Disciplinarum Scientia. Série: Naturais e Tecnológicas, Santa Maria, v. 19, n. 1, p. 87-103, 2018.
Recebido em: 04.06.2018. Aprovado em: 30.07.2018.

ISSN 2176-462X

DESENVOLVIMENTO DE UM SISTEMA DE RECOMENDAÇÃO PARA BIBLIOTECAS DIGITAIS[1]

DEVELOPMENT OF A RECOMMENDATION SYSTEM FOR DIGITAL LIBRARIES

Leonardo Antônio da Rosa Furlan[2], Alexandre de Oliveira Zamberlan[3], Sylvio André Garcia Vieira[3] e Ana Paula Canal[4]

RESUMO

Novos artigos aderem às Bibliotecas Digitais na mesma velocidade que são publicados. No entanto, torna-se difícil encontrar conteúdo de grande valia em meio a tamanha quantidade de informação. Neste contexto, este trabalho tem por objetivo desenvolver um sistema de recomendação de artigos acadêmicos do Google Acadêmico, com o uso de técnicas de recomendação baseada em conhecimento e colaborativa. O perfil do usuário é traçado a partir da extração dos dados do seu Currículo Lattes, dispensando o preenchimento de formulários longos no momento do cadastro. A metodologia ágil *Feature Driven Development* norteou o desenvolvimento deste trabalho que resulta em um sistema Web. Foi observado que é possível fazer várias combinações de técnicas de recomendação para alcançar o resultado desejado.

Palavras-chave: aprendizagem, coletores de documentos, Google Acadêmico, técnicas de recomendação.

ABSTRACT

New articles adhere to Digital Libraries at the same rate as they are published. However, it is difficult to find great value content among such a large quantity of information. In this context, the objective of this work is to develop a recommendation system of articles belonging to Google Scholar, with the use of knowledge-based and collaborative recommendation techniques. The user's profile is drawn from the data extraction of his Lattes Curriculum, avoiding the filling of long forms at the time of registration. The agile methodology Feature Driven Development guided the development of this work that results in a Web system. It was observed that it is possible to make several combinations of recommendation techniques to achieve the desired result.

Keywords: *learning, web crawlers, Google Scholar, recommendation techniques.*

[1] Trabalho Final de Graduação - TFG.

[2] Acadêmico do curso de Sistemas de Informação - Universidade Franciscana. E-mail: leonardo.furlan@live.com

[3] Colaboradores. Docentes do curso de Sistemas de Informação - Universidade Franciscana. E-mail: alexz@ufn.edu.br; sylvio@ufn.edu.br

[4] Orientadora. Docente do curso de Sistemas de Informação - Universidade Franciscana. E-mail: apc@ufn.edu.br

INTRODUÇÃO

O crescimento da quantidade de informação presente na Internet se dá de forma constante nas últimas décadas, facilitada pela evolução da tecnologia. A abundância de documentos digitais e a rápida introdução de novos serviços de *e-business* sobrecarregam o usuário, fazendo com que este tome decisões errôneas (RICCI; ROKACH; SHAPIRA, 2011). Com isso, cria-se um inconveniente no momento da escolha, seja por um livro, um filme ou qualquer outro produto.

As aplicações que crescem com a Web fizeram com que as bibliotecas do mundo cogitassem a possibilidade de serem digitais. As Bibliotecas Digitais democratizam o acesso à informação, unindo acervos e eliminando a barreira logística (MARTINS; SILVA, 2017). Ainda assim, esses repositórios muitas vezes possuem uma gama muito grande de arquivos. Para lidar com essa "sobrecarga de informação", a incorporação de sistemas de recomendação nos acervos pode ser uma solução.

Os Sistemas de Recomendação abrangem um conjunto de ferramentas e técnicas de software para produzir sugestões a um usuário. As recomendações têm relação com processos de tomada de decisão, como qual notícia ler ou qual produto comprar, e geralmente são direcionadas para usuário com pouca experiência em escolher entre vários itens (RICCI; ROKACH; SHAPIRA, 2011). Os primeiros Sistemas de Recomendação surgiram em meados da década de 90 (BARBOSA, 2014). Inicialmente, eles serviam para fazer indicações de filmes, livros e artigos acadêmicos, embora atualmente sejam mais utilizados em sites de comércio eletrônico.

O Google Acadêmico é usado por muitos pesquisadores como ferramenta de busca de artigos científicos, devido à simplicidade e ao nível de abrangência das pesquisas (GOOGLE ACADÊMICO, 2017). Estudantes de várias regiões do mundo são beneficiados com os artigos disponíveis neste acervo. Porém, nem sempre os artigos trazidos de uma pesquisa atendem à necessidade, fazendo com que a busca se torne exaustiva. A combinação de um sistema de recomendação com a ferramenta do Google pode ser de grande interesse de estudantes e pesquisadores.

O uso de redes sociais e portais de currículos facilita a descoberta das necessidades de cada usuário. O Currículo Lattes integra bases de currículos de grupo de pesquisas e instituições em um único portal de grande confiabilidade e abrangência, além da riqueza de informações (CNPQ, 2017). Nesse contexto, o objetivo geral deste trabalho é criar um sistema de recomendação Web de artigos acadêmicos disponíveis no Google Acadêmico para refinar uma pesquisa sobre artigos.

Para atingir o objetivo geral, têm-se os seguintes objetivos específicos: (1) Fazer a coleta de dados do Currículo Lattes do usuário. (2) Usar o Google Acadêmico como fonte de artigos acadêmicos para o usuário. (3) Aplicar as técnicas de recomendação colaborativa e baseada em conhecimento para exibir os resultados do Google Acadêmico, a partir da inferência de dados do perfil de usuário.

SISTEMAS DE RECOMENDAÇÃO

No cotidiano, indivíduos contam com recomendações de outras pessoas por meio de diálogos, jornais, revistas e livros (RESNICK; VARIAN, 1997). Os sistemas de recomendação ajudam o processo de indicação à medida que o torna mais eficaz. Em ambientes digitais, os sites de *e-commerce* adotam os sistemas de recomendação para encontrar os produtos mais adequados a seus clientes, uma vez que isso faz o lucro do negócio aumentar (ALVAREZ et al., 2016).

Segundo Cheng et al. (2016), um Sistema de Recomendação é visto como um sistema de pesquisa, o qual a instrução de entrada é feita a partir de informações encadeadas e de usuário, e o resultado é uma tabela com itens classificados. Depois de formulada a consulta, a recomendação busca situar o usuário em meio aos itens mais relevantes para ele. Resnick e Varian (1997) usam a expressão "sistemas de recomendação" por ser um termo genérico, e o defende por dois motivos. Em primeiro lugar, os "recomendadores" não podem colaborar explicitamente com os destinatários, que podem ser desconhecidos entre si. Em segundo lugar, as recomendações podem sugerir itens particularmente interessantes, além de indicar aqueles que devem ser filtrados.

