Desenvolvimento de um Sistema de Recomendação para Bibliotecas Digitais

Leonardo Antônio da Rosa Furlan[1], Ana Paula Canal[1]

[1]Curso de Sistemas de Informação – Centro Universitário Franciscano (UNIFRA) – Santa Maria – RS – Brasil

leonardo.furlan@unifra.edu.br, apc@unifra.br

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. In this context, the objective of this work is to develop a recommendation system of articles belonging to Google Scholar, through knowledge-based and collaborative recommendation techniques. In addition, the user's profile is drawn from the data extraction of his Curriculum Lattes, avoiding the filling of long forms at the time of registration. The FDD (Feature Driven Development) method 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 result.*

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. Neste contexto, este trabalho tem por objetivo desenvolver um sistema de recomendação de artigos acadêmicos do Google Acadêmico, através das técnicas de recomendação baseada em conhecimento e colaborativa. Além disso, 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. O método ágil FDD (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.*

1. Introdução

O crescimento da quantidade de informação presente na Internet se dá de forma constante nas últimas décadas, sendo facilitada pela evolução da tecnologia. A abundância de documentos digitais faz com que o indivíduo com pouca experiência pessoal de escolha, tenha que optar entre as várias alternativas que são apresentadas [Reategui e Cazella 2005]. Com isso, cria-se um inconveniente no momento da escolha, seja por um livro, um filme ou qualquer outro produto.

Considerando que a *Web* é a maior fonte de informação do mundo, surgiram então as Bibliotecas Digitais [Martins Júnior et al. 2011], que segundo Thomas e Karen (2001, apud Primo e Loh, 2006), são "coleções de recursos digitais selecionados de acordo com determinados critérios, organizados em alguma forma lógica, de modo acessível para a recuperação e distribuídas em redes de computadores". 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.

Burke (2007) define um sistema de recomendação como qualquer sistema que produz recomendação individual ou que tenha capacidade de levar o usuário para o objeto de interesse dentre várias opções. Em geral, as recomendações são feitas a partir de uma identificação de perfil de usuário e de suas preferências, que podem ser feitas de forma implícita ou explícita.

O uso de redes sociais e portais de currículos facilitam 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 sistema de recomendação que foi desenvolvido neste trabalho tem função de refinar uma pesquisa sobre artigos acadêmicos, de forma que os resultados exibidos sejam de interesse do indivíduo.

1.1 Justificativa

Sistemas de recomendação se diferem de mecanismos de busca simples pelo fato de abordarem os critérios: individualidade, utilidade e relevância [Burke 2007]. Leino e Räihä (2007) supõe que o crescimento do comércio eletrônico agrega em si a propagação dos sistemas de recomendação. Burke (2007) postula que esses sistemas são de notável apelo em ambientes onde a quantidade de informações é muito superior do que a eficiência dos indivíduos em pesquisar e visualizar todo o conteúdo. A recomendação entitula-se desejada não somente no *e-commerce*, como também em filmes, livros e artigos acadêmicos.

O Google Scholar é usado por muitos pesquisadores como ferramenta de busca de artigos científicos, devido ao nível de amplitude das pesquisas [Mugnaini e Strehl 2008]. 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.

1.2 Objetivo geral

Este trabalho tem por objetivo criar um sistema de recomendação *Web* de artigos acadêmicos disponíveis no Google Scholar.

1.3 Objetivos específicos

Para que o objetivo geral seja atingido, tem-se os seguintes objetivos específicos:

- Fazer a coleta de dados do Currículo Lattes do usuário;
- Usar o Google Scholar como fonte de artigos acadêmicos para o usuário;

- Aplicar as técnicas de recomendação colaborativa e baseada em conhecimento para exibir os resultados do Google Scholar, a partir da inferência de dados do perfil de usuário.

2. Referencial teórico

Esta seção aborda os conceitos relacionados a sistemas de recomendação, coleta de informações, estratégias dos sistemas de recomendação e as ferramentas e tecnologias utilizadas.

2.1 Sistemas de recomendação

Na vida cotidiana, indivíduos contam com recomendações de outras pessoas através de um diálogo, jornais, revistas e livros [Resnick et al. 1997]. Ainda segundo Resnick et al. (1997), os sistemas de recomendação ajudam o processo de indicação à medida que o torna mais eficaz. Reategui e Cazella (2005) afirma que os *websites* de comércio eletrônico empregam os sistemas de recomendação com objetivo de aumentar o lucro, aplicando diversas técnicas para encontrar os produtos mais adequados a seus clientes.

