Universidade Federal de Santa Catarina, Araranguá, 2013.

CHURCH, K. W.; GALE, W. A. Concordances for Parallel Text. **Proceedings of the Seventh Annual Conference of the UW Centre for the New OED and Text Research**. Oxford, England: 40-62 p. 1991.

FERNEDA, Edberto. **Recuperação de informação**: análise sobre a contribuição da ciência da computação para a ciência da informação. 2003. 147 f. Tese (Doutorado) - Curso de Ciências da Comunicação, Universidade de São Paulo, São Paulo, 2003.

FIGUEIRA FILHO, Fernando M.; GEUS, Paulo Lício de; ALBUQUERQUE, João Porto de. Sistemas de Recomendação e Interação na Web Social. In: **I WORKSHOP DE ASPECTOS DA INTERAÇÃO HUMANO-COMPUTADOR NA WEB SOCIAL**, Porto Alegre, p.24-27, 2008.

FOLTZ, Peter W.; DUMAIS, Susan T.. Personalized Information Delivery: An Analysis of Information Filtering Methods. **Communications Of The Acm.** New York, p. 51-60. dez. 1992.

FOX, Christopher. LEXICAL ANALYSIS AND STOPLISTS. In: FRAKES, William B.; BAEZA-YATES, Ricardo (Ed.). **Information Retrieval**: Data Structures & Algorithms. New Jersey: Prentice Hall, 1992. Cap. 7. p. 101-136.

GARCIA, Cássio Alan; FROZZA, Rejane. Sistema De Recomendação De Produtos Utilizando Mineração De Dados. **Tecno-lógica**, Santa Cruz do Sul, v. 1, n. 17, p.78-90, jan/jun. 2013. Disponível em: <http://online.unisc.br/seer/index.php/tecnologica/article/viewFile/3283/2692>. Acesso em: 06 abr. 2014.

GERONIMO, Alisson de Villa; ANACLETO, Matheus Medeiros. **Um Sistema de Recomendação de Conteúdo Suportado pela Computação Distribuída**. 2014. 87 f. TCC (Graduação) - Curso de Tecnologias de Informação e Comunicação, Universidade Federal de Santa Catarina, Araranguá, 2014.

Goldberg, David. et al. Using collaborative filtering to weave an information Tapestry. **Communications of the ACM**, New York, v.35, n.12, p. 61-70, Dec. 1992.

GONÇALVES, Alexandre Leopoldo. **Um modelo de descoberta de conhecimento baseado na correlação de elementos textuais e expansão vetorial aplicado à engenharia e gestão do conhecimento**. Florianópolis, SC, 2006. 196 f. Tese (Doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia de Produção.

HARMAN, Donna et al. INTERVETED FILES. In: FRAKES, William B.; BAEZA-YATES, Ricardo (Ed.). **Information Retrieval**: Data Structures & Algorithms. New Jersey: Prentice Hall, 1992. Cap. 3. p. 33-44.

HATCHER, Erik; GOSPODNETIC, Otis; MCCANDLESS, Michael. **Lucene in Action**. 2. ed. Stamford: Manning Publications, 2010.

HERLOCKER, Jonathan Lee. **Understanding and Improving Automated Collaborative Filtering Systems**. 2000. 148 f. Tese (Doutorado) - Curso de Philosophy, Faculty Of The Graduate School Of The University Of Minnesota, Minnesota, 2000.

JANNACH, Dietmar et al. **Recommender Systems**: An Introduction. New York: Cambrigde University Press, 2010. 335 p.

JONES, W. P; FURNAS, G. W. Pictures of relevance: A geometric analysis of similarity measures. **Journal of the American Society for Information Science**, 38: 420–442, 1987.

KURAMOTO, Hélio. **Sintagmas Nominais**: uma nova Proposta para a Recuperação de Informação. Datagramazero: Revista de Ciência da Informação, Rj, v. 3, fev. 2002.

LOPES, Gisele Rabello. **Sistemas de Recomendação para Bibliotecas Digitais sob a Perceptiva da Web Semântica**. 2007. 69 f. Dissertação (Mestrado) - Curso de Programa de Pós Graduação em Computação, Departamento de Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2007.

MANNING, Christopher D.; RAGHAVAN, Prabhakar; SCHÜTZE, Hinrich. **An Introduction to Information Retrieval**. Cambridge: Online Edition (c) 2009 Cambridge Up, 2009.

MIRANDA, Lígia Maria Café de. Elaboração de tesauros utilizando-se o programa de elaboração de tesouros em microcomputador (Tecer). **Revista de Biblioteconomia de Brasília**, Brasília, v. 18, n. 2, p.185-182, jul/dez. 1990.

MONTEIRO, Silvana Drumond; FIDENCIO, Marcos Vinicius. **As dobras semióticas do ciberespaço: da web visível à invisível**. Transinformação, v.25, p.35-46, 2013.

MOOERS, C. N. Zatocoding applied to mechanical organization of knowledge. American Documentation. v. 2, p. 20-32, 1951.