Os Sistemas de Recomendação funcionam por meio de técnicas. Algumas podem usar dados simples, como avaliações de usuários em itens. Outras técnicas, como ontologias, são mais dependentes do conhecimento e, por isso, mais complexas. Dentre as mais consolidadas, pode-se citar a recomendação baseada em conteúdo, a filtragem colaborativa, a recomendação demográfica, a recomendação baseada em conhecimento, entre outras (RICCI; ROKACH; SHAPIRA, 2011). Este trabalho considera duas abordagens de sistemas de recomendação distintas: sistemas de filtragem colaborativa e sistemas de filtragem baseada em conhecimento, cuja finalidade é a recomendação.

RECOMENDAÇÃO COLABORATIVA

A técnica de Recomendação Colaborativa é a que apresenta as tecnologias mais maduras. Os sistemas de recomendação colaborativa agregam avaliações e reconhecem as semelhanças entre os usuários com base em suas classificações, gerando assim novas recomendações (RICCI; ROKACH; SHAPIRA, 2011).

Para realizar o processo de filtragem colaborativa, devem existir três etapas: (1) a representação dos dados de entrada, em que o usuário avalia alguns itens com intuito de demonstrar seus interesses e, conforme as avaliações vão sendo feitas, os dados vão sendo armazenados no banco de dados; (2) a formação de vizinhança, etapa em que o sistema compara o perfil do usuário alvo com o perfil dos demais usuários do sistema para identificar similaridade, tendo em vista o índice de similaridade válido para considerar vizinhos; e (3) a geração da recomendação, em que o sistema

recomenda itens ao usuário alvo com base nos itens que seus vizinhos mais gostaram (CAZELLA; NUNES; REATEGUI, 2010).

Uma das soluções utilizadas para a recomendação colaborativa consiste no uso de algoritmos que atuam sobre a base de usuário, denominados K-*Nearest Neighbors (KNN)*. Ricci, Rokach e Shapira (2011) citam três passos a serem seguidos para uso dessa solução: calcular o índice de similaridade em relação ao usuário alvo (métrica de similaridade); selecionar um subconjunto de usuários com índices de similaridade mais altos (vizinhos) para considerar na predição; e normalizar as avaliações e computar as predições ponderando as avaliações dos vizinhos com seus pesos.

O cálculo da similaridade (passo 1) é feito frequentemente com o uso da correlação de Pearson. A equação que representa esse cálculo é dada pela fórmula (1), descrita por Cazella, Nunes e Reategui (2010).

$$corr_{ab} = \frac{\sum_i (r_{ai} - \overline{r_a})(r_{bi} - \overline{r_b})}{\sqrt{\sum_i (r_{ai} - \overline{r_a})^2 \sum_i (r_{bi} - \overline{r_b})^2}} \quad (1)$$

Na equação, o $corr_{ab}$ é a correlação do usuário alvo a com um usuário *b*; r_{ai} é a nota que o usuário *a* atribuiu ao item *i*; r_a é a média de todas as avaliações do usuário *a* em comum com o usuário *b*; r_{bi} é a nota que o usuário *b* atribuiu ao item *i*; r_b é a média das avaliações que o usuário *b* possui em comum com o usuário *a*. Quando há mais de uma avaliação em comum, o resultado da equação normalmente varia entre 1 e -1, sendo 1 para similaridade total e -1 para similaridade oposta (CAZELLA; NUNES; REATEGUI, 2010).

No passo 2, Cazella, Nunes e Reategui (2010) estabelecem a ideia de que o subconjunto seja selecionado considerando um resultado de similaridade acima de 0,3. Esse subconjunto é então submetido a uma segunda equação para fazer o cálculo da predição, isto é, a nota que se supõe que o item a ser recomendado terá, cuja fórmula (2) é mostrada abaixo.

$$p_{ai} = \overline{r_a} + \frac{\sum_{b=1}^{n} (r_{bi} - \overline{r_b}) * corr_{ab}}{\sum_{b=1}^{n} |corr_{ab}|} \quad (2)$$

Sendo p_{ai} a predição de um item *i* para um usuário *a*; r_a é a média das notas que o usuário *a* deu aos itens que seus similares também avaliaram; r_{bi} é a nota que o usuário *b* atribuiu ao item *i*; r_b é a média das avaliações do usuário *b* em comum com o usuário *a*; $corr_{ab}$ é a correlação do usuário *a* com um usuário *b*.

Cazella, Nunes e Reategui (2010) identificaram algumas limitações quanto ao uso da técnica de recomendação colaborativa, como o problema do primeiro avaliador, o qual novos itens adicionados só serão recomendados para o usuário após serem avaliados por outros usuários. Há também o problema de pontuações esparsas: se o número de usuários for pequeno em relação à quantidade de itens no sistema, é possível que as pontuações se tornem muito esparsas. Um outro problema identificado

é o da própria similaridade, em que usuários que possuem preferências muito variantes enfrentarão dificuldade em encontrar usuários com gostos similares. Neste caso, a recomendação será pobre.

RECOMENDAÇÃO BASEADA EM CONHECIMENTO

A técnica de recomendação baseada em conhecimento não estima totalmente a utilidade de um item antes de recomendá-lo a um usuário. No entanto, aplicam-se heurísticas para pesar o tanto que um item é útil, através do conhecimento adquirido sobre o usuário (RICCI; ROKACH; SHAPIRA, 2011). Segundo Busatto (2013), o sistema que utiliza essa técnica retira informações do conhecimento adquirido sobre o usuário para definir suas necessidades. Porém, o grande problema de recomendação baseada em conhecimento é justamente adquirir o conhecimento (BUSATTO, 2013). De acordo com Felfernig et al. (2011), os engenheiros do conhecimento têm bastante dificuldade em transformar o conhecimento adquirido em representações formais e executáveis. Além disso, os usuários devem submeter-se ao preenchimento de formulários, a fim de evidenciar seus interesses. Sistemas que utilizam a técnica baseada em conhecimento não são capazes de fazer descobertas de interesse como na técnica colaborativa. Todavia, a técnica baseada em conhecimento auxilia a técnica colaborativa, visto que a primeira não tem problemas em iniciar do zero: as condições para ela iniciar são expostas em uma sessão de recomendação (FELFERNIG et al., 2011).