Os desenvolvedores dos primeiros sistemas de recomendação, cunharam a expressão "filtragem colaborativa" [Resnick et al. 1997], visando caracterizar um tipo de sistema específico no qual a seleção de informação era feita pela colaboração humana [Reategui e Cazella 2005]. Resnick et al. (1997) prefere utilizar 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, as recomendações podem sugerir itens particularmente interessantes, além de indicar aqueles que devem ser filtrados.

Burke (2007) especifica que os sistemas de informação precisam ter dados de base (informações que o sistema possui antes de o processo de recomendação iniciar), dados de entrada (informações que os usuários acrescentam para auxílio da recomendação), e um algoritmo que faça a união dos dados prévios e dos dados de entrada, resultando na recomendação. Este trabalho leva em consideração duas abordagens de sistemas de recomendação distintas: sistemas de filtragem colaborativa e sistemas de filtragem baseada em conhecimento, cuja finalidade é a recomendação.

As técnicas de recomendação doutrinam o funcionamento dos sistemas de recomendação, através da identificação de padrões para personalização de relacionamento com o usuário [Reategui e Cazella 2005]. Burke (2007) define cinco principais técnicas: recomendação colaborativa, recomendação baseada em conteúdo, recomendação demográfica, recomendação baseada em utilidade, recomendação baseada em conhecimento. A seguir são expostas as técnicas que foram utilizadas neste trabalho.

2.1.1 Recomendação colaborativa

Esta técnica é provavelmente a mais implementada e 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 [Burke 2007].

Segundo Queiroz (2003), para realizar o processo de filtragem colaborativa, devem existir três etapas: a representação dos dados de entrada, onde 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; a formação de vizinhança, etapa onde 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 a geração da recomendação, onde o sistema recomenda itens ao usuário alvo com base nos itens que seus vizinhos mais gostaram.

Uma das soluções mais utilizadas para a recomendação colaborativa, consiste no uso de algoritmos que atuam sobre a base de usuário, denominados KNN (*K-nearest neighbors*) [Sampaio 2006]. Reategui e Cazella (2005) cita 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 [Reategui e Cazella 2005]. A equação que representa esse cálculo é dada pela fórmula (1) mostrada abaixo, descrita por Cazella et al (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 et al. 2010].

No passo 2, Cazella el al. (2010) estabelece 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, nota que se supõe que o item a ser recomendado terá, cuja fórmula (2) descrita por Reategui e Cazella (2005) 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* [Reategui e Cazella 2005].

Cazella et al. (2010) identificou 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 a 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.

2.1.2 Recomendação baseada em conhecimento

Nesta técnica, a recomendação é feita a partir de inferências de preferências do usuário e suas exigências, através de conhecimento estruturado [Burke 2007]. Segundo Busatto (2013), o sistema que utiliza essa técnica retira informações do conhecimento adquirido sobre o usuário para definir suas necessidades.

Burke (2007) afirma que o perfil de usuário é qualquer estrutura capaz de inferir informações, como por exemplo uma consulta feita no Google. Porém, o grande problema de recomendação baseada em conhecimento é justamente adquirir o conhecimento [Busatto 2013]. Geralmente, a obtenção de informações é feita através de questionários, obrigando o usuário a informar aquilo que está buscando [Burke 2007]. Sistemas que utilizam a técnica baseada em conhecimento não são capazes de fazer descobertas de interesse como na técnica colaborativa. Todavia, ao ter o conhecimento estruturado, a precisão da recomendação é aumentada e as limitações da técnica de recomendação colaborativa são superadas [Burke 2007].

2.2 *Web Crawler*

A tradução do termo *Web Crawlers* seria coletores de documentos. Também conhecidos como robôs ou aranhas, são programas que baixam automaticamente conteúdo de páginas da *Web* [Liu 2011]. O uso de coletores de documentos é importante, visto que 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].

2.3 Currículo Lattes

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 capacidade de estudantes e pesquisadores na seleção de financiamentos [CNPq 2017].

