

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA SUL-RIO-GRANDENSE - CAMPUS PASSO FUNDO
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET**

Gabriel Albrecht Bonato

**How2Code: Sistema de recomendação para o
ensino de Algoritmos**

Prof. Dr. João Mário Lopes Brezolin

Passo Fundo

2020

Gabriel Albrecht Bonato

**How2Code: Sistema de recomendação para o ensino de
Algoritmos**

Monografia apresentada ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-rio-grandense, Câmpus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Prof. Dr. João Mário Lopes Brezolin

Passo Fundo

2020

Gabriel Albrecht Bonato

How2Code: Sistema de recomendação para o ensino de Algoritmos

Trabalho de Conclusão de Curso aprovado em 10/11/2020 como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet.

Banca Examinadora:

Prof. Dr. João Mário Lopes Brezolin

Prof^a. Dr^a. Anubis Graciela de Moraes
Rossetto

Prof. Me. Rafael Marisco Bertei

**Passo Fundo
2020**

RESUMO

A disciplina de Algoritmos, nos cursos de Computação, em geral é o primeiro contato que o aluno tem com os conceitos de lógica e linguagens de programação. Devido à complexidade dos seus conteúdos, eles deparam-se com diversas dificuldades que muitas vezes podem levá-los à evasão do curso. Assim, considerando o processo individual de aprendizado de cada aluno, observou-se que a implementação de um sistema de recomendação personalizado poderia auxiliar no processo de aprendizagem dos alunos. Além disso, também apresenta a avaliação da funcionalidade de recomendação.

Palavras-chaves: Sistema de Recomendação; Estilos de Aprendizagem; Materiais de Aprendizagem; Recomendação de Materiais de Aprendizagem; Apache Mahout.

ABSTRACT

The Algorithms, in Computing courses, usually is the first contact that the student has with the concepts of logic and programming languages. Due to the complexity of its contents, the students face various difficulties that often can lead them to absteeism. Thus, considering the individual learning process of each student, it was observed that the implementation of a personalized recommendation system could help on their learning process. Also, presents the evaluation of the recommendation functionality.

Keywords: Recommendation Systems; Learning Styles; Learning Materials; Recommendation of Learning Materials; Apache Mahout.

LISTA DE FIGURAS

Figura 1 – Arquitetura e Funcionamento	20
Figura 2 – Tela de Login	21
Figura 3 – Página Inicial	21
Figura 4 – Tela de criação de conta	22
Figura 5 – Menu Lateral para usuário anônimo	23
Figura 6 – Menu Lateral para usuário logado	23
Figura 7 – Lista de Materiais de Aprendizagem sem recomendação	24
Figura 8 – Lista de Materiais de Aprendizagem com recomendação	24
Figura 9 – Visualização de Materiais de Aprendizagem para usuário anônimo	25
Figura 10 – Visualização de Materiais de Aprendizagem para usuário logado	25
Figura 11 – Diagrama de Fluxos do Controle de Recomendações	27
Figura 12 – Arquitetura do Sistema para Resposta à Requisição	27
Figura 13 – Personas Utilizadas para o Desenvolvimento do Sistema	28

LISTA DE ABREVIATURAS E SIGLAS

CCP	Coefficiente de Correlação de Pearson
CCS	Coefficiente de Correlação de Spearman
CVS	Cosseno de Vetor de Similaridade
EA	Estilo de Aprendizagem
GUI	Graphic User Interface
IES	Instituição de Ensino Superior
IFSUL	Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense
MAE	Mean Absolute Error
ML	Machine Learning
NMAE	Normalized Mean Average Error
PWA	Progressive Web App
RMSE	Root of Mean Square Error
SGBD	Sistema de Gerenciamento de Banco de Dados
SR	Sistema de Recomendação
TI	Tecnologia da Informação
TIC	Tecnologia da Informação e Comunicação
ZDP	Zona de desenvolvimento proximal

Sumário

1	INTRODUÇÃO	8
1.1	MOTIVAÇÃO	8
1.2	OBJETIVOS	9
1.2.1	Objetivos Específicos	9
1.3	ORGANIZAÇÃO DO TRABALHO	9
2	REFERENCIAL TEÓRICO	11
2.1	EVASÃO NOS CURSOS DE COMPUTAÇÃO	11
2.2	ESTILOS DE APRENDIZAGEM	12
2.3	PERSONAS	13
2.4	SISTEMAS DE RECOMENDAÇÃO	13
2.5	AVALIAÇÃO DE SISTEMAS DE RECOMENDAÇÃO	16
2.6	TRABALHOS CORRELATOS	17
3	IMPLEMENTAÇÃO DO SISTEMA	18
3.1	TECNOLOGIAS UTILIZADAS	18
3.1.1	Ionic Framework	18
3.1.2	MongoDB	18
3.1.3	Spring Boot	19
3.1.4	Apache Mahout	19
3.1.5	YouTube API	19
3.2	ARQUITETURA DO SISTEMA	20
4	TESTES E VALIDAÇÕES	29
5	CONSIDERAÇÕES FINAIS	31
	REFERÊNCIAS	32
	APÊNDICE A – BUSCA DE MATERIAIS PELA YOUTUBE API	35
	APÊNDICE B – ELABORAÇÃO DE CONJUNTO DE DADOS PARA O APACHE MAHOUT	37
	APÊNDICE C – ALGORITMO DE RECOMENDAÇÃO	41
	APÊNDICE D – ALGORITMO DE AVALIAÇÃO DO SR	42

1 INTRODUÇÃO

A evasão discente é um problema que afeta todos os níveis educacionais, no entanto, é algo bastante comum nos cursos de Computação, principalmente, referente à disciplina de Algoritmos.

Segundo Filho *et al.* (2007), a evasão de cursos superiores trata-se de uma questão global e não de algo que ocorre somente no Brasil. Ainda de acordo com o autor, “as perdas de estudantes que iniciam mas não terminam seus cursos são desperdícios sociais, acadêmicos e econômicos” (FILHO *et al.*, 2007). Desse modo, os investimentos nas IES não trazem os resultados esperados.

Conforme Bardagi e Hutz (2009), pode-se definir como evasão “a saída do aluno do curso de origem, sem concluí-lo”. Essa desistência pode ser causada por diversos motivos, conforme apontado por eles, dentre as razões estão o descontentamento com horários, falta de cursos noturnos, impossibilidade de trabalhar e estudar, mau relacionamento entre professor e aluno, pouca integração social à universidade, expectativas não correspondidas e falta de informações sobre curso e profissão, mau desempenho, reprovações e atraso, e problemas financeiros (BARDAGI; HUTZ, 2009).

Deste modo, o presente trabalho procura desenvolver um Sistema de Recomendação de Materiais de Aprendizagem para a disciplina de Algoritmos I do curso de Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense (IFSUL), campus Passo Fundo, de forma que venha a auxiliar no aprendizado dos discentes, estabelecendo novos meios de contato com os conteúdos propostos em aula. Objetiva-se, nesse sentido, proporcionar um recurso que motive os estudantes e evite a sua evasão do curso.

1.1 MOTIVAÇÃO

A disciplina de Algoritmos dos cursos de Computação, em geral, é o primeiro contato com os conceitos de lógica e linguagens de programação. Devido a complexidade do conteúdo da disciplina os alunos deparam-se com diversas dificuldades no estudo dos conteúdos propostos.

Essas dificuldades muitas vezes podem levá-los a evadir do curso em que estão matriculados, trazendo tanto impactos econômicos para as Instituições de Ensino Superior (IES) como sociais. Assim, considerando o processo individual de aprendizado de cada aluno, muitas vezes, faz-se necessário oferecer recursos para que o aluno possa estender os seus estudos para além do período de aula.

Atualmente, a evasão discente dos cursos de Computação possui índices bastante elevados, sendo um dos motivos que induzem os alunos a desistência do curso, as dificuldades que eles vêm a enfrentar com os conteúdos das disciplinas de Algoritmos. Desse modo, o desenvolvimento da aplicação visa auxiliar a sanar as dúvidas desses alunos, fornecendo-lhes outro meio que contribua para o desenvolvimento da sua aprendizagem, além de auxiliar na proposição de novas metodologias ensino.

1.2 OBJETIVOS

Desenvolver uma aplicação que objetiva auxiliar no aprendizado dos alunos de Algoritmos a partir da recomendação de materiais de aprendizagem. Essa aplicação deve sugerir itens baseando-se na similaridade das avaliações de materiais de aprendizagem realizadas pelos usuários do sistema, a fim de identificar perfis semelhantes, principalmente, a semelhança entre os perfis de usuário dos alunos do IFSUL.

1.2.1 Objetivos Específicos

- Realizar a revisão bibliográfica sobre a evasão de alunos dos cursos de Computação e os sistemas de recomendação de materiais de aprendizagem;
- Produzir, com o Framework Apache Mahout, uma interface que permita a avaliação e realize a recomendação dos materiais de aprendizagem;
- Desenvolver uma *Graphic User Interface* (GUI) para a aplicação através do Framework Ionic;
- Elaborar um serviço de API Rest com o Framework Java Spring Boot que permita a comunicação entre a GUI e os controladores de dados e informações;
- Desenvolver uma interface por meio da linguagem de programação Java, com a utilização da biblioteca YouTube API, que permita a busca por materiais de aprendizagem;
- Criar, através do Framework Java Spring Data, um meio que realize a persistência e a recuperação de informações no banco de dados MongoDB;
- Avaliar o algoritmo de recomendação baseando-se na massa de dados gerada pelos alunos do curso de Computação do IFSUL, campus Passo Fundo.

1.3 ORGANIZAÇÃO DO TRABALHO

O presente trabalho é constituído por capítulos, o que possibilita melhor organização e compreensão dos temas abordados.

O Capítulo 2 apresenta os conceitos utilizados no desenvolvimento do sistema proposto neste trabalho, fazendo, inicialmente, uma revisão bibliográfica a respeito da evasão nos cursos de computação, dos conceitos de aprendizagem, estilos de aprendizagem, personas, *Machine Learning* e sistemas de recomendação, e como funciona a recomendação de materiais de aprendizagem.