MATERIAL E MÉTODOS

O sistema desenvolvido atua coletando artigos já existentes no Google Acadêmico com base nas áreas de interesses especificadas no Currículo Lattes do usuário. Para retirar essas informações do arquivo, são usados recursos da linguagem de programação *Hypertext Preprocessor* (PHP). Para que seja possível reunir os resultados de uma página do Google Acadêmico e incorporar no sistema, é necessário fazer a utilização do conceito de *Web Crawler*. Além do mecanismo empregado na elaboração da regra de negócio, foram utilizadas diversas ferramentas para definições de interface e usabilidade.

A tradução do termo Web Crawlers seria “coletores de documentos”. Também conhecidos como robôs ou aranhas, os coletores de documentos são programas que baixam automaticamente conteúdo de páginas da Web (LIU, 2011). O uso de coletores de documentos é importante pois a Web não é uma coleção estática de páginas, mas sim uma entidade dinâmica. O papel do Web Crawler é ajudar aplicativos a manter seu repositório atualizado, visto que muitos links são acrescentados e excluídos da Web rapidamente. O uso dos rastreadores é mais difundido em apoio de motores de busca, tanto que os rastreadores são os principais consumidores da largura de banda da Internet. Eles coletam páginas para os buscadores a fim de alimentar seus índices. O Google e o Yahoo! fazem uso desta ferramenta (LIU, 2011).

O Currículo Lattes é adotado em todo o território nacional para registrar o progresso de estudantes e pesquisadores, memorizando dados como instituições frequentadas, áreas de atuação, especialidades do conhecimento, entre outros. Os dados de usuários armazenados têm sido usados por órgãos e instituições. A finalidade da análise consiste em avaliar o proveito e a capacidade de estudantes e pesquisadores na seleção de financiamentos (CNPQ, 2017).

O Google Acadêmico é uma ferramenta do Google que permite fazer pesquisas sobre literatura acadêmica, como artigos, teses, livros e resumos de universidades e repositórios online. Essa ferramenta mescla resultados de várias bases em um único buscador. O Google Acadêmico classifica os documentos do repositório pesando o texto de cada documento, o local de publicação, por quem foi escrito, e o número de citações que esse documento possui em demais publicações (GOOGLE ACADÊMICO, 2017).

O processo adotado para o desenvolvimento do Software foi o *Feature Driven Development* (FDD) e é uma metodologia ágil. Segundo Pressman (2011), o FDD segue algumas abordagens, como colaboração entre pessoas da equipe, gerência de complexidade e problemas de projetos baseado em funcionalidades, comunicação técnica através de meios verbais, gráficos e textos. Essa metodologia engloba cinco processos: (1) desenvolver um modelo abrangente, onde é feito um estudo sobre o domínio do negócio e a definição do escopo do projeto; (2) construir uma lista de funcionalidades que atendam às necessidades do cliente; (3) planejar através de funcionalidades, ordenando a lista gerada no processo anterior por prioridade; (4) projetar através de funcionalidades, na qual é definida uma atividade a ser realizada para cada funcionalidade; (5) construir através de funcionalidades, onde se produz o código para cada funcionalidade definida.

O sistema foi desenvolvido para plataforma Web, fazendo uso de linguagens de programação como *Hypertext Preprocessor* (PHP[5]), *Hypertext Markup Language* (HTML), *Cascading Style Sheets* (CSS) e Javascript, bem como a manipulação da linguagem *eXtensible Markup Language* (XML). O MySQL foi escolhido para armazenamento e recuperação de dados, e o layout das páginas foi facilitado pelo uso do *framework* Twitter Bootstrap[6]. O Twitter Bootstrap foi o escolhido para agregar equilíbrio de riqueza representacional das páginas. O *framework* traz um emaranhado de soluções em HTML, CSS e JavaScript, facilitando o desenvolvimento de projetos responsivos.

A linguagem HTML serve para representar o conteúdo de uma página Web e a forma como o navegador interpreta as informações, isto é, descrever a formatação dos elementos de uma página (IEPSEN, 2018). A linguagem CSS é utilizada para formatar o visual dos elementos da linguagem de marcação HTML (IEPSEN, 2018). A linguagem de programação PHP é voltada para a criação de páginas dinâmicas, atuando ao lado do servidor e sendo capaz de se conectar com um banco de dados. O Javascript é uma linguagem de programação interpretada, executada ao lado do cliente, que permite

[5] Disponível em: <http://php.net/manual/pt_BR/intro-whatis.php>.

[6] Disponível em: <http://getbootstrap.com>.

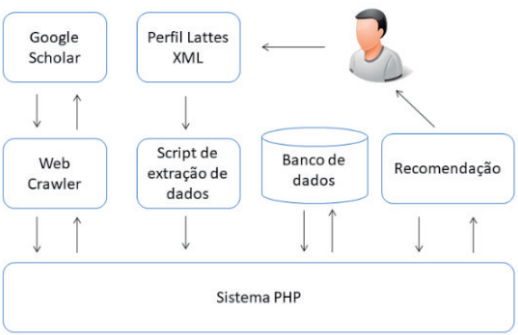
modificar o conteúdo, aparência e comportamento dos elementos de um website (IEPSEN, 2018). Segundo Hong et al. (2015), um arquivo XML é um documento legível para computadores e humanos, usado para tornar apropriada e simples a maneira como são armazenados e transferidos os dados entre aplicativos e ferramentas de software. O *Asynchronous* JavaScript *and* XML (Ajax) possibilita que uma aplicação Web faça solicitações assíncronas de informações ao servidor (WEI et al., 2017).

O Document Object Model (DOM) representa a forma como as *tags* de um documento HTML podem ser acessadas e manipuladas. Com o DOM, os dados são armazenados de forma hierárquica, na qual é possível distinguir os diferentes componentes de um objeto (IEPSEN, 2018). A extensão SimpleHTMLDOM[7] é uma classe PHP usada para facilitar o desenvolvimento de códigos de extração de conteúdo HTML. A classe contém um parser para o Modelo de Objetos para Documentos. O XAMPP é um pacote gratuito para desenvolvimento Web, constituído pelo servidor Apache, sistema gerenciador de banco de dados MySQL e interpretadores de linguagem PHP e Perl (APACHE FRIENDS, 2017). Seu uso para desenvolvimento do trabalho se dá na versão 1.8.3. Foi escolhido o sistema gerenciador de banco de dados MySQL (ORACLE CORPORATION, 2017) pela facilidade de uso e pela frequente presença em desenvolvimentos de sistema Web. É uma ferramenta gratuita, inclusa no pacote de instalação do XAMPP.

