

UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Mattheus Einloft

**DESENVOLVIMENTO DE UMA APLICAÇÃO *WEB* DE  
COMPARTILHAMENTO DE CONTEÚDOS DE GRADUAÇÃO COM USO  
DE FILTRAGEM COLABORATIVA**

Santa Maria, RS  
2021

**Mattheus Einloft**

**DESENVOLVIMENTO DE UMA APLICAÇÃO *WEB* DE COMPARTILHAMENTO DE  
CONTEÚDOS DE GRADUAÇÃO COM USO DE FILTRAGEM COLABORATIVA**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**. Defesa realizada por videoconferência.

ORIENTADOR: Prof. Sérgio Luis Sardi Mergen

---

©2021

Todos os direitos autorais reservados a Mattheus Einloft. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

End. Eletr.: [meinloft@inf.ufsm.br](mailto:meinloft@inf.ufsm.br)

**Mattheus Einloft**

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB DE COMPARTILHAMENTO DE CONTEÚDOS DE GRADUAÇÃO COM USO DE FILTRAGEM COLABORATIVA**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Bacharel em Ciência da Computação**.

**Aprovado em 30 de agosto de 2021:**



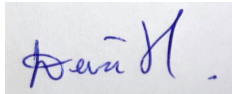
---

**Sérgio Luis Sardi Mergen, Dr. (UFSM)**  
(Presidente/Orientador)



---

**Giliane Bernardi, Dr. (UFSM)**



---

**Deise de Brum Saccol, Dr. (UFSM)**

## RESUMO

### DESENVOLVIMENTO DE UMA APLICAÇÃO *WEB* DE COMPARTILHAMENTO DE CONTEÚDOS DE GRADUAÇÃO COM USO DE FILTRAGEM COLABORATIVA

AUTOR: Mattheus Einloft

ORIENTADOR: Sérgio Luis Sardi Mergen

O avanço da Internet permitiu que ela se consolidasse como uma importante aliada na vida dos estudantes universitários. Existem diversas possibilidades para estudar *online*, como ler livros digitais, assistir aulas através de vídeos ou até mesmo participar de fóruns de discussão. Porém, a abundância de informações disponíveis em diversas fontes acaba dificultando aos estudantes encontrarem os conteúdos mais esclarecedores e com maior qualidade, tornando suas experiências de aprendizado menos eficientes. Com o intuito de solucionar isso, o presente trabalho aborda o projeto e o desenvolvimento de uma aplicação *web* de compartilhamento de conteúdos relacionados à disciplinas de graduação, com foco na recomendação de conteúdos por meio da técnica de filtragem colaborativa. O protótipo desenvolvido, além de permitir aos estudantes compartilharem, avaliarem e visualizarem conteúdos de graduação, realiza recomendações de novos conteúdos com base em estudantes que tenham as mesmas preferências.

**Palavras-chave:** Aplicação *web*. Compartilhamento de conteúdo. Graduação. Filtragem Colaborativa. Sistemas de Recomendação. Desenvolvimento *web*. Colaboração.

## ABSTRACT

### DEVELOPMENT OF AN UNDERGRADUATE CONTENTS SHARING WEB APPLICATION USING COLLABORATIVE FILTERING

AUTHOR: Mattheus Einloft

ADVISOR: Sérgio Luis Sardi Mergen

As the Internet evolved, it became an important ally in the lives of college students. There are several possibilities to study online, such as reading digital books, watching videos and participating in discussion forums. However, the huge amount of information available from several sources makes it difficult for students to find the higher quality contents, leading to a poor learning experience. In order to solve this, this work performs the analyzes and development of a web application for sharing contents related to undergraduate disciplines, focusing on content recommendation through the collaborative filtering technique. The developed prototype allows students to share, rate and view undergraduate contents. More importantly, it recommend contents based on other students who share the same interests.

**Keywords:** Web application. Content sharing. Graduation. Collaborative Filtering. Recommender Systems. Web development. Collaboration.

## LISTA DE FIGURAS

Figura 2.1 – Representação da Recomendação Baseada em Conteúdo. ....	13
Figura 2.2 – Representação da Recomendação Colaborativa. ....	15
Figura 3.1 – “O quão difícil você considera encontrar conteúdos de disciplinas de graduação gratuitos e de boa qualidade na Internet?” .....	21
Figura 3.2 – Respostas para as 4 <sup>a</sup> e 5 <sup>a</sup> perguntas da pesquisa .....	21
Figura 3.3 – Diagrama de casos de uso do módulo Usuários. ....	24
Figura 3.4 – Protótipos visuais das funcionalidades “Criar conta” e “Fazer <i>login</i> ”. ....	24
Figura 3.5 – Diagrama de casos de uso do módulo Tópicos. ....	25
Figura 3.6 – Protótipo visual da página principal. ....	26
Figura 3.7 – Protótipo visual da funcionalidade “Criar tópico”. ....	27
Figura 3.8 – Protótipo visual de um tópico. ....	27
Figura 3.9 – Protótipo visual da funcionalidade “Visualizar tópicos recomendados”. ..	28
Figura 3.10 – Exemplo de uma recomendação de tópicos. ....	29
Figura 4.1 – Arquitetura da aplicação. ....	30
Figura 4.2 – Diagrama Entidade-Relacionamento. ....	31
Figura 4.3 – Arquitetura do <i>back-end</i> . ....	33
Figura 4.4 – Arquitetura do <i>front-end</i> . ....	35
Figura 4.5 – Primeira sub-consulta. ....	37
Figura 4.6 – Segunda sub-consulta. ....	37
Figura 4.7 – Consulta SQL - Recomendação de tópicos. ....	38
Figura 4.8 – Página de criação de tópico. ....	39
Figura 4.9 – Página principal do sistema. ....	39
Figura 4.10 – Página do Tópico 1. ....	40
Figura 4.11 – Página do Tópico 1 (considerando que o usuário A curtiu o tópico 4). ..	40