O Capítulo 3 aborda assuntos relacionados à implementação do sistema proposto, iniciando pelas tecnologias utilizadas para o desenvolvimento, a arquitetura e o funcionamento do sistema, e, também, os problemas e as dificuldades enfrentados durante o desenvolvimento.

O Capítulo 4 trata da explicação dos testes e validações realizados no sistema e apresenta os resultados desses processos. E, para finalizar, o Capítulo 5 aborda as considerações finais da execução do presente trabalho.

2 REFERENCIAL TEÓRICO

Neste capítulo serão abordados os temas a respeito da evasão nos cursos de computação, o que é a aprendizagem, os estilos de aprendizagem, o que são personas, a técnica de *Machine Learning* e o que são sistemas de recomendação de materiais de aprendizagem.

2.1 EVASÃO NOS CURSOS DE COMPUTAÇÃO

A evasão ocorre para todas as áreas de conhecimento, porém a área de Exatas ganha destaque por possuir os índices mais altos (MORÃES; POMBEIRO, 2016), pois, de acordo com Rodrigues (2013), os alunos chegam à graduação com pouca base matemática, devido às deficiências que há nos níveis fundamental e médio, o que diminui o interesse dos alunos pela área, causando-lhes frustrações, podendo até mesmo causar impactos irreversíveis a eles. Nos cursos de computação, muitas vezes a evasão ocorre devido às dificuldades com o raciocínio lógico requerido nas disciplinas de programação (ALVES; JAQUES, 2014).

De acordo com Rodrigues (2013), em 2013 uma pesquisa da IDC Brasil apontou que o país era o 4º maior mercado de Tecnologia da Informação e Comunicação (TIC) do mundo, e o 7º maior mercado de Tecnologia da Informação (TI). Além disso a projeção aponta que ele alcançará a 3ª posição do mercado de TIC, e a 5ª posição do mercado de TI até o ano de 2022.

Desse modo, surge como agravante das consequências advindas da evasão dos cursos de Computação, o crescimento do mercado de tecnologia. Dessa forma, há uma grande dificuldade de os cursos acompanharem o crescimento da área. “No cenário de 2008 ter-se-ia uma projeção de 296 mil vagas abertas em TI enquanto menos de 50 mil concluíram os cursos da área de Exatas” (GOMES, 2012). E ainda, consoante Rodrigues *apud* Sgobbi (2013), “a carência por profissionais pode ser justificada pela desistência dos alunos dos cursos da área, como também, o fato da baixa concorrência por vagas”.

Somando isso à alta taxa de evasão dos cursos da área, surge a necessidade de encontrar soluções que venham a melhorar o aprendizado dos alunos quanto aos conteúdos de suas disciplinas, em especial as de Algoritmos, por serem disciplinas introdutórias e por necessitarem de raciocínio lógico elevado.

Conforme Costa, Aguiar e Magalhães (2013), “o crescente uso de tecnologia de informação e comunicação possibilita mudanças a cada dia no processo de ensino e aprendizagem”, facilitando o desenvolvimento de novas ferramentas digitais que visam

auxiliar nesse processo.

Dessa forma, o sistema a ser desenvolvido surge como uma proposta para auxiliar os alunos no estudo dos conteúdos de Algoritmos propostos em sala de aula, buscando estabelecer um recurso que colabore para minimizar o problema da evasão.

2.2 ESTILOS DE APRENDIZAGEM

O estudo do processo de aprendizagem permeia o trabalho de diversos autores. Segundo Vygotsky (1998), o aprendizado é um elemento propulsor do desenvolvimento do sujeito que o capacita não somente para a realização das operações solicitadas para o aprendizado, mas o instrumentaliza para utilizar esses conhecimentos nas mais variadas situações. Para explicar as dimensões do desenvolvimento, Vygotsky desenvolveu o conceito de zona de desenvolvimento proximal (ZDP). Na concepção dele, a ZDP consiste na “região” que representa a distância entre aquilo que o sujeito sabe e aquilo que ele pode vir a aprender.

Para haver aprendizagem, segundo Ausubel (2003), o conhecimento deve fazer sentido para o sujeito; o aprendiz deve correlacionar o novo conhecimento com os conceitos preexistentes em sua estrutura cognitiva. O forte embasamento conceitual por parte do discente poderá garantir um processo de aprendizagem consistente. Visando criar “âncoras” que transformem a aprendizagem mecânica em significativa, Ausubel aponta para o uso de organizadores prévios, uma estratégia para direcionar a estrutura cognitiva para a aprendizagem significativa. Os organizadores são materiais de aprendizagem que preparam o aluno para a aquisição de novos conhecimentos (AUSUBEL, 2003).

Como forma de se definir as diferenças entre o modo de aprendizado de cada indivíduo, há a teoria de Estilos de Aprendizagem (EA). De acordo com Ferreira (2018), “diversos estudos nas áreas de psicologia cognitiva e pedagogia apontam que cada indivíduo possui diferentes maneiras de captar, processar, analisar e organizar informações durante o processo de aprendizado, o que fundamenta o conceito de Estilos de Aprendizagem”. Do mesmo modo, um EA classifica a forma como os alunos recebem e processam as informações em escalas relativas (FELDER; SILVERMAN et al., 1988).

De acordo com o modelo de aprendizagem Felder-Silverman, pode-se classificar os estilos de aprendizagem dos indivíduos da seguinte forma: Ativo ou Reflexivo, se tem preferência por trabalhos em grupo ou individuais; Sensitivo ou Intuitivo, se mais voltado aos fatos concretos ou às teorias; Visual ou Verbal, se prefere representações visuais ou escritas; e Sequencial ou Global, se aprende de modo incremental ou integral (FELDER; SILVERMAN et al., 1988).

2.3 PERSONAS

Durante o desenvolvimento de sistemas, os usuários são uma fonte importante de conhecimento, pois eles podem contribuir de diferentes modos nesse processo, seja atuando diretamente no processo de Engenharia de Software, através de sugestões de modificações, ou indiretamente, por meio de avaliações do sistema (ALMALIKI; NCUBE; ALI, 2015).

Desse modo, tendo em vista que cada pessoa possui uma personalidade e que é necessário um grande número de pessoas envolvidas para a especificação dos sistemas, o custo torna-se elevado. Assim, as Personas surgem como uma forma de identificar as preferências dos usuários, mediante a criação de pessoas e personalidades fictícias (SHAHRI et al., 2016).

Há diversos benefícios advindos da utilização de Personas. Conforme Almaliki, Ncube e Ali (2015), o uso dessa ferramenta torna mais simples o processo de design, fornece uma forma mais eficaz de comunicação entre os profissionais que trabalharão no processo da engenharia do sistema, limita a moldagem das necessidades e desejos dos usuários pela conveniência e minimiza projetos autorreferenciais dos profissionais que trabalharão no processo, ajuda os engenheiros a limitarem-se a um conjunto limitado de personalidades, auxilia na revisão das necessidades para fins de validação e, também, serve como fonte de inspiração durante todo o processo.

Desse modo, ainda segundo eles (ALMALIKI; NCUBE; ALI, 2015), é necessário segmentar os usuários de acordo com os padrões encontrados entre eles, sendo que para cada um dos segmentos, deve ser elaborada uma persona, dando-lhe nome, idade, gênero, imagem e cenários, o que possibilita uma maior facilidade a entender as necessidades dos usuários que o sistema deverá suprimir.

Com a utilização das personas no desenvolvimento do sistema proposto, busca-se criar um sistema que realize a recomendação de materiais de aprendizagem de acordo com os perfis de usuário dos alunos da disciplina de Algoritmos I, do curso de Computação do IFSUL, campus Passo Fundo.

2.4 SISTEMAS DE RECOMENDAÇÃO

Considerando que, de acordo com Ismail e Ebrahimi (2015), as pesquisas apontam a preferência dos estudantes por buscar informações e recursos de aprendizagem através de motores de busca a outros meios de pesquisa, os sistemas de recomendação estão sendo apresentados como solução para os problemas desses motores, pois podem auxiliar os usuários a encontrar o material desejado com maior precisão.

Um sistema de recomendação tenta prever o interesse do usuário por produtos ou serviços, baseando-se em suas preferências. Assim, esse sistema deve utilizar de meios

como as avaliações de conteúdos ou até mesmo a navegação do usuário, para prever itens a serem sugeridos a ele (RICCI; ROKACH; SHAPIRA, 2010). Segundo Reategui e Cazella (2005),

Para que se possa recomendar itens a um usuário, é necessário se ter conhecimento sobre quem é este usuário. É necessário capturar e armazenar seus dados pessoais e comportamentais relativos. Portanto, é essencial que se possa identificar o usuário no momento em que ele acessa o sistema onde foram implantadas as rotinas de recomendação.

No contexto dos sistemas de recomendação, consoante Ferreira (2018),

Técnicas de aprendizado de máquina estão relacionados ao processamento e organização do conhecimento a fim de realizar observações que resultem em novos conhecimentos. A maneira com a qual essa manipulação de experiências é realizada é denominada pelos métodos de inferência lógica.

Nesse sentido, os métodos de inferência lógica serão responsáveis pelo aprendizado que o sistema de recomendação terá com relação aos usuários e suas informações. De acordo com Ferreira (2018), esses métodos podem ser classificados em indução, quando as premissas não necessariamente são verdadeiras, mas que têm como resultado uma hipótese; em dedução, parte de premissas das quais já se há um conhecimento definido, e a partir de cálculos em cima disso, tem-se como consequência um resultado verdadeiro; e em abdução, aplicadas no sentido de analisar informações sobre as quais há um conhecimento incompleto, afim de completá-lo por meio de hipóteses baseadas nas informações já conhecidas.