RESULTADOS E DISCUSSÃO

O sistema desenvolvido opera na plataforma Web. Seu layout é responsivo a fim de atender a experiência do usuário via celular. A atuação do website é focada em duas perspectivas: a primeira se refere ao dinamismo na extração dos dados do seu perfil Lattes para formulação da recomendação baseada em conhecimento; a segunda se refere à aplicação do conceito de recomendação colaborativa, onde o usuário receberá sugestões de artigos com base em usuários que tenham interesses semelhantes. No momento do cadastro, o sistema retira informações do Currículo Lattes do usuário através de um *script* PHP e então armazena no banco de dados. Através dos dados armazenados, o módulo Web Crawler recupera resultados de uma busca do Google Acadêmico e exibe no sistema, executando a recomendação ao usuário. A figura 1 ilustra a arquitetura do trabalho desenvolvido.

Figura 1 - Arquitetura do sistema proposto.

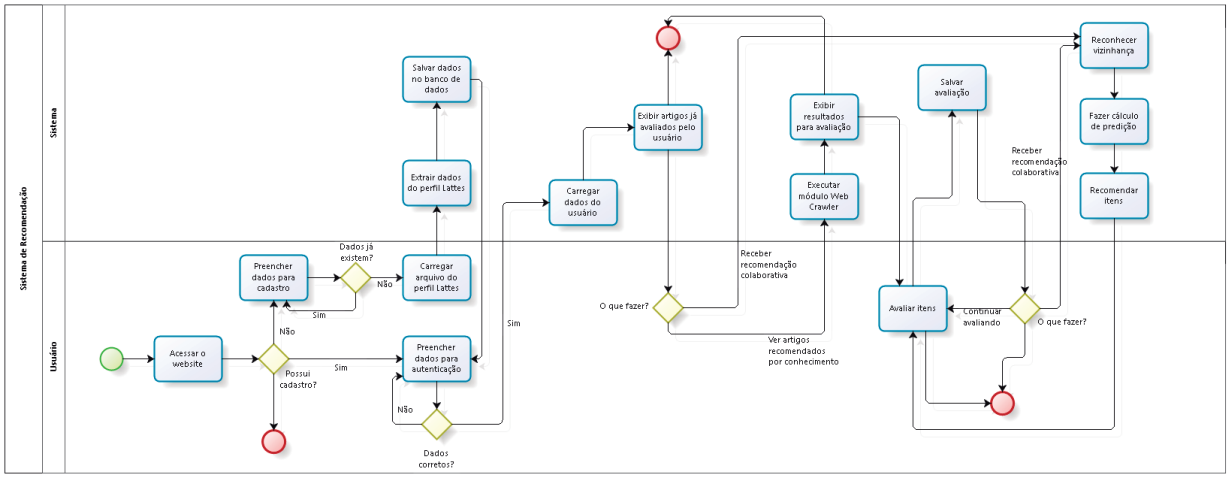


Fonte: construção do Autor.

[7] Disponível em: <http://simplehtmldom.sourceforge.net/>.

A figura 2 apresenta o diagrama de atividades, que compreende o fluxo de rotinas desenvolvidas para a interação do usuário com o sistema. Em um primeiro momento, o usuário acessa o *Website,* onde ele poderá fazer um cadastro ou fazer o *login,* caso já possua um cadastro. Ao fazer um novo cadastro, é necessário enviar o arquivo XML do Currículo Lattes. Os dados do currículo são extraídos e armazenados no banco de dados e então o usuário é direcionado novamente para a página inicial. Ao fazer *login*, o sistema exibe os artigos já avaliados pelo usuário e permite escolher entre receber recomendação baseada em conhecimento ou recomendação colaborativa. Na opção de receber recomendação baseada em conhecimento, o módulo Web Crawler é executado para mostrar os resultados. Na opção de receber recomendação colaborativa, são acionadas as funções de calcular a similaridade, reconhecendo a vizinhança. Após isso, é feito o cálculo da predição e são exibidos os resultados ao usuário. Em ambas as opções, é possível fazer avaliações sobre os itens exibidos.

Figura 2 - Diagrama de atividades.

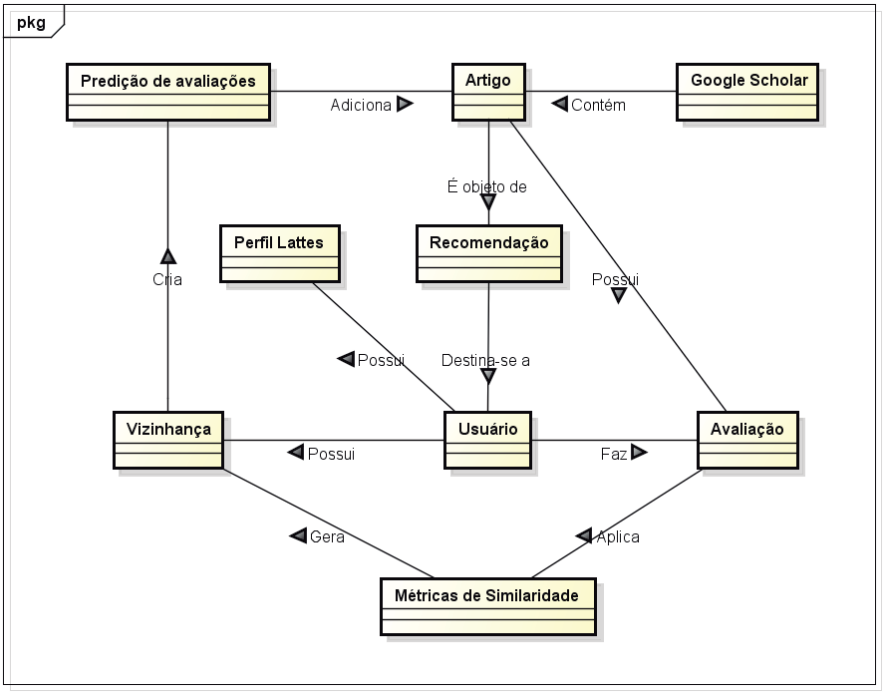


Fonte: construção do Autor.

O desenvolvimento do projeto seguiu as etapas da metodologia FDD. A definição do modelo abrangente se dá através do diagrama de domínio, ilustrado na figura 3. No diagrama, é possível ver as classes conceituais e as relações que elas possuem entre si.

Após a construção do diagrama de domínio, foi levantada uma lista de funcionalidades, que corresponde ao segundo passo da metodologia FDD. Nesta etapa, são analisadas as necessidades de desenvolvimento sob a visão de usuário e desenvolvedor. Com isto, foi possível elucidar os requisitos funcionais (RF) e os requisitos não funcionais (RNF) do sistema. Os requisitos funcionais definem tarefas e comportamentos que o sistema precisa ter, enquanto os requisitos não funcionais são as características do sistema em relação à usabilidade e suporte. No quadro 1 estão descritos os requisitos funcionais e, no quadro 2, os requisitos não funcionais.