2.4 Google Scholar

O Google Scholar, em português, 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* [Google Acadêmico 2017]. Essa ferramenta atua como meta-buscador, reunindo resultados de várias bases em um único buscador [Mugnaini and Strehl 2008]. Isso só é possível devido as bases que fornecem o conteúdo da busca autorizarem o acesso ao conteúdo, levanto a um aumento da leitura e citação dos trabalhos [Mugnaini and Strehl 2008].

O Google Acadêmico classifica os documentos do repositório pesando o texto de cada documento, o local de publicação, por quem foi escrito, e número de citações que esse documento possui em demais publicações [Google Acadêmico 2017].

2.5 *Feature Driven Development (FDD)*

Segundo Pressman (2011), o FDD segue algumas abordagens, como colaboração entre pessoas da equipe, gerencia de complexidade e problemas de projetos baseado em funcionalidades, e comunicação técnica através de meios verbais, gráficos e textos. Silva et al. (2009) afirma que essa metodologia engloba cinco processos. O primeiro é desenvolver um modelo abrangente, onde é feito um estudo sobre o domínio do negócio e a definição do escopo do projeto. O segundo consiste em construir uma lista de funcionalidades que atendam às necessidades do cliente. O terceiro é planejar através de funcionalidades, ordenando a lista gerada no processo anterior por prioridade. Projetar através de funcionalidades é o quarto processo, no qual é definida uma atividade a ser realizada para cada funcionalidade. O último processo é construir através de funcionalidades, onde produz-se o código para cada funcionalidade definida.

2.6 Outras ferramentas e tecnologias

O sistema foi desenvolvido na plataforma *Web*, fazendo o uso das linguagens de programação como PHP (*Hypertext Preprocessor)*, HTML (*Hypertext Markup Language*), CSS (*Cascading Style Sheets*) e *Javascript*, bem como manipulação da linguagem XML (*eXtensible Markup Language*). O MySQL foi escolhido para armazenamento e recuperação de dados, e o layout das páginas foi facilitado pelo uso do *framework* Twitter Bootstrap.

A linguagem HTML possui elementos denominados *tags* ou rótulos, que serve para orientar a forma como o navegador interpreta as informações, isto é, descreve a formatação dos elementos de uma página [Mendes 2004]. A linguagem CSS é utilizada como folhas de estilo para especificar o visual dos elementos da linguagem de marcação HTML [Bortolossi 2012]. Segundo Niederauer (2007), a linguagem de programação PHP – "*Hypertext Preprocessor"* – é voltada para criação de páginas dinâmicas, atuando no lado do servidor e sendo capaz de se conectar com um banco de dados.

O Javascript é uma linguagem de programação interpretada, executada do lado do cliente, que permite modificar de forma dinâmica o conteúdo e aparência de um *website* [Bortolossi 2012]. Segundo Mendes (2004), XML é uma linguagem de editoração que permite que as pessoas criem suas próprias linguagens. É neste ponto que a XML se diferencia da HTML, visto que a segunda possui quantidade predefinida de *tags* descrevendo seus elementos, e a XML permite que a pessoa defina suas próprias *tags*. O Ajax (*Asynchronous JavaScript and* XML) possibilita que uma aplicação Web faça

solicitações assíncronas de informações ao servidor, permitindo a adição de conteúdo à página que estamos trabalhando sem que seja preciso recarregá-la [Niederauer 2007].

O DOM (*Document Object Model*) é uma interface de programação para documentos XML e HTML, que representa a forma como as *tags* podem ser acessadas e manipuladas [Mendes 2004]. Com o DOM, os dados são armazenados de forma hierárquica, e com isso é possível distinguir os diferentes componentes de um objeto [Mendes 2004]. A extensão SimpleHTMLDOM [Sourceforge.net 2017] 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.

O Twitter Bootstrap [Bootstrap 2011] 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.