Assim, “na área educacional os sistemas de recomendação (SR) extraem e utilizam informações sobre o contexto e perfil dos alunos, dentre elas suas preferências, objetivos, estilos de aprendizagem e dados dos dispositivos tecnológicos utilizados” (REZENDE et al., 2018). Conforme Reategui e Cazella (2005), há três técnicas de filtragem de informações que podem ser aplicadas aos SRs: Filtragem Baseada em Conteúdo, a qual se baseia no conteúdo dos itens do sistema e também nas informações do perfil do usuário; Filtragem Colaborativa, não exige análise do conteúdo dos itens, pois a filtragem é baseada na troca de experiências entre usuários; e Filtragem Híbrida, é a técnica em que se une vantagens das duas técnicas de filtragem citadas anteriormente.

No sistema de recomendação desenvolvido, foi utilizada a técnica de filtragem colaborativa, pois a recomendação dar-se-á por meio da identificação de padrões de preferências entre cada usuário com base em suas avaliações de materiais de aprendizagem.

A recomendação por filtragem colaborativa consiste em prever itens para um determinado utente, baseando-se nas avaliações em comum que ele possui com outros usuários (HASSANIEH et al., 2018). Desse modo, é necessário que sejam realizados cálculos de similaridade entre as avaliações efetuadas por eles. Para isso, há algumas fórmulas que

possibilitam a realização dos cálculos de similaridade, a seguir serão descritas algumas delas.

O Coeficiente de Correlação de Pearson (CCP) é uma fórmula que calcula a média de similaridade entre duas variáveis e é medida entre -1 à +1. Assim, +1 indica itens similares e -1 itens não similares (SONDUR; CHIGADANI; NAYAK, 2016). De acordo com Hassanieh *et al.* (2018), “a principal desvantagem desta medida é não retornar um resultado preciso quando os dois usuários possuem uma avaliação em comum, ou quando certo usuário possui somente um item avaliado”.

Já o Cosseno de Vetor de Similaridade (CVS) é usado para estimar a semelhança entre dois vetores (SONDUR; CHIGADANI; NAYAK, 2016). O resultado da similaridade entre os itens dá-se por um valor entre -1 e +1. A desvantagem dessa métrica é que ela não considera as diferenças na média das avaliações realizadas por dois usuários, ou seja, “se um usuário avaliar 5 filmes como 1 de 5 estrelas, e outro usuário avaliar os mesmos 5 filmes como 5 de 5 estrelas, o CVS deles será de 1” (HASSANIEH *et al.*, 2018).

O Coeficiente de Correlação de Spearman (CCS), ao contrário do CCP, considera os ranques das avaliações, ou seja, o ranque 1 indica maior similaridade, enquanto maiores ranques indicam menor correlação (HASSANIEH *et al.*, 2018). Essa medida evita o problema de normalização de avaliações usando ranques, no entanto, “não é bom quando a faixa de avaliações possui apenas alguns valores, uma vez que criaria um grande número de avaliações empatadas, o que é mais caro que o CCP, pois as avaliações precisam ser armazenadas para calcular seu ranque” (HASSANIEH *et al.*, 2018).

No sistema de recomendação desenvolvido, utilizou-se o CCS para os cálculos de similaridade, pois apresentou melhores resultados de precisão com relação aos demais.

Além disso, para possibilitar o ensino personalizado através dessas ferramentas, os EAs são essenciais, de modo a permitirem a modelagem do perfil de cada aluno (OLIVEIRA *et al.*, 2018).

O presente trabalho objetiva desenvolver um sistema de recomendação de materiais de aprendizagem que auxilie alunos dos cursos de Computação nas disciplinas de Algoritmos. Esses sistemas usam técnicas de *Machine Learning* (ML) (Aprendizado de Máquina em Português) para prever itens que possam ser do interesse do usuário.

Para que se possa persistir as informações dos alunos e os dados de preferências deles durante a utilização desse sistema, será necessária a utilização do conceito de Big Data, pois o volume e o formato divergente dos dados armazenados podem dificultar ou até mesmo impossibilitar o processamento por sistemas relacionais tradicionais (IBM; ZIKOPOULOS; EATON, 2011). Da mesma forma, os bancos de dados NoSQL surgem como alternativa aos convencionais, pois mostram “uma abordagem diferente de persistência de dados, baseada em disponibilidade, desempenho e escalabilidade dos dados” (TOTH, 2011).

Além disso, de modo a possibilitar o desenvolvimento rápido da GUI, e possibilitar que essa seja acessada pelos mais variados dispositivos que tenham acesso à Internet, usar-se-á a tecnologia de Progressive Web Apps (PWA). As PWAs também oferecem alguns benefícios como responsividade, são interativas como um aplicativo, estão sempre atualizadas com a última versão disponível, podem ser encontradas através de mecanismos de busca, podem ser instaladas no dispositivo e ainda podem ser acessadas através de um endereço na internet (HUME, 2017).

Ainda, para que se permita a comunicação entre a GUI e o sistema provedor de recomendações, será necessário a utilização de Serviços WEB. Esses serviços permitem a interação entre duas máquinas em uma rede, conforme a W3C (MORAES et al., 2013). No caso do sistema proposto, será utilizado o estilo arquitetural Representation State Transfer (REST), o qual se trata de um conjunto de normas e padrões a serem seguidos na composição de um Serviço WEB (RIBEIRO; FRANCISCO, 2016).

2.5 AVALIAÇÃO DE SISTEMAS DE RECOMENDAÇÃO

As avaliações dos sistemas de recomendação são importantes pois possibilitam saber a qualidade das técnicas, dos métodos e algoritmos de recomendação, facilitando a comparação de quais retornam melhores resultados, baseado em diferentes soluções (BOBADILLA et al., 2013).

A avaliação dos sistemas de recomendação deve ser realizada durante várias fases, desde o tempo de projeto, com a execução de vários algoritmos de recomendação utilizando a mesma base de dados, comparando sua performance, até após seu lançamento, em que se faz necessário a avaliação com usuários reais, através da análise de logs a fim de melhorar a performance (RICCI; ROKACH; SHAPIRA, 2010).

De acordo com Bobadilla *et al.* (2013), as métricas de qualidade podem ser classificadas da seguinte forma: métricas de predição, como Mean Absolute Error (MAE), Root of Mean Square Error (RMSE), Normalized Mean Average Error (NMAE); métricas de conjunto de recomendação, como Precisão, Recall e F1; métricas de nível de recomendação, como half-life e discounted cumulative gain; e métricas de diversidade, como a diversidade e a inovação dos itens recomendados.

Ainda de acordo com eles (BOBADILLA et al., 2013), “a confiança dos usuários em um determinado SR não depende diretamente da acurácia dos conjuntos de possíveis predições. Um usuário ganha confiança no SR quando este usuário concorda com um reduzido conjunto de recomendações feitos pelo SR”.

Assim, as métricas de qualidade de recomendação mais utilizadas são: Precisão, que indica a proporção de itens relevantes com relação aos itens recomendados; Recall, que

indica a proporção de verdadeiros-positivos com relação aos itens recomendados relevantes; e F1, que se trata da combinação da Precisão e do Recall.

2.6 TRABALHOS CORRELATOS

Existem diversas pesquisas envolvendo o uso de sistemas de recomendação. Ferro *et al.* (2011) desenvolveram um sistema no ambiente virtual Moodle, com as funcionalidades de Perfil do Usuário, em que os usuários alunos deveriam indicar as áreas de sua preferência; Perfil do Material Didático, em que os usuários professores ou administradores deveriam indicar as áreas que compõem o Perfil do Material Didático ao submeter um material didático; Biblioteca, onde os alunos ao entrarem e visualizarem algum conteúdo, teriam a área do material didático adicionada ao seu perfil, caso ainda não fizessem parte dele, ou passariam a ter maior importância, caso já fizessem parte dele; Módulo visualizador de recomendações, em que as recomendações seriam exibidas no início da tela, de modo que não fosse necessário muitos cliques para que os usuários conseguissem visualizá-la; e ainda a Avaliação de Recomendação, em que o usuário, ao clicar sobre a recomendação, poderá avaliá-la, e após a avaliação poderá fazer o download do material didático.

Além disso, Vieira e Nunes (2012) desenvolveram o DICA, que se trata de um sistema de recomendação de objetos de aprendizagem baseado em conteúdo. Inicialmente, através do uso da técnica de Filtragem Baseada em Conteúdo, ele identifica a proximidade entre os temas dos Objetos de Aprendizagem, selecionando 20 itens com temas mais próximos. Após isso, ele verifica a quantidade de vezes que um termo ocorre no documento e, em seguida, identifica a relevância de um termo em um conjunto de documentos e, dessa forma, é realizada a comparação para apresentar quais Objetos possuem o maior número de termos em comum, dentre os 20 selecionados anteriormente.

Ainda, Reis e Barrére (2014), propuseram uma arquitetura em que conteúdos previamente categorizados com relação ao nível de conhecimento, pelo professor, sejam recomendados aos alunos, possibilitando a eles, inclusive, indicar conteúdos. Conforme o sistema for utilizado, informações de perfil, contexto e qualidade de experiência serão utilizados para que seja realizada a composição de uma recomendação adaptada. Para validação dessa arquitetura, eles desenvolveram um protótipo denominado *Context Aware Learning*.

Cabe ressaltar que o sistema proposto difere-se dos demais por ser baseado em personas, ou seja, será desenvolvido de modo personalizado aos perfis de usuário dos alunos do IFSUL, câmpus Passo Fundo. Também levará em consideração os estilos de aprendizagem, identificando padrões entre grupos de usuários, com base nas avaliações dos materiais realizadas por eles.

3 IMPLEMENTAÇÃO DO SISTEMA

Neste capítulo, serão detalhadas as tecnologias utilizadas no desenvolvimento do sistema, sua arquitetura e seu funcionamento.

3.1 TECNOLOGIAS UTILIZADAS

Nesta seção serão descritas as tecnologias necessárias para o desenvolvimento do sistema proposto.