Figura 3 - Diagrama de Domínio.



Fonte: construção do Autor.

Quadro 1 - Requisitos funcionais do sistema.

RF01 - Controle de Usuário: O sistema deverá gerenciar funções de cadastro e exclusão de usuário.	
Complexidade: baixa	**Relevância:** essencial
RF02 - Extração de dados: O sistema deverá extrair dinamicamente dados do usuário de seu perfil Lattes no formato XML.	
Complexidade: média	**Relevância:** essencial
RF03 - Coletor de documentos: O sistema deverá rastrear e exibir artigos do Google Acadêmico.	
Complexidade: média	**Relevância:** essencial
RF04 - Avaliação: O sistema deverá permitir avaliação dos artigos e armazenamento das avaliações.	
Complexidade: média	**Relevância:** essencial
RF05 - Recomendação por conhecimento: O sistema deverá fazer a busca de artigos com base nos dados adquiridos do usuário.	
Complexidade: baixa	**Relevância:** essencial
RF06 - Recomendação colaborativa: O sistema deverá sugerir itens a um determinado usuário, com base na similaridade com outros usuários.	
Complexidade: alta	**Relevância:** essencial
RF07 - Cálculo de similaridade: O sistema deverá fazer cálculo da correlação de Pearson para achar os vizinhos do usuário.	
Complexidade: alta	**Relevância:** essencial
RF08 - Cálculo de predição: O sistema deverá aplicar equação para calcular a nota predita de um artigo para determinado usuário.	
Complexidade: alta	**Relevância:** essencial

Fonte: construção do Autor.

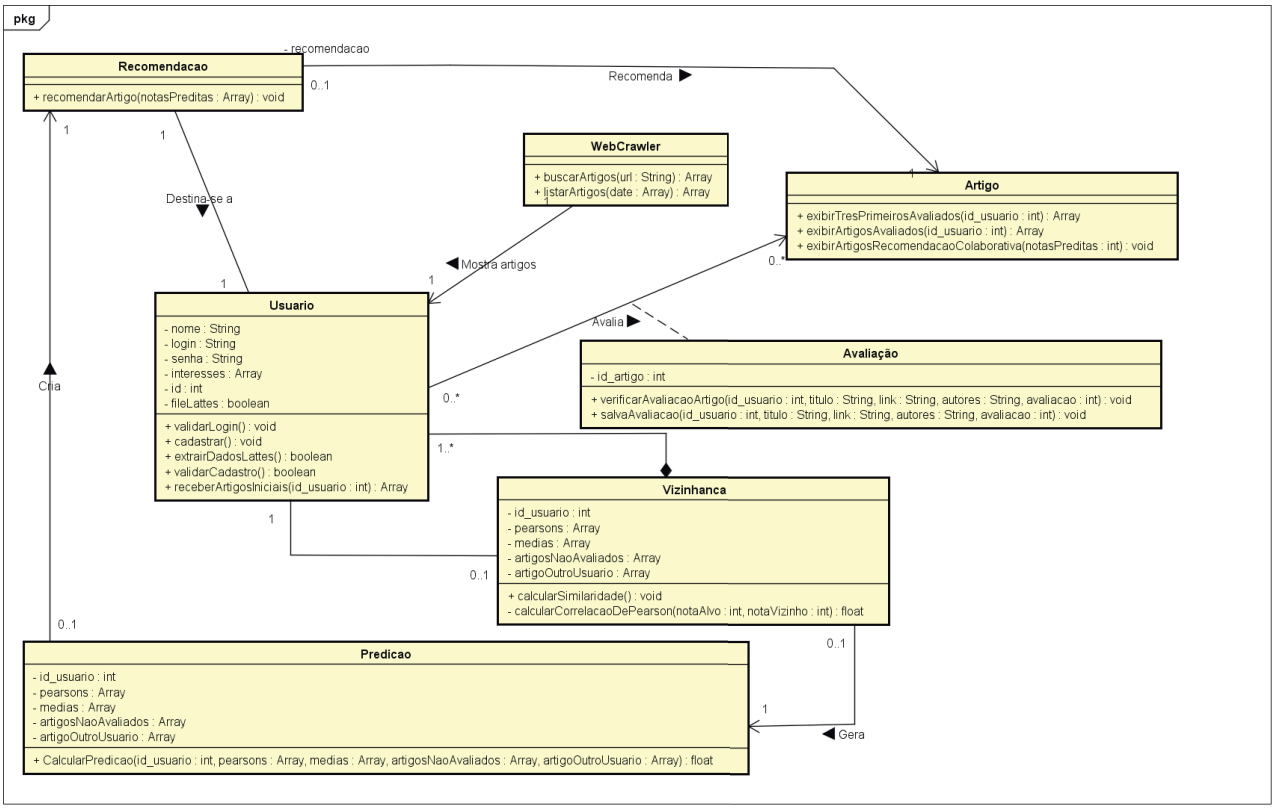
Quadro 2 - Requisitos não funcionais do sistema.

RNF01 - Linguagem de programação: O sistema será desenvolvido com o uso das linguagens PHP, CSS, HTML e *Javascript*.
RNF02 - Banco de dados: O banco de dados a ser utilizado é o *MySQL*.
RNF03 - Servidor *Web*: O sistema deverá funcionar no servidor Apache.
RNF04 - *Layout:* O *layout* do sistema deve utilizar o *Bootstrap*, a fim de manter a exibição responsiva.
RNF05 - Nota considerada: O sistema considera apenas artigos com nota igual ou maior que 3 para recomendação colaborativa.

Fonte: construção do Autor.

A partir dos requisitos funcionais, é possível fazer o diagrama que representa a visão estática do sistema. O diagrama de classes (Figura 4) mostra as classes envolvidas na implementação do sistema e seus relacionamentos. Cada classe apresenta os atributos e parâmetros que foram utilizados. Este diagrama representa a etapa de construção por funcionalidade da metodologia FDD.

Figura 4 - Diagrama de Classes.



Fonte: construção do Autor.