3. Trabalhos relacionados

O trabalho de Busatto (2013) trata do desenvolvimento de um sistema *Web* de divulgação e recomendação de eventos que acontecem em Porto Alegre. Os eventos são minerados pelo administrador do sistema através de um módulo que extrai informações de eventos de sites definidos. Essa extração é feita com uso de *Web Crawler.* O cadastro e *login* de usuário é feito utilizando a interface de programação de aplicativos (API) do *Facebook*, pela qual são extraídos os interesses do usuário presentes na rede social. Após a obtenção dos dados do usuário, a recomendação é feita através de diversos critérios e cálculos, exibindo os eventos mais relevantes. Diferente disso, o sistema desenvolvido neste trabalho obtém os dados do Currículo Lattes. Além disso, o *website* de Busatto permite que o usuário faça avaliações sobre dados que já foram adicionados no banco de dados pelo administrador, enquanto o trabalho proposto adiciona os itens no banco de dados apenas após a avaliação, sem a intervenção de administrador.

O trabalho de Barcellos et al. (2007) propõe o desenvolvimento de um sistema de recomendação de artigos acadêmicos. O usuário deve se cadastrar com login, senha e informar uma palavra-chave a qual será a base de seu interesse. Na mesma tela, é preciso também enviar seu currículo da plataforma Lattes no formato XML. O sistema disponibiliza um campo de pesquisa, onde as buscas são feitas diretamente no acervo do Google Scholar. O resultado é salvo em um arquivo no formato HTML, através de um executável chamado wget[1], e após o arquivo ser criado, o mesmo é carregado no *website.*

[1] É uma ferramenta de linha de comando não-interativa, podendo ser chamada por scripts. Seu objetivo é baixar arquivos e páginas da internet.

O protótipo funciona sob a técnica de filtragem de informação baseada em conteúdo, e sugere artigos que sejam de interesse do mesmo com base nas suas navegações anteriores. Já o sistema proposto neste trabalho, utiliza técnicas de recomendação colaborativa e baseada em conhecimento, além de receber os dados do Google Scholar através do conceito de *Web Crawler*.

Um sistema de recomendação para bibliotecas digitais é apresentado por Martins Júnior et al. (2011). Este trabalho considera qualquer biblioteca digital que provê metadados no formato Dublin Core[2] e que dê suporte ao protocolo Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH[3]). As técnicas de recomendação utilizadas são a filtragem colaborativa e a filtragem baseada em conteúdo, gerando assim, um sistema de recomendação com abordagem híbrida. As informações para recomendação baseada em conteúdo são obtidas por meio do Currículo Lattes do usuário, enviado no momento do cadastro. Já a recomendação colaborativa se dá através de avaliações sobre os documentos.

Ao ter os dados do perfil de usuário, são feitas requisições OAI-PMH para sites cadastrados no sistema. Com isso, é retornado um arquivo no formato XML com metadados de documentos digitais que atendam aos termos requisitados. Após isso, os dados no formado Dublin Core são extraídos do XML gerado e guardados na base de dados. A recomendação é feita então, e cada artigo exibido tem possibilidade de receber avaliação, para ser usado na filtragem colaborativa posteriormente. Diferentemente do sistema desenvolvido neste trabalho, o trabalho de Martins Júnior et al. (2011) não faz uso de recomendação baseada em conhecimento nem de *Web Crawler*. Ao passo que os trabalhos apresentam diferenças e similaridades, o objetivo final é o mesmo: a recomendação ao usuário.

Observa-se nos trabalhos relacionados que a identificação do perfil de usuário é fator determinante para a recomendação. Para que o usuário não perca muito tempo preenchendo formulários, os projetos apresentam proposta de extração de dados dinâmica de perfil de rede social e do Currículo Lattes. A recomendação é feita de forma discreta, de maneira que há uma melhora nas recomendações de acordo com o uso em colaboração.

4. Desenvolvimento e funcionamento do sistema

O sistema desenvolvido opera exclusivamente na plataforma *Web*, sendo seu *layout* responsivo a fim de atender a experiência do usuário via celular. A atuação do *website* é focada em duas perspectivas: a primeira, o dinamismo na extração dos dados do seu perfil Lattes para formulação da recomendação baseada em conhecimento; a segunda, a aplicação do conceito de recomendação colaborativa, onde o usuário receberá sugestões de artigos com base em usuários que tenham interesses semelhantes. A Figura 1 ilustra a arquitetura do trabalho proposto.

[2] Padrão de metadados utilizado para descrever objetos digitais, tais como, livros, artigos, entre outros.
[3] Protocolo utilizado para coleta de registros de metadados em repositórios.