3.1.1 Ionic Framework

É um conjunto de ferramentas que possibilitam o desenvolvimento fácil e rápido de aplicações Híbridas. Compõe um ecossistema em que estão inseridos o Framework Angular e o Framework ReactJS, baseados em JavaScript, e o Apache Cordova, responsável pela construção de aplicações nativas (WILKEN; BRADLEY, 2016).

O Ionic Framework fornece grande número de componentes desenvolvidos através das tecnologias HTML, CSS e JavaScript, para o desenvolvimento de uma poderosa interface de usuário. Além disso, também conta com um sistema de linha de comando que auxilia o desenvolvedor em tarefas como criação de projeto e construção do software (WILKEN; BRADLEY, 2016).

O Ionic será utilizado para desenvolver a interface responsiva do sistema, oferecendo possibilidade de acesso, por se tratar de uma PWA, em vários dispositivos.

3.1.2 MongoDB

O MongoDB é um Sistema de Gerenciamento de Banco de Dados (SGBD) cujo modelo e estratégias de persistências têm enfoque na alta taxa de transferência de dados, seja para leitura como gravação (BANKER, 2011). Além disso, ele também possui a capacidade de se expandir através de recursos como índices secundários, consultas de intervalo, classificação, agregações e índices geoespaciais (CHODOROW, 2013).

O modelo desse banco de dados é orientado a documentos, e esses documentos são um conjunto de propriedades com seus respectivos valores, e esses podem ser simples, sequências de caracteres, números, datas, matrizes ou até mesmo outros documentos (BANKER, 2011).

Desse modo, a abordagem orientada a documentos desses bancos permite que sejam representadas estruturas hierárquicas complexas em um único documento, e isso se encaixa

no modelo de programação orientada a objetos, facilitando, assim, o desenvolvimento das aplicações (CHODOROW, 2013).

Além disso, a medida que o volume de dados armazenados para uma determinada aplicação cresce, surgem as dificuldades de como escalonar as bases de dados. Com a orientação a documentos, o MongoDB possui maior facilidade de dividir os dados entre vários servidores de maneira automática, fazendo com que os desenvolvedores não necessitem se preocupar com o dimensionamento da aplicação (CHODOROW, 2013).

O MongoDB será utilizado como SGBD, porque os dados das avaliações dos usuários tendem a possuir formatos bastante divergentes uns dos outros, o que tornaria o processamento lento se utilizado banco de dados convencional.

3.1.3 Spring Boot

O Spring Boot surgiu como uma grande inovação do ecossistema Spring, pois facilita o desenvolvimento de aplicações Java através de recursos como configurações automáticas das funcionalidades e gerenciamento simplificado de dependências (WALLS, 2016).

O Framework Spring Boot será utilizado no desenvolvimento, para que permita, por meio das configurações automáticas de funcionalidades e gerenciamento, a simplificação da criação da interface de API REST do sistema.

3.1.4 Apache Mahout

O Apache Mahout é um framework para *Machine Learning* escrito na linguagem de programação Java. Esse framework tem enfoque em três áreas de *Machine Learning*, são elas mecanismo de recomendação, *clustering* e classificação (ANIL et al., 2010).

Em alguns cenários a quantidade de dados a serem processados é tão grande a ponto de tornar impossível o processamento em uma única máquina. Através da utilização do Apache Hadoop, o Mahout possibilita a escalabilidade do processamento a um conjunto de máquinas (ANIL et al., 2010).

3.1.5 YouTube API

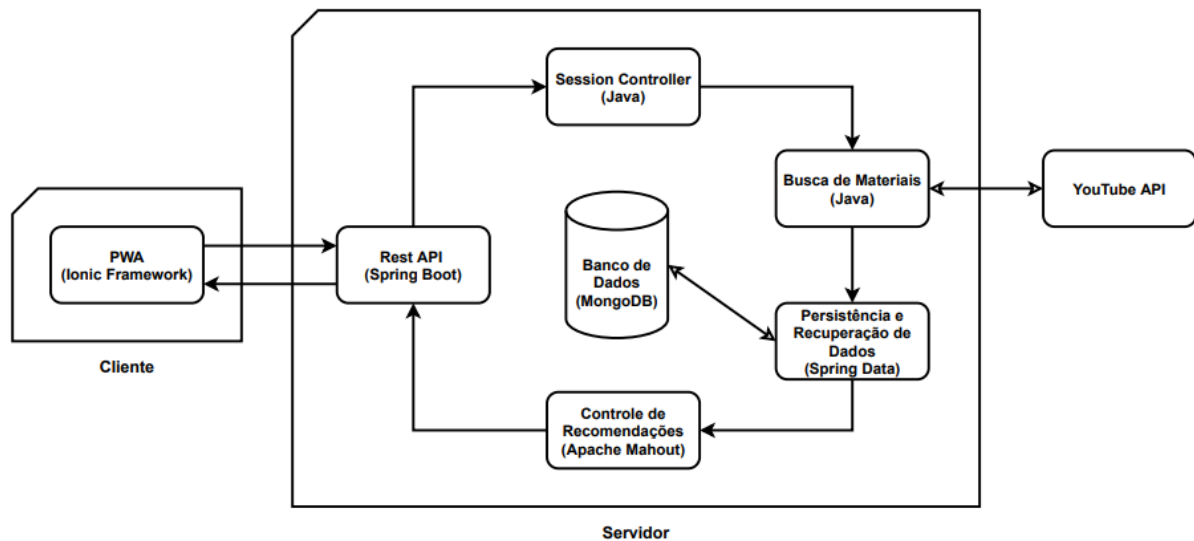
A YouTube API é uma ferramenta que permite tanto pesquisa por vídeos através da informação de diversos parâmetros suportados por ela, que permitem a filtragem por esses critérios (Qin; Menezes; Silaghi, 2010).

Essa ferramenta será utilizada no sistema proposto para a realização de pesquisas por materiais de aprendizagem disponíveis na plataforma de compartilhamento de vídeos YouTube, os quais serão classificados e exibidos ao usuário.

3.2 ARQUITETURA DO SISTEMA

O sistema é composto por uma arquitetura Cliente-Servidor (Figura 1), na qual o Cliente (*front-end*), é responsável por realizar as requisições de dados ao Servidor (*back-end*). O servidor, por sua vez, processa a requisição e retorna uma resposta ao Cliente com os dados solicitados. Isso permite a comunicação entre vários Clientes com o Servidor ao mesmo tempo e o acesso a informações compartilhadas.

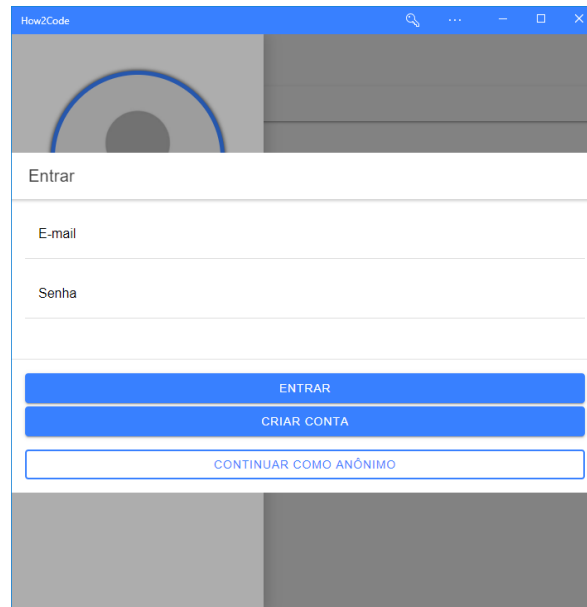
Figura 1 – Arquitetura e Funcionamento



Fonte: Própria, 2020

Desse modo, o Cliente é composto por uma PWA, que foi desenvolvida com a utilização do Ionic Framework. Quando o usuário abrir esta aplicação, será exibida a Tela de Login conforme a Figura 2, na qual haverá as opções de fazer Login, em que o usuário poderá entrar na sua conta já criada; criar uma nova conta, em que o usuário poderá criar uma nova conta, caso ainda não tenha; e continuar como usuário anônimo, que permitirá o acesso a aplicação sem necessidade de Login.

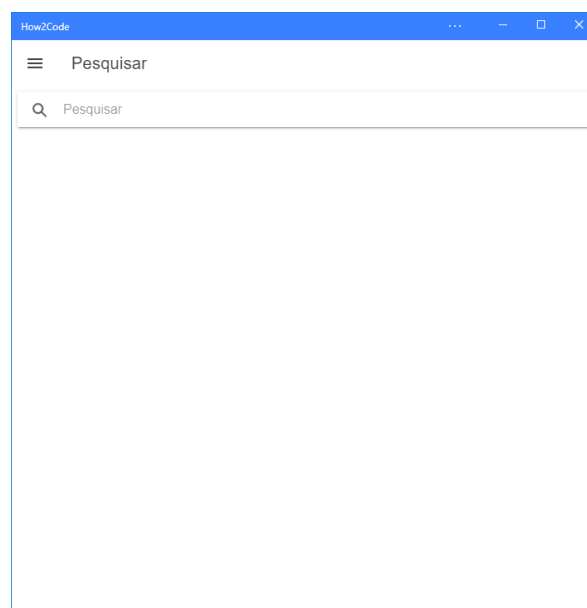
Figura 2 – Tela de Login



Fonte: Própria, 2020

Caso o usuário da aplicação opte por realizar o Login ou continuar como anônimo, ele será direcionado para a página inicial. Como se pode visualizar na Figura 3, a tela inicial é composta por um cabeçalho, no qual estão o botão para expandir o menu lateral e o título da página, e por um corpo, no qual está localizado o campo para realizar a busca por materiais de aprendizagem.