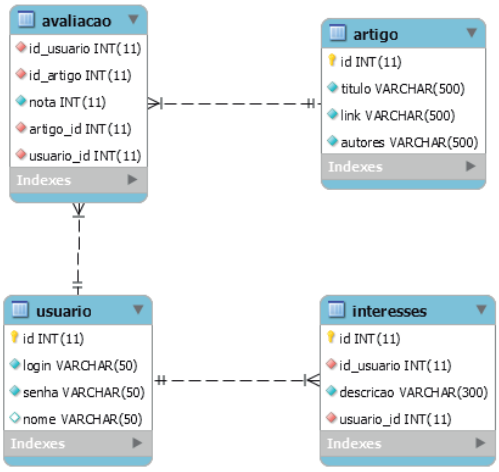
No Modelo Entidade-Relacionamento (MER) é possível visualizar todas as tabelas nas quais os dados serão armazenados, bem como seus atributos e suas relações entre si (Figura 5). A tabela ‘usuário’ se relaciona com a tabela ‘artigo’ de forma indireta, ou seja, uma entidade associativa está entre elas. A tabela ‘avaliação’ armazena a nota dada por um usuário a um determinado artigo. Já a tabela ‘interesses’ armazena a(s) área(s) de atuação do usuário, extraídas do perfil Lattes.

Quanto ao funcionamento, o usuário acessará o website onde encontrará opção de login e cadastro. Ao fazer o cadastro, é necessário informar o seu nome de usuário e senha, e enviar seu currículo da plataforma Lattes no formato XML. Após validar os dados do cadastro, o sistema fará a extração de dados relevantes do arquivo XML enviado, utilizando a classe *DOMDocument* e o recurso *DOMXPath*, que são componentes nativos do PHP e, então, armazenará no banco de dados.

O *DOMDocument* permite trazer o arquivo XML do perfil Lattes para dentro de um objeto PHP. Já o *DOMXPath* permite percorrer elementos XML e atributos. Os dados como nome e áreas de atuação do usuário estão localizados dentro de *tags* do arquivo, sendo possível localizar através

do *DOMXPath*. O algoritmo desenvolvido localiza a *tag* AREAS-DE-ATUACAO e dentro dela há nenhuma ou várias *tags* denominadas AREA-DE-ATUACAO. Dentro de cada AREA-DE-ATUACAO há atributos contendo dados, os quais o algoritmo trata em ordem de prioridade: primeiramente o atributo NOME-DA-ESPECIALIDADE, em segundo o NOME-DA-SUB-AREA-DO-CONHECIMENTO e em terceiro o NOME-DA-AREA-DO-CONHECIMENTO. Não havendo dados dentro do primeiro atributo com maior prioridade, ele passa para o segundo e, por fim, para o terceiro. A figura 6 mostra a estrutura do arquivo XML de um Currículo Lattes.

Figura 5 - Modelo Entidade-Relacionamento.



Fonte: construção do Autor.

Figura 6 - Estrutura do Currículo Lattes no formato XML.

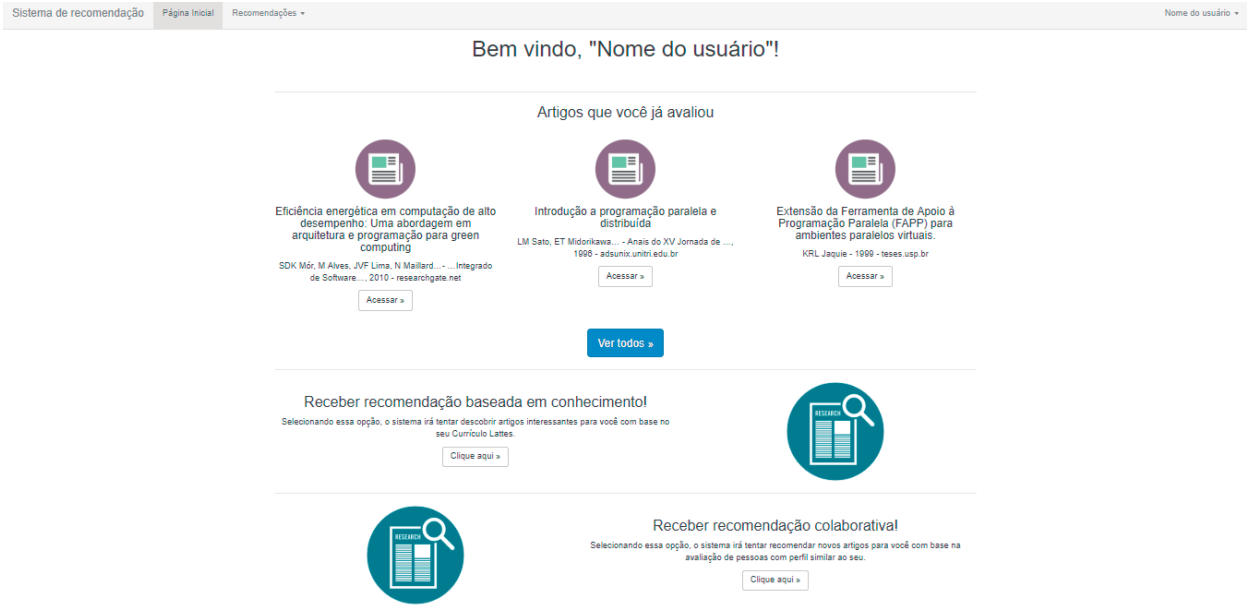
```
▼<AREAS-DE-ATUACAO>
   <AREA-DE-ATUACAO SEQUENCIA-AREA-DE-ATUACAO="1" NOME-GRANDE-AREA-DO-
   CONHECIMENTO="CIENCIAS_EXATAS_E_DA_TERRA" NOME-DA-AREA-DO-CONHECIMENTO="Ciência da
   Computação" NOME-DA-SUB-AREA-DO-CONHECIMENTO="Tecnologia da Informação" NOME-DA-
   ESPECIALIDADE="Programação Paralela"/>
   <AREA-DE-ATUACAO SEQUENCIA-AREA-DE-ATUACAO="2" NOME-GRANDE-AREA-DO-
   CONHECIMENTO="CIENCIAS_EXATAS_E_DA_TERRA" NOME-DA-AREA-DO-CONHECIMENTO="Ciência da
   Computação" NOME-DA-SUB-AREA-DO-CONHECIMENTO="Matemática da Computação" NOME-DA-
   ESPECIALIDADE=""/>
   <AREA-DE-ATUACAO SEQUENCIA-AREA-DE-ATUACAO="3" NOME-GRANDE-AREA-DO-
   CONHECIMENTO="CIENCIAS_EXATAS_E_DA_TERRA" NOME-DA-AREA-DO-CONHECIMENTO="Ciência da
   Computação" NOME-DA-SUB-AREA-DO-CONHECIMENTO="Metodologia e Técnicas da Computação" NOME-
   DA-ESPECIALIDADE="Interfaces Usuário Máquina"/>
 </AREAS-DE-ATUACAO>
```

Fonte: construção do Autor.