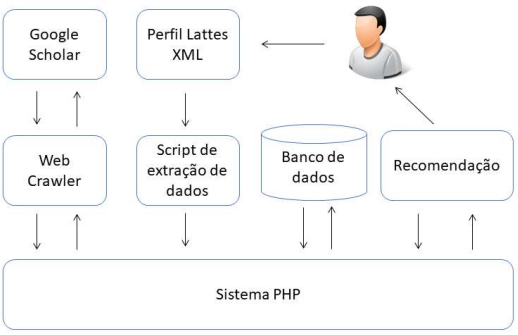


Figura 1. Arquitetura do sistema proposto

A Figura 2 apresenta o diagrama de atividades, para que possa ser entendido o fluxo do sistema.

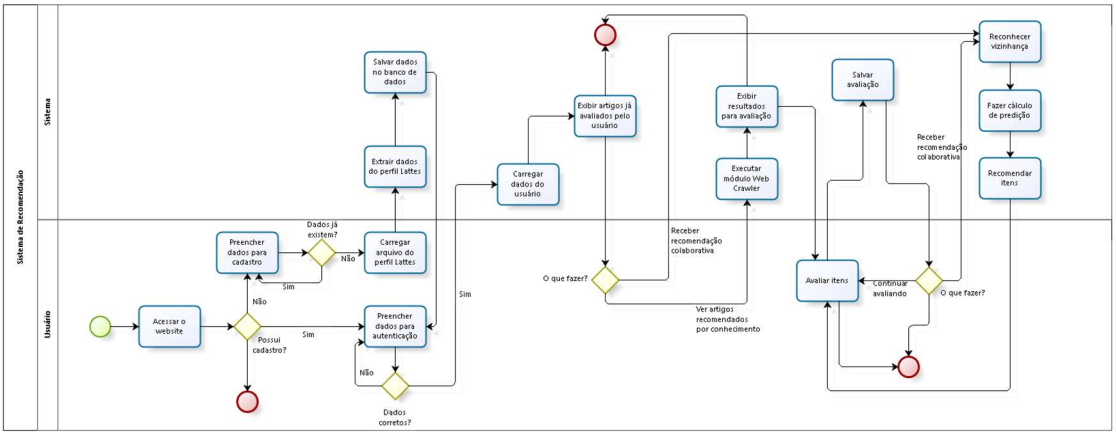


Figura 2. Diagrama de atividades

O desenvolvimento do projeto seguiu as etapas da metodologia FDD, visando mostrar de maneira detalhada os processos. 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 possuem entre si.

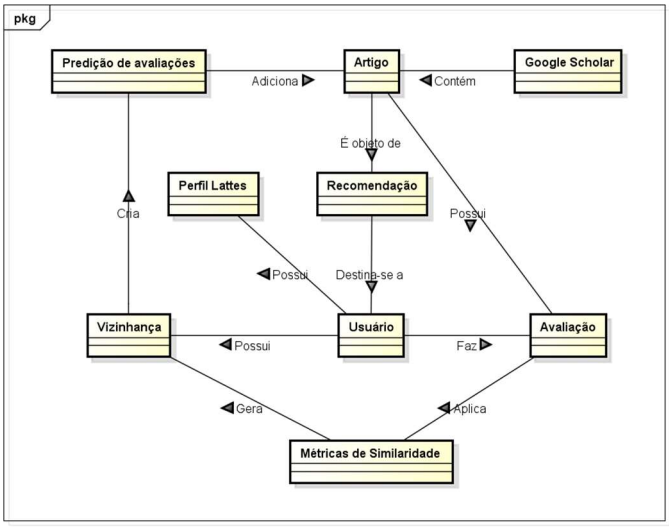


Figura 3. Diagrama de domínio

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 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 a usabilidade e suporte. No Quadro 1 estão descritos os requisitos funcionais, e no Quadro 2, os requisitos não funcionais.

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 Scholar.	
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

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.

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.