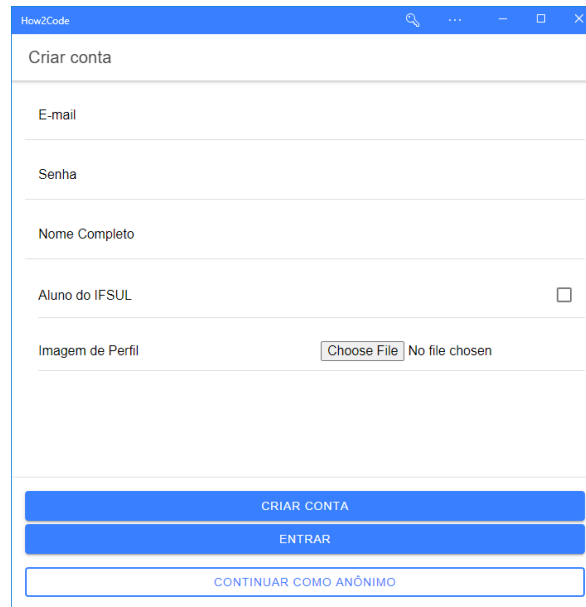
Figura 3 – Página Inicial



Fonte: Própria, 2020

Por outro lado, se o usuário optar por Criar conta, na tela de login, ele será direcionado à tela de criação de conta. Conforme a Figura 4, essa tela é composta pelo cabeçalho, no qual está o título da página, um formulário em que o usuário deve informar alguns dados de identificação, e também pode informar se é um aluno do IFSUL e definir uma imagem de perfil. Além disso, abaixo há as opções de Criar conta, que deve ser selecionada após o preenchimento dos dados no formulário, a opção Entrar, que redireciona o usuário para o formulário de login e a opção de continuar como anônimo que o redireciona para a página inicial.

Figura 4 – Tela de criação de conta



How2Code

Criar conta

E-mail

Senha

Nome Completo

Aluno do IFSUL

Imagem de Perfil No file chosen

CRIAR CONTA

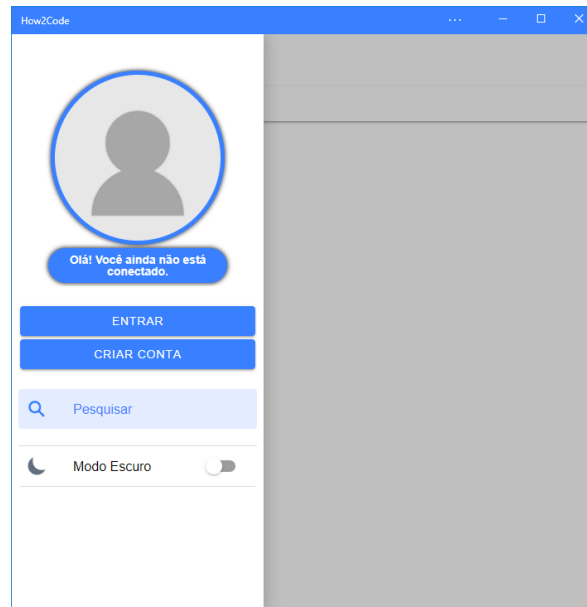
ENTRAR

CONTINUAR COMO ANÔNIMO

Fonte: Própria, 2020

Na página inicial, ao expandir o menu lateral, navegando como usuário anônimo, como na Figura 5, é possível identificar a exibição de uma imagem de usuário não informada, seguida de uma mensagem de que o usuário não está conectado. Abaixo, ainda há as opções de abrir tela de login e criação de conta, a lista de páginas, composta pelo item de menu Pesquisar, que redireciona para a página inicial, e uma opção para o usuário decidir se deseja utilizar o modo claro ou escuro da aplicação.

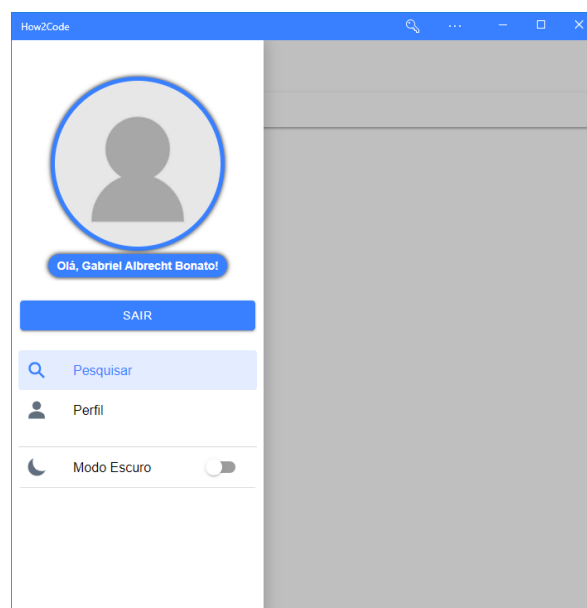
Figura 5 – Menu Lateral para usuário anônimo



Fonte: Própria, 2020

Por outro lado, estando conectado, no menu lateral conforme Figura 6, é possível visualizar a imagem de perfil do usuário, se ela tiver sido especificada, seguida de uma mensagem de saudação que contém o nome do usuário conectado. Também é possível verificar a presença de um botão para sair da conta. Ademais, além das opções também presentes para usuários anônimos (Pesquisar e definição de Modo Escuro), há o item de menu Perfil, que possibilita a visualização dos dados pessoais do usuário conectado.

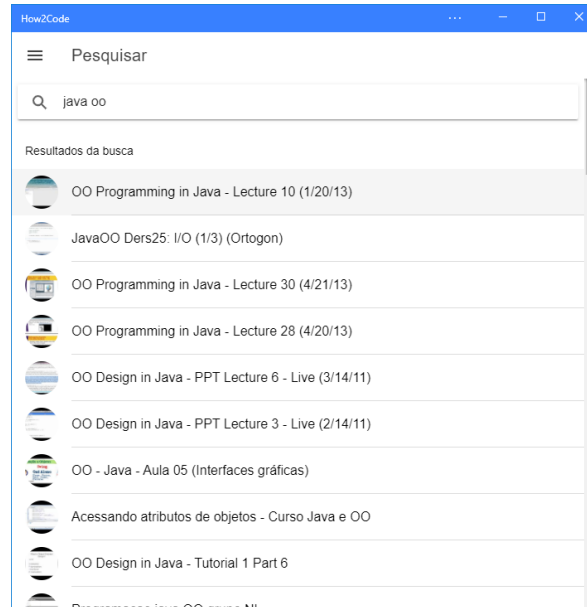
Figura 6 – Menu Lateral para usuário logado



Fonte: Própria, 2020

Ainda na página inicial, ao realizar uma busca com usuário não autenticado ou se a busca não retornar itens recomendados, são listados os itens logo abaixo do campo de busca, de acordo com a Figura 7.

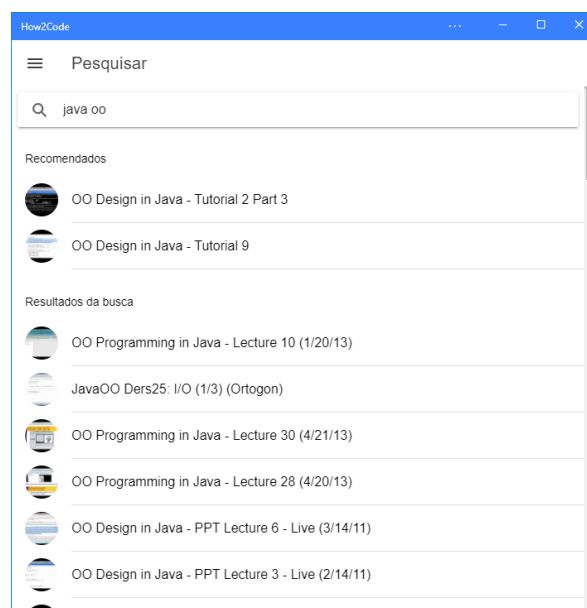
Figura 7 – Lista de Materiais de Aprendizagem sem recomendação



Fonte: Própria, 2020

Se a busca for realizada por um usuário autenticado e tiver itens a serem recomendados, a exibição dar-se-á conforme a Figura 8, contando com uma lista de recomendados e outra para os resultados da busca.

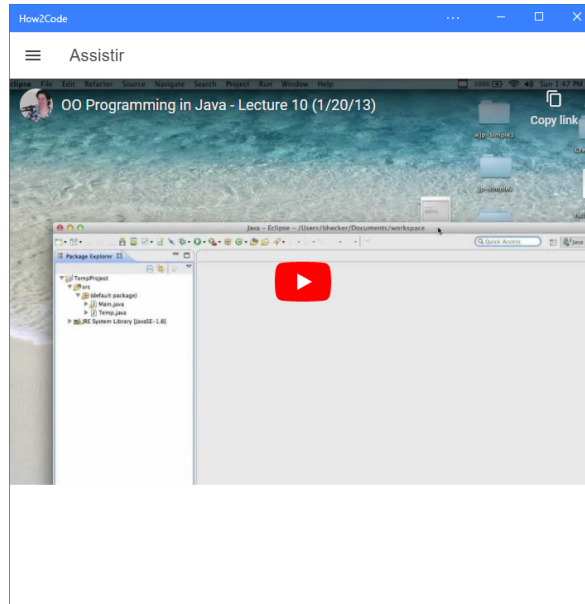
Figura 8 – Lista de Materiais de Aprendizagem com recomendação



Fonte: Própria, 2020

Ao optar por visualizar algum dos materiais de aprendizagem, o usuário anônimo visualizará uma tela como a da Figura 9, na qual constará o material de aprendizagem a ser visualizado.