Para iniciar o processo de recomendação, é necessário saber quem é este usuário e quais são as suas preferências. O sistema desenvolvido trabalha com a forma de identificação de usuário no servidor que, de acordo com Cazella, Nunes e Reategui (2010), caracterizado por disponibilizar ao usuário um formulário de cadastro contendo informações pessoais como nome, sexo, endereço, entre outros. Essas informações são armazenadas em um servidor. Sempre que o usuário fizer o processo de *login*, são carregados conhecimentos adquiridos únicos daquele perfil. A figura 7 mostra a página inicial exibida para o usuário após fazer o *login*.

Após a identificação do usuário, é possível fazer a coleta de dados. O sistema desenvolvido trabalha com a coleta de dados de forma explícita. Segundo Cazella, Nunes e Reategui (2010), no modo de coleta explícita o usuário informa os dados de seu interesse através de preenchimento de questionários/formulários e/ou avaliações de itens. Ao fazer o *login*, o sistema irá carregar os dados do usuário, como nome, áreas de interesse e artigos já avaliados (se houver). A partir disso, o usuário contará com as opções de buscar novos artigos para avaliação com base no conhecimento adquirido pelo sistema, ou então, receber recomendação colaborativa.

Figura 7 - Tela exibida após o *login*.



Fonte: construção do Autor.

A opção de buscar novos artigos acionará o módulo *Web Crawler*. O módulo *Web Crawler* é uma classe que contém métodos de manipulação do DOM de um documento, herdados de uma segunda classe chamada "simple_html_dom.php". A classe "simple_html_dom.php" funciona como um parser e é abordada como uma extensão chamada *SimpleHTMLDOM*, desenvolvida a fim de simplificar os métodos nativos do PHP. Para utilizá-la, é preciso criar uma instância da mesma dentro da classe *Web Crawler*. É através dessa funcionalidade que o sistema fará uma varredura no Google Acadêmico, a fim de encontrar artigos da área de interesse do usuário.

As áreas de interesse do usuário formam um vetor de palavras. Estas palavras serão unidas com a URL[8] do Google Acadêmico, gerando assim o caminho da página que apresenta os resultados da pesquisa com base nas palavras informadas. Cada *website* presente na internet possui um arquivo HTML único. Porém, é possível identificar padrões que se repetem em mais de uma página do mesmo *website*. Após ser identificado esse padrão, o módulo *Web Crawler* conseguirá navegar entre os tags a fim de encontrar e exibir no sistema apenas os resultados convenientes. É importante afirmar que em nenhum momento o módulo manipula o site diretamente. Todos os elementos do site são trazidos

[8] Se refere ao endereço de rede no qual se encontra algum recurso informático.

para um objeto PHP através do *DOMDocument*, da mesma forma que são trazidos os arquivos XML, e então os elementos do objeto são percorridos.

Com isso, é abordado o conceito de filtragem baseada em conhecimento. À medida que o sistema rastreia os artigos, os resultados vão sendo exibidos com o título do artigo, autores e *link*, sendo possível inserir uma avaliação. A nota de avaliação deve variar de 1 a 5, onde 1 significa pouco interesse e 5 muito interesse. Esse dado é armazenado para que posteriormente possa ser feita a recomendação colaborativa. Quando um usuário avalia um artigo, o sistema verifica se o artigo já consta na base de dados, comparando o *link* do artigo. Caso não conste, é feita a adição dos dados do artigo e também os dados da avaliação. Caso o artigo já exista no banco de dados, apenas os dados da avaliação são armazenados, associando o identificador do artigo já existente. A figura 8 mostra a tela de artigos exibidos pela opção de receber recomendação baseada em conhecimento.

Figura 8 - Tela de Recomendação Baseada em Conhecimento.

Fonte: construção do Autor.

A recomendação colaborativa trará resultados apenas se atender as seguintes premissas: o usuário precisa ter feito avaliações de itens; o usuário precisa ter vizinhos, isto é, usuários com índices de similaridade aceitáveis. Para exemplificar a forma que ocorre o processo de recomendação colaborativa, a figura 9 mostra uma matriz contendo dados de avaliações e resultado da aplicação desses dados no cálculo da similaridade (Correlação de Pearson).

Figura 9 - Matriz de avaliações e resultados de similaridade.

	A1	A2	A3	A4	A5	A6	A7
U1	3	?	4	3	2	?	?
U2	4	5	4	4	2	4	
U3	5	1	1	5	4		3
U4	4	5	3	3	3		
U5	1	3		2			1
U6	3		3	2	2	3	2
U7	5	3	1		2	3	3

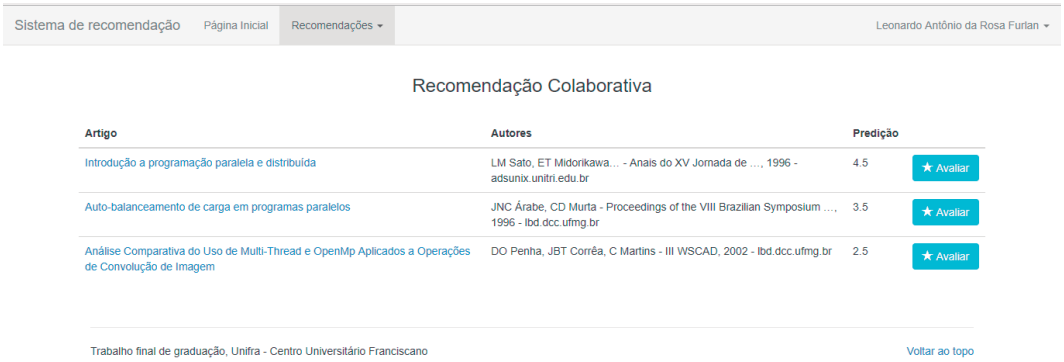
Usuário	Pearson (Similaridade)
U2	0,81649658092773
U3	-0,64699663922063
U4	0
U5	0
U6	0,70710678118655
U7	-0,24019223070763

Fonte: construção do Autor.

Na matriz da esquerda, os elementos com iniciais “U” seguidos de número são os usuários, os elementos com iniciais “A” são os artigos avaliados e os elementos com sinal de interrogação são as notas que o sistema precisa descobrir para o usuário alvo U1. A tabela da direita apresenta a similaridade entre o usuário alvo U1 e os usuários que tiveram pelo menos uma avaliação em comum e que avaliaram os itens que o usuário alvo não avaliou.