MORAIS, Sérgio Francisco dos Santos. **Sistemas de Recomendação em Rapid Miner**: um caso de estudo. 2012. 81 f. Dissertação (Mestrado) - Curso de Análise de Dados e Sistemas de Apoio à Decisão, Departamento de Faculdade de Economia, Universidade do Porto, Porto, 2012.

O'REILLY, Tim. What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. **International Journal of Digital Economics**, Munich, N.65, p. 17-37, mar 2007.

PRIMO, Alex . **O aspecto relacional das interações na Web 2.0**. E-Compós (Brasília), v. 9, p. 1-21, 2007.

REATEGUI, Eliseo Berni; CAZELLA, Sílvio César. Sistemas de Recomendação. In: XXV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 25., 2005, São Leopoldo. **A Universalidade da Computação:** Um Agente de Inovação e Conhecimento. São Leopoldo: SBC, 2005. p. 306 - 348.

RICCI, Francesco et al (Ed.). **Recommender Systems Handbook**. New York: Springer, 2011.

SALTON, Gerard; MCGILL, Michael J.. **Introduction to modern information retrieval.** New York: Mcgraw-hill, 1983.

SANTINI, R. M. **Os usuários e a desorganização da cultura**: os Sistemas de Recomendação e as consequências da classificação para os usos sociais da música na Internet. Tese de Doutorado em Ciência da Informação. Rio de Janeiro: UFF-IBICT, 2010.

SCHAFER, J. Ben et al. Collaborative Filtering Recommender Systems. In: BRUSILOVSKY, Peter; KOBSA, Alfred; NEJDL, Wolfgang (Ed.). **The Adaptive Web**: Methods and Strategies of Web Personalization. New York: Springer-verlag Berlin Heidelberg, 2007. Cap. 9. p. 299-333.

SCHAFER, J. Ben; KONSTAN, Joseph; RIEDL, John. 1999. Recommender systems in e-commerce**. InProceedings of the 1st ACM conference on Electronic commerce (EC '99)**. ACM, New York, NY, USA, 158-166.

SCHIESSL, José Marcello. **Descoberta de conhecimento em texto aplicada a um sistema de atendimento ao consumidor**. 2007. 106 f. Dissertação (Mestrado) - Universidade de Brasília, Brasília, 2007.

SILVA FILHO, Luiz Alberto da. **Mineração De Regras De Associação Utilizando Kdd E Kdt:: Uma Aplicação Em Segurança Pública**. 2009. 85 f. Dissertação (Mestrado em Ciência da Computação) - Instituto De Ciências Exatas E Naturais, Universidade Federal do Pará, Pará, 2009.

SINGHAL, Amit. Modern Information Retrieval: A Brief Overview. **Ieee Computer Society Technical Committee On Data Engineering.** p. 35-43. 2001.

SONAWANE, Amol. **Usando o Apache Lucene para Procura de Texto.** 2014. Disponível em: <https://www.ibm.com/developerworks/br/java/library/os-apache-lucenesearch/#ibm-pcon>. Acesso em: 18 maio 2014.

SOUZA, Renato Rocha. Sistemas de recuperação de informações e mecanismos de busca na web: panorama atual e tendências. **Perspectivas em Ciência da Informação,** Belo Horizonte, v. 11, n. 2, p.161-173, maio/ago 2006.

SRINIVASAN, Padmini . THESAURUS CONSTRUCTION. In: FRAKES, William B.; BAEZA-YATES, Ricardo (Ed.). **Information**

Retrieval: Data Structures & Algorithms. New Jersey: Prentice Hall, 1992. Cap. 9. p. 179-292.

TOFFLER, Alvin. **The third wave**. Bantam Books: New York, 1980.

VAN RIJSBERGEN, Cornelis. J. **Information Retrieval**. London: Butterworths, 1975. 147 p.

VIEIRA, Simone Bastos. Indexação automática e manual: revisão de literatura. **Revista Ciência da Informação**, Brasília, v. 17, n. 1, jan./jun. 1988. Disponível em: <http://repositorio.unb.br/bitstream/10482/12901/1/ARTIGO_IndexacaoAutomaticaManual.pdf> Acesso em: 13 nov. 2013.

WU, Chen. WSDL term tokenization methods for IR-style Web services discovery. **Science of Computer Programming**, n. 77, v. 3, p. 355-374, 2011.

XAVIER, Bruno Missi; SILVA, Alcione Dias da; GOMES, Geórgia Regina Rodrigues**.** Analise Comparativa De Algoritmos De Redução De Radicais E Sua Importância Para A Mineração De Texto. **Revista Eletrônica Pesquisa Operacional para o Desenvolvimento**, Rio de Janeiro, v. 5, n.1, p. 84-99, jan/abr. 2013. Disponível em: <http://www.dataci.es.gov.br/publicacoes_cientificas/publicacoes/189-2122-1-PB.pdf>. Acesso em: 19 jan. 2014.