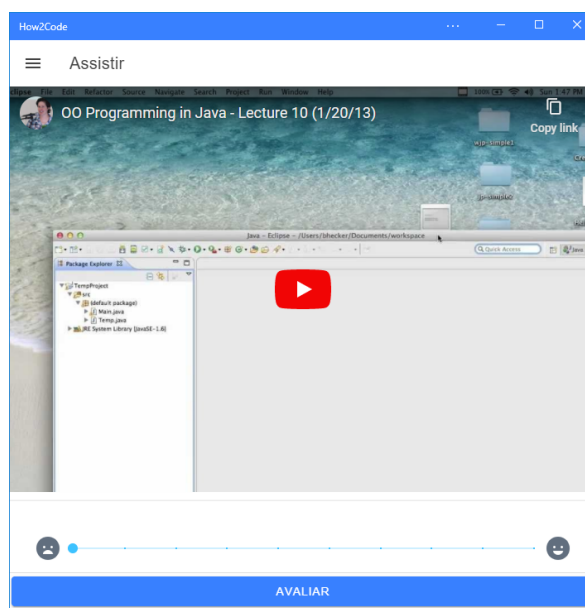
Figura 9 – Visualização de Materiais de Aprendizagem para usuário anônimo



Fonte: Própria, 2020

Por outro lado, se estiver autenticado, além do material de aprendizagem, no rodapé constará um formulário em que o usuário poderá avaliar o material com uma nota de 1 a 10, indicando o quanto ele gostou desse item, de acordo com a Figura 10.

Figura 10 – Visualização de Materiais de Aprendizagem para usuário logado



Fonte: Própria, 2020

Como o sistema será composto por uma arquitetura Cliente-Servidor, será utilizado o modelo REST para o Serviço WEB, que utiliza dos métodos HTTP para transmissão de pacotes de dados. Assim, será implementado no Servidor uma interface REST que receberá as requisições do Cliente.

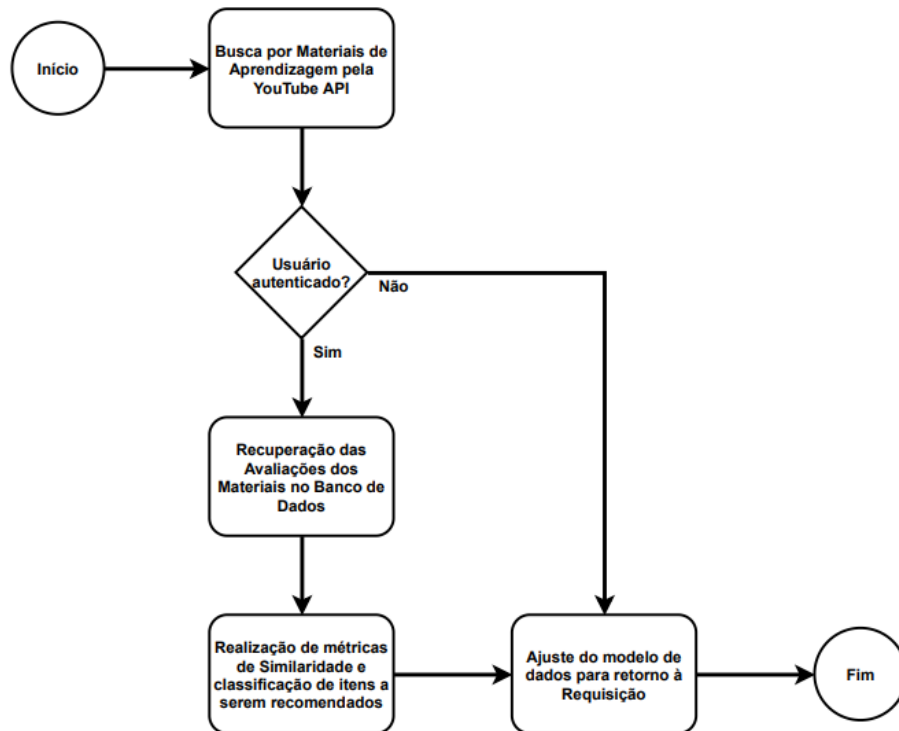
Quando o usuário pesquisar por Materiais de Aprendizagem no Cliente, será realizada uma requisição ao Servidor, na qual serão enviados os termos da busca. Ao receber essa requisição, ocorrerá um controle de sessão, para que o sistema consiga verificar se o usuário está logado ou não. Essa informação será utilizada durante a recomendação dos materiais de aprendizagem.

Em seguida, o sistema realizará uma busca de Materiais de Aprendizagem através da tecnologia YouTube API com os termos enviados pelo usuário. Então, se o usuário estiver autenticado, será realizada uma busca na base dados NoSQL, na qual será utilizado o MongoDB como SGBD, por todas as avaliações realizadas por todos os usuários.

Após isso, as avaliações dos materiais de aprendizagem serão encaminhadas a um Controle de Recomendações, que com o auxílio do framework Apache Mahout, será responsável por calcular a similaridade entre perfis de usuário com base nas avaliações semelhantes entre eles, ou seja, identificando a similaridade entre o estilo de aprendizagem do usuário atual com os outros usuários. O cálculo de similaridade utilizado no presente sistema é o coeficiente de correlação de Spearman. Após elaborar o conjunto de dados com os índices de similaridade, serão utilizados os itens avaliados por 25% do número total de usuários cadastrados em banco, para então ser realizada a classificação dos materiais e executada a recomendação.

Conforme a Figura 11, pode-se visualizar os fluxos do Controle de Recomendações. Caso seja um usuário anônimo, apenas será retornado os itens do resultado da requisição à YouTube API.

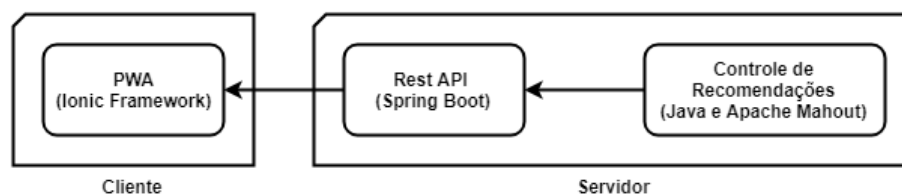
Figura 11 – Diagrama de Fluxos do Controle de Recomendações



Fonte: Própria, 2020

Por fim, de acordo com a Figura 12, os resultados serão retornados à interface de API REST, que será responsável por enviar uma resposta ao Cliente, o qual irá exibir a listagem de materiais de aprendizagem ao usuário.

Figura 12 – Arquitetura do Sistema para Resposta à Requisição



Fonte: Própria, 2020

No banco de dados, serão armazenadas as informações de perfil de cada usuário cadastrado e também as suas avaliações de materiais de aprendizagem.

O conceito de personas foi utilizado no sistema de modo a simplificar a engenharia de requisitos e possibilitar que as recomendações sejam personalizadas para vários segmentos de usuários, conforme a Figura 13, pode-se visualizar as personas nas quais o sistema foi baseado, sendo algumas delas matriculadas no curso de Ciência da Computação, e também abrangendo combinações de estilos de aprendizagem.

Figura 13 – Personas Utilizadas para o Desenvolvimento do Sistema



Fonte: Própria, 2020

Com a utilização das personas, foi possível desenvolver o sistema de recomendação personalizado para os alunos do curso de Ciência da Computação do IFSUL, levando em consideração os estilos de aprendizagem de cada um.

De modo a tentar diferenciar os perfis de aprendizagem de cada aluno, foram utilizados os estilos de aprendizagem definidos na escala Felder-Silverman. Sendo eles Ativo ou Reflexivo, Intuitivo ou Sensitivo, Verbal ou Visual e Global ou Sequencial.

Além disso, também foram considerados os perfis de alunos do curso de Ciência da Computação do IFSUL e profissionais que atuam na análise e desenvolvimento de sistemas.

4 TESTES E VALIDAÇÕES

Com o intuito de avaliar a qualidade de itens recomendados pelo sistema antes da sua liberação para acesso a usuários reais, foi elaborada uma massa de dados simulada contendo 10 mil usuários, 1,1 milhão de avaliações e 400 materiais de aprendizagem diferentes, de modo a possibilitar a execução das métricas de avaliação.

Para a elaboração desse conjunto de dados, foram definidos 3 níveis de avaliação, sendo o nível 1 avaliações com valores menores que 4, o nível 2 maiores ou iguais a 4 e menores que 8, e o nível 3 avaliações maiores ou iguais a 8. Além disso, os materiais de aprendizagem foram divididos em 4 grupos, com a finalidade de simular diferentes tipos de materiais.

A partir disso, para que fosse possível a combinação dos 3 níveis de avaliação com os 4 grupos de materiais de aprendizagem, fez-se necessário a divisão dos usuários em 81 grupos, simulando os diferentes perfis e estilos de aprendizagem, sendo que cada grupo possuía um valor aproximado de 123 usuários.

Baseando-se nessas informações, a massa de dados foi elaborada por meio da criação de 10 mil usuário fictícios, da coleta de 400 materiais de aprendizagem diferentes do YouTube e com esses usuários e materiais de aprendizagem foram geradas 1,1 milhão de avaliações.

Para gerar as avaliações seguiu-se um processo em que se gerou 300 mil avaliações de materiais com base nos grupos de usuários e materiais de aprendizagem previamente definidos. A partir disso, foram geradas mais cerca de 800 mil avaliações de forma randômica, ou seja, sem definição de valores para os grupos de usuários e materiais de aprendizagem.

Observou-se durante a realização das avaliações do sistema de recomendação, que à medida que aumentava a proporção de avaliações para usuários e materiais de aprendizagem, apresentava-se a melhora na precisão e na recuperação dos itens recomendados.

Desse modo, com as 300 mil avaliações geradas com base nos grupos de usuário e materiais citados anteriormente, obteve-se aproximadamente 10% de precisão. Ao expandir a massa de dados para 600 mil avaliações, sendo que dessas, 300 mil foram geradas de forma randômica, a precisão chegou a aproximadamente 40%. Observando o aumento da precisão após o aumento das avaliações realizadas, gerou-se mais cerca de 500 mil avaliações randômicas e, assim, atingiu-se a precisão de 83%, recall de 92% e F1 de 87%. Ou seja, a cada 100 itens recomendados, 83 deles são relevantes ao usuário e, além disso, a cada 100 itens recomendados relevantes ao usuário, 92 são verdadeiros-positivos.

Além disso, foram realizadas avaliações utilizando os cálculos de similaridade

Coefficiente de Correlação de Pearson, Cosseno de Vetor de Similaridade e Coeficiente de Correlação de Spearman, no entanto, este último apresentou melhor qualidade na execução das métricas de avaliação, apesar de apresentar performance inferior aos demais.