Partindo dos resultados obtidos com a aplicação da Correlação de Pearson, foram selecionados usuários com índices acima de 0,3 para considerar no cálculo da predição: apenas os usuários U2 e U6. A figura 10 mostra a tela da recomendação colaborativa. Nesta página, o usuário recebe as notas que o sistema supôs que ele daria com base em suas notas e nas notas de seus similares. É possível que o usuário faça uma avaliação nesses itens com outra nota.

Figura 10 - Layout da página de Recomendação Colaborativa.



Fonte: construção do Autor.

CONCLUSÕES

O principal objetivo do trabalho foi desenvolver um sistema de recomendação de artigos acadêmicos provenientes do Google Acadêmico. Vários conceitos e técnicas foram estudados para fundamentar a solução proposta. Buscaram-se alternativas às propostas que constam nos trabalhos relacionados. Foi visto que é possível fazer várias combinações de técnicas de recomendação para alcançar o resultado. Além disso, foi possível notar que a possibilidade de obter dados sobre o usuário com base no Currículo Lattes contorna o problema de preenchimento de formulários.

Diante disso, foi criado um sistema de recomendação que, embora não tenha sido hospedado, mostrou resultados funcionais no que tange a extração de dados do Currículo Lattes e as recomendações. A avaliação do sistema foi feita de forma qualitativa, por meio de testes realizados por um grupo de seis pessoas de forma local na máquina onde o sistema opera. Foi possível constatar que o cadastro e a recomendação baseada em conhecimento funcionam. Em geral, as pessoas que testaram o sistema acharam o cadastro diferenciado e as recomendações iniciais recebidas adequadas. O fato de poder acessar os artigos do Google Acadêmico diretamente do sistema chamou a atenção. Para os

testes da recomendação colaborativa, foi necessário projetar dados de avaliações feitas por usuários sobre um número limitado de artigos. O resultado dos testes se apresentou satisfatório, visto que os cálculos são feitos corretamente.

O trabalho teve algumas limitações impostas pelo Google Acadêmico. Uma delas é o bloqueio de requisições de buscas feitas pelo sistema. Normalmente, o bloqueio ocorre a partir de 300 solicitações de busca sem intervalo de tempo, embora não tenham sido feitos testes para comprovar um intervalo autorizado. Outra limitação ocorre no tratamento do tipo de codificação binária dos resultados recebidos do Google Acadêmico. Muitas tentativas foram feitas para corrigir o problema da acentuação, mas o sucesso foi apenas parcial. A terceira limitação é sobre os resultados exibidos do Google Books. O *link* dos livros muda a cada consulta. Então, embora os usuários estejam avaliando o mesmo livro, o sistema irá interpretar que são livros diferentes, visto que a comparação é feita com os *links*.

Por fim, sugere-se para trabalhos futuros encontrar soluções para o bloqueio do Google Acadêmico, colocando tempo de espera entre uma busca e outra, ou buscando alternativas de direcionamento quanto à identificação do endereço do servidor. Sugere-se a inclusão de um campo de busca manual. Assim, no caso de os resultados obtidos não serem satisfatórios, o usuário pode fazer uma busca pelas palavras digitadas por ele. Sugere-se também a adição de um campo para limitar os retornos por data, a fim de abranger os artigos mais recentemente adicionados no Google Acadêmico. Por último, deve-se ainda buscar formas de tratar os resultados exibidos do Google Books. Como este sistema foi desenvolvido como trabalho de conclusão de curso de graduação, pretende-se que sua hospedagem seja feita em breve em um dos servidores da Instituição.

REFERÊNCIAS

ALVAREZ, E. B. et al. Os Sistemas de Recomendação, Arquitetura da Informação e a Encontrabilidade da Informação. **Transinformação**, v. 28, n. 3, p. 275-286, 2016.

APACHE FRIENDS. **XAMPP Installers and Downloads for Apache Friends**. 2017. Disponível em: <https://bit.ly/1VqmUgH>. Acesso em: maio 2017.

BARBOSA, C. E. M. **Estudo de técnicas de filtragem híbrida em sistemas de recomendação de produtos.** 2014. 84p. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Federal de Pernambuco, Recife, PE, 2014.

BUSATTO, C. **O que tá valendo?** Um sistema Web de recomendação de eventos. 2013. 68p. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, 2013.

CAZELLA, S. C.; NUNES, M. A. S.; REATEGUI, E. B. A Ciência da Opinião: Estado da arte em Sistemas de Recomendação. In: XXX CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO - JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA (JAI), Belo Horizonte, 20 a 23 de julho de 2010. **Anais...** Belo Horizonte, 2010.

CHENG, H.-T. et al. Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. Boston, MA, EUA, 15 de setembro de 2016. **Anais...** Boston , 2016, p. 7-10.

CNPq. **Plataforma Lattes**. 2017. Disponível em: <http://lattes.cnpq.br/>. Acesso em: maio 2017.

FELFERNIG, A. et al. Developing constraint-based recommenders. In: RICCI, F.; ROKACH, L.; SHAPIRA, B.; KANTOR P. Ed. **Recommender systems handbook**. Boston, MA: Springer, 2011. p. 187-215.

GOOGLE ACADÊMICO. **About Google Scholar.** 2017. Disponível em: <https://bit.ly/2PvftKa>. Acesso em: maio 2017.

IEPSEN, E. F. **Lógica de Programação e Algoritmos com JavaScript**: uma introdução à programação de computadores com exemplos e exercícios para iniciantes. São Paulo: Novatec Editora, 2018.

HONG, T. et al. An ontology to represent energy-related occupant behavior in buildings. Part II: Implementation of the DNAS framework using an XML schema. **Building and Environment**, v. 94, p. 196-205, 2015.

LIU, B. **Web Data Mining.** 2. ed. Berlin: Springer, 2011.

MARTINS, D. L.; SILVA, M. F. Critérios de avaliação para sistemas de bibliotecas digitais: uma proposta de novas dimensões analíticas. **INCID Revista de Ciência da Informação e Documentação**, v. 8, n. 1, p. 100-121, 2017.

ORACLE CORPORATION. **MySQL**. 2017. Disponível em: <https://www.mysql.com/>. Acesso em: maio 2017.

PRESSMAN, R. S. **Engenharia de Software** - Uma Abordagem Profissional. 7. ed. Porto Alegre: AMGH Editora Ltda, 2011.

RESNICK, P.; VARIAN, H. R. Recommender Systems. **Communications of the ACM**, v. 40, n. 3, p. 56-58, 1997.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to Recommender Systems Handbook. In: **Recommender Systems Handbook**. 2. ed. Boston, MA: Springer, 2011.

WEI, J.-D. et al. Embedded-Based Graphics Processing Unit Cluster Platform for Multiple Sequence Alignments. **Evolutionary Bioinformatics**, v. 13, p. 1-10, 2017.