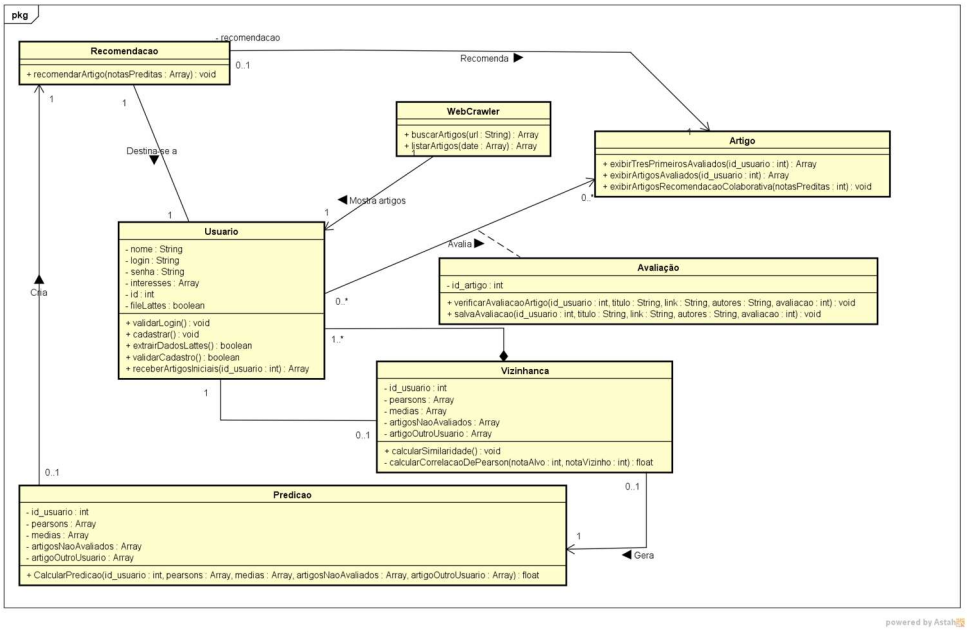


Figura 4. Diagrama de classes

No Modelo Entidade-Relacionamento (MER) é possível visualizar todas as tabelas nas quais os dados serão armazenados, bem como seus atributos e as relações entre si (Figura 5). A tabela ‘usuario’ se relaciona com a tabela ‘artigo’ de forma indireta, ou seja, uma entidade associativa está entre elas. A tabela ‘avaliacao’ 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.

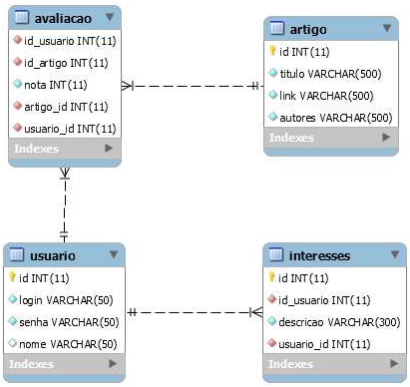


Figura 5. Modelo Entidade-Relacionamento

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 também 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, o qual 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.

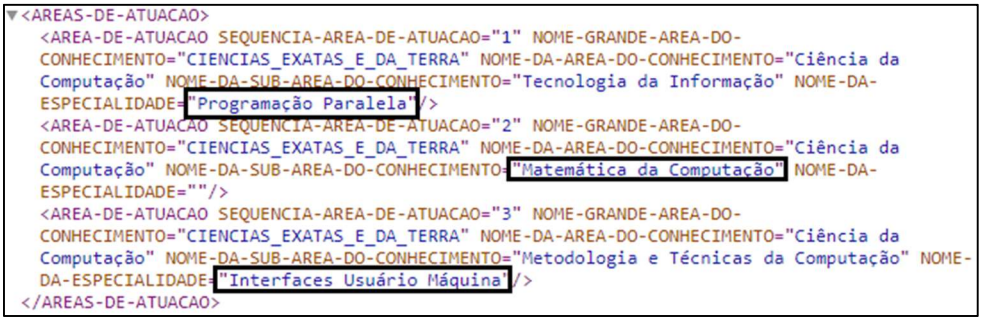


Figura 6. Estrutura do Currículo Lattes no formato XML

Para iniciar o processo de recomendação, é necessário saber quem é este usuário e quais as suas preferências. O sistema desenvolvido trabalha com a forma de identificação de usuário no servidor que, de acordo com Reategui e Cazella (2005), é quando é disponibilizado 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.

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 Reategui e Cazella (2005), 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.

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 Scholar, 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[4] do Google Scholar, 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 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 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 feito 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 feito 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 7 mostra a tela de artigos exibidos pela opção de receber recomendação baseada em conhecimento.