Para que fosse possível avaliar o sistema de recomendação quanto a personalização para os alunos do curso de Ciência da Computação do IFSUL, o sistema foi disponibilizado de forma online, e o link de acesso colocado no ambiente Moodle da disciplina de Algoritmos I, durante o segundo semestre de 2019.

Os alunos foram convidados a utilizar a plataforma como apoio a seus estudos e realização de tarefas propostas em aula. Ao fim do período de utilização do sistema pelos alunos, obteve-se o registro total de 11 usuários cadastrados e 34 avaliações de materiais de aprendizagem.

Devido à baixa volumetria de dados para a realização da avaliação dos algoritmos de recomendação de forma personalizada aos alunos do IFSUL, não foi possível obter resultados. Desse modo, essa avaliação será realizada futuramente a medida que a base de dados for sendo incrementada.

5 CONSIDERAÇÕES FINAIS

O presente trabalho apresenta-se como um meio a contribuir para o desenvolvimento da aprendizagem dos alunos, fornecendo-lhes recomendações de materiais de aprendizagem que possam ser relevantes aos seus estudos, levando em consideração suas preferências e estilos de aprendizagem.

Ademais, o sistema também conta com uma grande quantia de materiais de aprendizagem, tendo em vista que os itens são advindos do YouTube. Além de contar com uma pré-classificação, do YouTube, dos itens mais relevantes à pesquisa realizada, também são realizados cálculos de similaridade, a fim de encontrar itens relevantes mais específicos ao usuário.

Percebe-se que os objetivos do trabalho proposto foram alcançados parcialmente. O sistema desenvolvido está realizando as recomendações adequadamente, atingindo bons índices de precisão, dependendo da proporção entre o número de avaliações e o número de usuários e materiais de avaliação. No entanto, não foi possível realizar a avaliação do sistema quanto ao estilo de aprendizagem dos alunos do curso de Computação do IFSUL, campus Passo Fundo, devido à baixa quantidade de avaliação de materiais de aprendizagem realizadas por eles.

Desse modo, em trabalhos futuros, será feita a disponibilização do sistema para um grupo maior de usuários, para que assim, gerando uma massa de dados maior, seja possível realizar a avaliação dos conjuntos de itens de recomendação utilizando dados reais e verificar outros Algoritmos que possam se mostrar mais efetivos à recomendação.

Além disso, para o presente trabalho, as recomendações estão sendo realizadas em tempo real, no momento em que o usuário busca por materiais de aprendizagem. Isso, em um curto prazo, devido ao baixo volume de dados não traz grandes impactos à usabilidade do sistema, no entanto, à medida que for aumentando o número de usuários, o número de avaliações e o número de materiais de aprendizagem avaliados, haverá impacto de performance.

Assim, será necessário, em projetos futuros, o desenvolvimento de uma interface que realize de forma periódica a indexação de itens a serem recomendados para cada usuário, para que os resultados das buscas tornem-se mais rápidos e seja necessário o uso de menos recursos de processamento.

REFERÊNCIAS

- ALMALIKI, M.; NCUBE, C.; ALI, R. Adaptive software-based feedback acquisition: A persona-based design. In: *2015 IEEE 9th International Conference on Research Challenges in Information Science (RCIS)*. [S.l.: s.n.], 2015. p. 100–111. ISSN 2151-1349. Citado na página 13.
- ALVES, F. P.; JAQUES, P. Um ambiente virtual com feedback personalizado para apoio a disciplinas de programação. In: *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. [S.l.: s.n.], 2014. v. 3, n. 1, p. 51. Citado na página 11.
- ANIL, R. et al. *Mahout in Action*. Manning Publications Co. Sound View Ct. 3B Greenwich, CT 06830: [s.n.], 2010. 350 p. Disponível em: <<http://manning.com/owen/>>. Citado na página 19.
- AUSUBEL, D. *Aquisição e retenção de conhecimentos: uma perspectiva cognitiva*. Lisboa: Portugal: Platanos, 2003. Citado na página 12.
- BANKER, K. *MongoDB in Action*. Greenwich, CT, USA: Manning Publications Co., 2011. ISBN 1935182870, 9781935182870. Citado na página 18.
- BARDAGI, M. P.; HUTZ, C. S. "Não havia outra saída": percepções de alunos evadidos sobre o abandono do curso superior. *Psico-USF*, scielo, v. 14, p. 95 – 105, abr. 2009. ISSN 1413-8271. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-82712009000100010&nrm=iso>. Citado na página 8.
- BOBADILLA, J. et al. Recommender systems survey. *Knowledge-based systems*, Elsevier, v. 46, p. 109–132, 2013. Citado na página 16.
- CHODOROW, K. *MongoDB: the definitive guide: powerful and scalable data storage*. [S.l.]: "O'Reilly Media, Inc.", 2013. Citado 2 vezes nas páginas 18 e 19.
- COSTA, E.; AGUIAR, J.; MAGALHÃES, J. Sistemas de recomendação de recursos educacionais: Conceitos, técnicas e aplicações. In: _____. [S.l.: s.n.], 2013. p. 57–78. ISBN 9788576692836. Citado na página 11.
- FELDER, R. M.; SILVERMAN, L. K. et al. Learning and teaching styles in engineering education. *Engineering education*, v. 78, n. 7, p. 674–681, 1988. Citado na página 12.
- FERREIRA, L. D. *Técnicas de aprendizado de máquina aplicadas à classificação de estudantes a partir de estilos de aprendizagem*. Dissertação (Mestrado), 2018. Disponível em: <<https://doi.org/10.11606/d.55.2018.tde-24102018-151910>>. Citado 2 vezes nas páginas 12 e 14.
- FERRO, M. R. da C. et al. Um Modelo de Sistema de Recomendação de Materiais Didáticos para Ambientes Virtuais de Aprendizagem. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2011. v. 1, n. 1. Citado na página 17.

FILHO, R. L. L. e. S. et al. A evasão no ensino superior brasileiro. *Cadernos de Pesquisa*, scielo, v. 37, p. 641 – 659, dez. 2007. ISSN 0100-1574. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-15742007000300007&nrm=iso>. Citado na página 8.

GOMES, R. C. Um método proposto para o ensino de algoritmos no contexto da educação a distância. *São Paulo*, 2012. Citado na página 11.

HASSANIEH, L. et al. Similarity measures for collaborative filtering recommender systems. In: . [S.l.: s.n.], 2018. p. 1–5. Citado 2 vezes nas páginas 14 e 15.

HUME, D. A. *Progressive Web Apps*. Manning Publications, 2017. ISBN 1617294586. Disponível em: <<https://www.amazon.com/Progressive-Apps-Dean-Alan-Hume/dp/1617294586?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1617294586>>. Citado na página 16.

IBM; ZIKOPOULOS, P.; EATON, C. *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data*. 1st. ed. [S.l.]: McGraw-Hill Osborne Media, 2011. ISBN 0071790535, 9780071790536. Citado na página 15.

ISMAIL, M. A.; EBRAHIMI, A. Recommender systems for research materials: An overview. In: . [S.l.: s.n.], 2015. Citado na página 13.

MORAES, J. et al. Web Services. *Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Rio Grande do Sul, Brasil*, 2013. Citado na página 16.

MORÃES, M. J. F.; POMBEIRO, O. J. Evasão nos cursos de graduação em computação de curitiba. *Anais do EVINCI-UniBrasil*, v. 1, n. 4, p. 2088–2103, 2016. Citado na página 11.

OLIVEIRA, H. L. et al. Detecção automática e dinâmica de estilos de aprendizagem para sistemas adaptativos inteligentes para a educação utilizando a meta-heurística vitis vinífera. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2018. v. 29, n. 1, p. 1323. Citado na página 15.

Qin, S.; Menezes, R.; Silaghi, M. A recommender system for youtube based on its network of reviewers. In: *2010 IEEE Second International Conference on Social Computing*. [S.l.: s.n.], 2010. p. 323–328. Citado na página 19.

REATEGUI, E. B.; CAZELLA, S. C. Sistemas de recomendação. In: . [S.l.: s.n.], 2005. Citado na página 14.

REIS, G.; BARRÉRE, E. Recomendação colaborativa de conteúdos educacionais para dispositivos portáteis. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2014. v. 25, n. 1, p. 934. Citado na página 17.

REZENDE, P. A. et al. Recomendação baseada no perfil e contexto tecnológico do aluno. In: *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*. [S.l.: s.n.], 2018. v. 29, n. 1, p. 1273. Citado na página 14.

RIBEIRO, M. F.; FRANCISCO, R. E. Web Services REST Conceitos, análise e implementação. *2525-3859*, 10 2016. Citado na página 16.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Recommender systems handbook. In: _____. [S.l.: s.n.], 2010. v. 1-35, p. 1–35. Citado 2 vezes nas páginas 14 e 16.

RODRIGUES, F. S. Estudo sobre a evasão no curso de Ciência da Computação da UFRGS. 2013. Citado na página 11.

SHAHRI, A. et al. Engineering software-based motivation: A persona-based approach. In: *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*. [S.l.: s.n.], 2016. p. 1–12. ISSN 2151-1357. Citado na página 13.

SONDUR, M. S. D.; CHIGADANI, M. A. P.; NAYAK, S. Similarity measures for recommender systems: a comparative study. *Journal for Research*, v. 2, n. 3, 2016. Citado na página 15.

TOTH, R. M. Abordagem NoSQL—uma real alternativa. *Sorocaba, São Paulo, Brasil: Abril*, v. 13, 2011. Citado na página 15.

VIEIRA, F. J. R.; NUNES, M. A. S. N. Dica: Sistema de recomendação de objetos de aprendizagem baseado em conteúdo. *Scientia Plena*, v. 8, n. 5, 2012. Citado na página 17.