Figura 7. Tela de recomendação baseada em conhecimento

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 8 mostra uma matriz contendo dados de avaliações e resultado da aplicação desses dados no cálculo da similaridade (Correlação de Pearson).

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

	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

Figura 8. Matriz de avaliações e resultados de similaridade

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 os 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 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 9 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 9. Layout da página de recomendação colaborativa

5. Conclusões

O principal objetivo do trabalho foi desenvolver um sistema de recomendação de artigos acadêmicos provenientes do Google Scholar. Vários conceitos e técnicas foram estudados para fundamentar a solução proposta. Buscou-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 disto, 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. Em relação aos testes realizados por 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 Scholar. 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 é no tratamento do tipo de codificação binária dos resultados recebidos do Google Scholar. 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 como trabalhos futuros encontrar soluções para o bloqueio do Google Scholar, colocando tempo de espera entre uma busca e outra, ou buscando alternativas de direcionamento quanto a identificação do endereço do servidor. Sugere-se a inclusão de um campo de busca manual. Assim, em caso de os resultados obtidos não serem satisfatórios, o usuário pode fazer uma busca pelas palavras digitadas por ele. Por último, deve-se ainda buscar formas de tratar os resultados exibidos do Google Books.

Referências

Apache Friends (2017). XAMPP Installers and Downloads for Apache Friends. https://www.apachefriends.org/pt_br/index.html. Acessado em Maio de 2017.

Barcellos, C. D., Musa, D. L., Brandao, A. L. and Warpechowski, M. (2007). Sistema de Recomendação Acadêmico para Apoio a Aprendizagem. *Revista Novas Tecnologias na Educação, v. 5, n. 2.*

Bootstrap (2011). Bootstrap . The world's most popular mobile-first and responsive front-end framework. http://getbootstrap.com. Acessado em Maio de 2017.

Bortolossi, H. J. (2012). Criando conteúdos educacionais digitais interativos em matemática e estatística com o uso integrado de tecnologias: GeoGebra, JavaView, HTML, CSS, MathML e JavaScript. *Revista do Instituto GeoGebra Internacional de São Paulo, v. 1, n. 1, p. 28-36*.

Burke, R. (2007). Hybrid web recommender systems. *The adaptive web*, p. 377–408.

Busatto, C. (2013). O que tá valendo ? Um sistema Web de recomendação de eventos. Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul. Trabalho de Conclusão de Curso.

Cazella, S. C., Nunes, M. and Reategui, E. (2010). A Ciência da Opinião: Estado da arte em Sistemas de Recomendação. *CSBC XXX Congresso da SBC Jornada de Atualização de InformáticaJAI*, n. August 2016, p. 161–216.

CNPq (2017). Plataforma Lattes. http://lattes.cnpq.br/. Acessado em Maio de 2017.

Google Acadêmico (2017). About Google Scholar. https://scholar.google.com.br/intl/pt-BR/scholar/about.html, Acessado em Maio de 2017.

Leino, J. e Räihä, K.-J. (2007). Case Amazon: Ratings and Reviews as Part of Recommendations. *Proceedings of the 2007 ACM conference on Recommender systems - RecSys '07*, p. 137.

Liu, B. (2011). *Web Data Mining*. 2ª edição. Springer-Verlag Berlin Heidelberg.

Martins Júnior, H. N., Costa, E. B., Oliveira, T. T. M., Silva, A. P. and Bittencourt, I. I. (2011). Sistema de Recomendação Híbrido para Bibliotecas Digitais que Suportam o Protocolo OAI-PMH. *Em: XXII Simpósio Brasileiro de Informática na Educação, SBIE*, p. 140–149.

Mendes, A. (2004). *Programando com XML*. 1ª edição. Elsevier Editora.

Mugnaini, R. e Strehl, L. (2008). Recuperação e impacto da produção científica na era google: uma análise comparativa entre o google acadêmico e a web of science. *Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação*, v. 13, n. 1, p. 92–105.

Niederauer, J. (2007). *Web Interativa com Ajax e PHP*. 1ª Edição. Novatec Editora.

Oracle Corporation (2017). MySQL. https://www.mysql.com/. Acessado em Maio de 2017.

Pressman, R. S. (2011). *Engenharia de Software - Uma Abordagem Profissional*. 7ª edição. AMGH Editora Ltda.

Primo, T. e Loh, S. (2006). Técnicas de Recomendação para usuários de Bibliotecas Digitais. Simpósio Brasileiro de Sistemas de Informação, Curitiba, Paraná.

Queiroz, S. R. de M. (2003). Group Recommendation Strategies Based On Collaborative Filtering. Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Pernambuco, Recife. Dissertação de Mestrado.

Reategui, E. e Cazella, S. C. (2005). Sistemas de Recomendação. *XXV Congresso da Sociedade Brasileira de Computação A Universalidade da Computação Um Agente de Inovação e Conhecimento*, p. 306–348.

Resnick, P., Varian, H. R. and Editors, G. (1997). Recommender Systems. *Communications of the ACM*, v. 40, n. 3, p. 56–58.

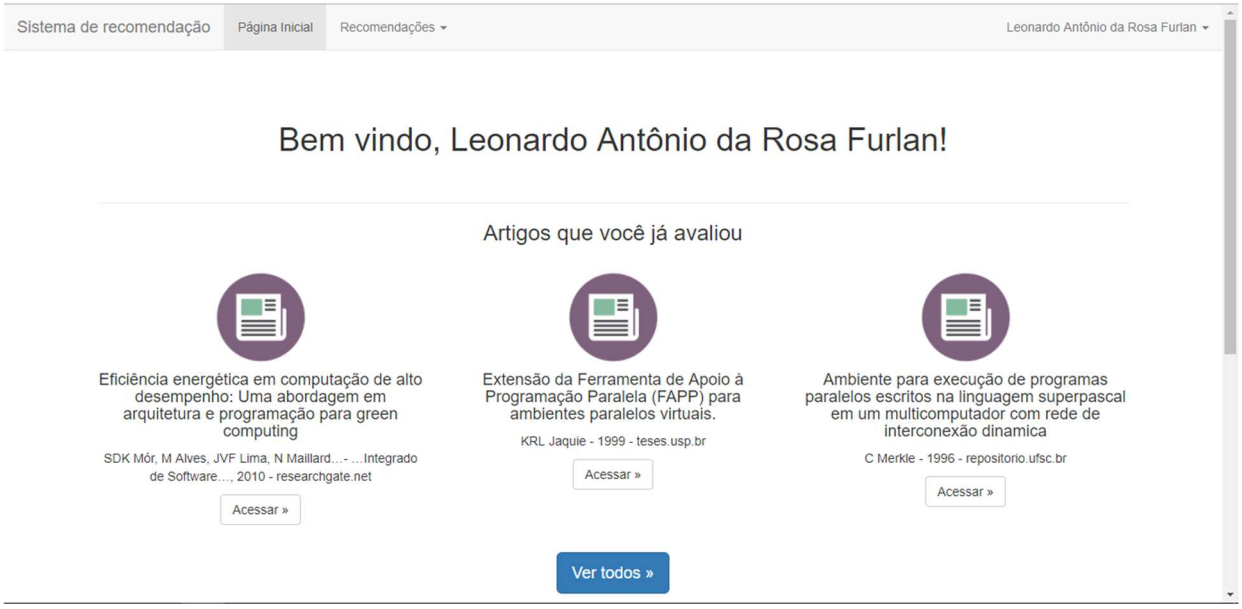
Sampaio, I. A. (2006). Aprendizagem Ativa em Sistemas de Filtragem Colaborativa. Pós-graduação em Ciência da Computação, Universidade Federal de Pernambuco, Recife. Dissertação de Mestrado.

Silva, F. G., Hoentsch, S. C. P. and Silva, L. (2009). Uma análise das Metodologias Ágeis FDD e Scrum sob a Perspectiva do Modelo de Qualidade MPS . BR. *Scientia Plena*, v. 5, p. 1–13.

Sourceforge.net (2017). PHP Simple HTML DOM Parser. http://simplehtmldom.sourceforge.net/. Acessado em Maio de 2017.

APÊNDICE A

Tela inicial do sistema após realizer login, acessada de um notebook.



APÊNDICE B

Tela inicial do sistema, acessada de um Smartphone Iphone SE.

APÊNDICE C

Tela de recomendação baseada em conhecimento, acessada de um Smartphone Iphone SE.

APÊNDICE D

Tela de recomendação colaborativa, acessada de um Smartphone Iphone SE.