VYGOTSKY, L. S. *A formação social da mente: o desenvolvimento dos processos psicológicos superiores*. 6rd. ed. São Paulo: Martins Fontes, 1998. Citado na página 12.

WALLS, C. *Spring Boot in action*. [S.l.]: Manning Publications, 2016. Citado na página 19.

WILKEN, J.; BRADLEY, A. *Ionic in action: Hybrid mobile apps with Ionic and AngularJS*. [S.l.]: Manning Publications, 2016. Citado na página 18.

APÊNDICE A – Busca de Materiais pela YouTube API

```
1 public static Map<String, Object> getVideos(String query,
2     String pageToken) {
3
4     try {
5         YouTube youtube = new YouTube.Builder(new NetHttpTransport(),
6             new JacksonFactory(), new HttpRequestInitializer() {
7                 public void initialize(HttpRequest request)
8                     throws IOException {
9                     }
10            }).setApplicationName("RecommenderSystem").build();
11
12        YouTube.Search.List search = youtube.search().list("id, snippet");
13
14        search.setKey(API_KEY);
15        search.setQ(query);
16
17        if (pageToken != null) {
18            search.setPageToken(pageToken);
19        }
20
21        search.setMaxResults((long) 100);
22        search.setType("video");
23        search.setOrder("rating");
24        search.setPart("snippet");
25        search.setFields(
26            "nextPageToken, items(id/videoId, snippet/title, snippet/
27                thumbnails/default/url)");
28
29        SearchListResponse searchResponse = search.execute();
30        List<SearchResult> searchResultList = searchResponse.getItems();
31        List<LearningMaterial> learningMaterials = new ArrayList<>();
32
33        for (SearchResult searchResult : searchResultList) {
34            LearningMaterial learningMaterial = new LearningMaterial();
35
36            learningMaterial.setTitle(searchResult.getSnippet()
37                .getTitle());
38
39            learningMaterial.setVideoId(searchResult.getId()
40                .getVideoId());
```

```
40         learningMaterial.setThumbnail(searchResult.getSnippet()  
41             .getThumbnails().getDefault().getUrl());  
42  
43  
44         learningMaterials.add(learningMaterial);  
45     }  
46  
47     Map<String, Object> response = new HashMap<String, Object>();  
48  
49     response.put("learningMaterials", learningMaterials);  
50     response.put("nextPageToken", searchResponse.getNextPageToken());  
51  
52     LogUtil.logDebug(  
53         "Getting youtube recommendations " + response.toString());  
54  
55     return response;  
56 } catch (Exception e) {  
57     LogUtil.logError("Failed getting YoutubeVideos: " + e);  
58     return null;  
59 }  
60 }
```

APÊNDICE B – Elaboração de conjunto de dados para o Apache Mahout

```

1 private void getDataModel(String currentUserId ,
2     List<LearningMaterial> youtubeItems ,
3     Map<Long, LearningMaterial> codeLearningMaterial ,
4     FastByIDMap<PreferenceArray> userData) {
5
6     Set<Rating> ratings = getAllRatings();
7
8     Map<String, Long> videoIdCode = new HashMap<String, Long>();
9     Map<String, Long> userIdCode = new HashMap<String, Long>();
10
11    Map<Long, List<Preference>> userCodePreferences =
12        new HashMap<Long, List<Preference>>();
13
14    long materialCont = 01;
15    long userCont = 11;
16    for (Rating ratedItem : ratings) {
17        List<Preference> preferences = new ArrayList<Preference>();
18        String videoId = ratedItem.getLearningMaterial().getVideoId();
19        Float rating = ratedItem.getRating();
20        String userId = ratedItem.getUserId();
21        LearningMaterial learningMaterial =
22            ratedItem.getLearningMaterial();
23
24        if (currentUserId.equals(ratedItem.getUserId())) {
25            if (videoIdCode.containsKey(videoId)) {
26                long videoCode = videoIdCode.get(videoId);
27
28                if (userCodePreferences.containsKey(01)) {
29                    preferences = userCodePreferences.get(01);
30                    preferences.add(
31                        new GenericPreference(0, videoCode, rating));
32                    userCodePreferences.put(01, preferences);
33                } else {
34                    preferences.add(
35                        new GenericPreference(0, videoCode, rating));
36                    userCodePreferences.put(01, preferences);
37                }
38            } else {
39                if (userCodePreferences.containsKey(01)) {
40                    preferences = userCodePreferences.get(01);

```

```
41         preferences.add(new GenericPreference(
42             0, materialCont, rating));
43
44         userCodePreferences.put(01, preferences);
45     } else {
46         preferences.add(new GenericPreference(
47             0, materialCont, rating));
48
49         userCodePreferences.put(01, preferences);
50     }
51
52     codeLearningMaterial.put(materialCont, learningMaterial);
53     videoIdCode.put(learningMaterial.getVideoId(),
54         materialCont);
55     materialCont++;
56 }
57
58     userData.put(0, new GenericUserPreferenceArray(preferences));
59     continue;
60 }
61
62 if (userIdCode.containsKey(userId)) {
63     long userCode = userIdCode.get(userId);
64
65     if (videoIdCode.containsKey(videoId)) {
66         long videoCode = videoIdCode.get(videoId);
67
68         if (userCodePreferences.containsKey(userCode)) {
69             preferences = userCodePreferences.get(userCode);
70             preferences.add(new GenericPreference(userCode,
71                 videoCode, rating));
72             userCodePreferences.put(userCode, preferences);
73         } else {
74             preferences.add(new GenericPreference(userCode,
75                 videoCode, rating));
76             userCodePreferences.put(userCode, preferences);
77         }
78     } else {
79         if (userCodePreferences.containsKey(userCode)) {
80             preferences = userCodePreferences.get(userCode);
81             preferences.add(new GenericPreference(userCode,
82                 materialCont, rating));
83             userCodePreferences.put(userCode, preferences);
84         } else {
85             preferences.add(new GenericPreference(userCode,
86                 materialCont, rating));
87             userCodePreferences.put(userCode, preferences);
```

```
88         }
89
90         codeLearningMaterial.put(materialCont, learningMaterial);
91         videoIdCode.put(learningMaterial.getVideoId(),
92             materialCont);
93         materialCont++;
94     }
95
96     userData.put(0, new GenericUserPreferenceArray(preferences));
97     continue;
98 }
99
100 userIdCode.put(userId, userCont);
101
102 if (videoIdCode.containsKey(videoId)) {
103     long videoCode = videoIdCode.get(videoId);
104
105     if (userCodePreferences.containsKey(userCont)) {
106         preferences = userCodePreferences.get(userCont);
107         preferences.add(new GenericPreference(
108             userCont, videoCode, rating));
109
110         userCodePreferences.put(userCont, preferences);
111     } else {
112         preferences.add(new GenericPreference(
113             userCont, videoCode, rating));
114
115         userCodePreferences.put(userCont, preferences);
116     }
117 } else {
118     if (userCodePreferences.containsKey(userCont)) {
119         preferences = userCodePreferences.get(userCont);
120         preferences.add(new GenericPreference(userCont,
121             materialCont, rating));
122         userCodePreferences.put(userCont, preferences);
123     } else {
124         preferences.add(new GenericPreference(userCont,
125             materialCont, rating));
126         userCodePreferences.put(userCont, preferences);
127     }
128
129     codeLearningMaterial.put(materialCont, learningMaterial);
130     videoIdCode.put(learningMaterial.getVideoId(), materialCont);
131     materialCont++;
132 }
133
134 userCont++;
```



```
135
136     continue;
137 }
138
139 for (Long userCode : userCodePreferences.keySet()) {
140     userData.put(userCode, new GenericUserPreferenceArray(
141         userCodePreferences.get(userCode)));
142 }
143 }
```

APÊNDICE C – Algoritmo de Recomendação

```
1 private List<RecommendedItem> getRecommendations(  
2     FastByIDMap<PreferenceArray> userData) throws TasteException {  
3  
4     DataModel model = new GenericDataModel(userData);  
5     UserSimilarity similarity = new SpearmanCorrelationSimilarity(model);  
6     UserNeighborhood neighborhood = new NearestUserNeighborhood(  
7         (int) (userRepository.count() * 0.25), similarity, model);  
8  
9     UserBasedRecommender recommender = new GenericUserBasedRecommender(  
10        model, neighborhood, similarity);  
11  
12    if (evaluate) {  
13        EvaluateRecommender runnable = new EvaluateRecommender(model,  
14            neighborhood, similarity, NUMBER_RECOMMENDATIONS);  
15  
16        Thread thread = new Thread(runnable);  
17        thread.start();  
18    }  
19  
20    List<RecommendedItem> recommendations = recommender.recommend(0,  
21        NUMBER_RECOMMENDATIONS);  
22  
23    LogUtil.logDebug(recommendations.toString());  
24  
25    return recommendations;  
26 }
```

APÊNDICE D – Algoritmo de Avaliação do SR

```
1 public void run() {
2     try {
3         RecommenderBuilder recommenderBuilder = new RecommenderBuilder(
4             neighborhood, similarity);
5
6         recommenderBuilder.buildRecommender(model);
7
8         RecommenderIRStatsEvaluator statsEvaluator =
9             new GenericRecommenderIRStatsEvaluator();
10
11        IRStatistics stats = statsEvaluator.evaluate(recommenderBuilder,
12            null, model, null, numberRecommendations,
13            GenericRecommenderIRStatsEvaluator.CHOOSE_THRESHOLD, 1.0);
14
15        LogUtil.logDebug("F1Measure " + stats.getF1Measure());
16        LogUtil.logDebug("FallOut " + stats.getFallOut());
17        LogUtil.logDebug("NormalizedDiscountedCumulativeGain "
18            + stats.getNormalizedDiscountedCumulativeGain());
19
20        LogUtil.logDebug("Precision " + stats.getPrecision());
21        LogUtil.logDebug("Reach " + stats.getReach());
22        LogUtil.logDebug("Recall " + stats.getRecall());
23    } catch (TasteException e) {
24        e.printStackTrace();
25    }
26 }
```