## LISTA DE QUADROS

Quadro 3.1 – Atores do sistema e suas descrições. ....	23
Quadro 4.1 – Rotas da API e suas descrições. ....	33
Quadro 4.2 – Curtidas dos usuários no exemplo da Figura 3.10 .....	38



## LISTA DE ABREVIATURAS E SIGLAS

<i>API</i>	Application Programming Interface
<i>REST</i>	Representational State Transfer
<i>HTTP</i>	HyperText Transfer Protocol
<i>SGBD</i>	Sistema de Gerenciamento de Banco de Dados
<i>ORM</i>	Object Relational Mapper
<i>SQL</i>	Structured Query Language
<i>CORS</i>	Cross-origin resource sharing
<i>JWT</i>	JSON Web Tokens
<i>HTML</i>	Hypertext Markup Language
<i>CSS</i>	Cascading Style Sheets
<i>SVG</i>	Scalable Vector Graphics

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>9</b>
1.1	OBJETIVO GERAL .....	10
1.2	OBJETIVOS ESPECÍFICOS .....	10
1.3	ESTRUTURAÇÃO DO TRABALHO .....	10
<b>2</b>	<b>REVISÃO DA LITERATURA</b> .....	<b>12</b>
2.1	INTRODUÇÃO À SISTEMAS DE RECOMENDAÇÃO .....	12
2.2	RECOMENDAÇÃO BASEADA EM CONTEÚDO .....	12
2.3	RECOMENDAÇÃO COLABORATIVA .....	14
2.4	RECOMENDAÇÃO HÍBRIDA .....	16
2.5	TRABALHOS RELACIONADOS .....	17
<b>3</b>	<b>ANÁLISE DO PROJETO</b> .....	<b>20</b>
3.1	PESQUISA DE OPINIÃO .....	20
3.2	VISÃO GERAL DO PROJETO .....	22
3.3	DEFINIÇÃO DOS REQUISITOS .....	22
<b>3.3.1</b>	<b>Módulo Usuários</b> .....	<b>23</b>
<b>3.3.2</b>	<b>Módulo Tópicos</b> .....	<b>25</b>
3.3.2.1	<i>Visualizar lista de tópicos</i> .....	25
3.3.2.2	<i>Criar tópico</i> .....	26
3.3.2.3	<i>Visualizar tópico, Curtir tópico e Cancelar curtida</i> .....	27
3.3.2.4	<i>Visualizar tópicos recomendados</i> .....	28
<b>4</b>	<b>DESENVOLVIMENTO DO SISTEMA</b> .....	<b>30</b>
4.1	VISÃO GERAL DA ARQUITETURA .....	30
4.2	BASE DE DADOS .....	31
<b>4.2.1</b>	<b>Tecnologias utilizadas</b> .....	<b>32</b>
4.3	<i>BACK-END</i> .....	32
<b>4.3.1</b>	<b>Tecnologias utilizadas</b> .....	<b>34</b>
4.4	<i>FRONT-END</i> .....	35
<b>4.4.1</b>	<b>Tecnologias utilizadas</b> .....	<b>36</b>
4.5	IMPLEMENTAÇÃO DA RECOMENDAÇÃO DE TÓPICOS .....	36
<b>4.5.1</b>	<b>Demonstração da recomendação de tópicos</b> .....	<b>38</b>
<b>5</b>	<b>CONCLUSÃO</b> .....	<b>41</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>43</b>

## 1 INTRODUÇÃO

Com a evolução da tecnologia da informação e de dispositivos computacionais, o uso da Internet está cada vez mais consolidado em nosso dia a dia. Graças à sua enorme capacidade de armazenamento e compartilhamento de informações, temos praticamente tudo à nossa disposição: bate-papo com pessoas de qualquer lugar, busca por informações sobre qualquer assunto, diversas formas de entretenimento, entre outros.

Um dos diversos motivos para utilizarmos a Internet é para o nosso aprendizado, seja através de pesquisas em sites de busca, vídeos, leitura de livros ou também participando de cursos *online*. Com a evolução tecnológica, temos à nossa disposição uma quantidade enorme de informações, em muito pouco tempo.

O aumento do uso de tecnologias da informação e a consolidação da *Web* proporcionaram um alto crescimento da Educação a Distância (EaD) na última década. Hoje é possível cursar uma graduação de forma digital através de Ambientes Virtuais de Aprendizagem (AVA), espaços virtuais para a educação que possibilitam, aos sujeitos envolvidos nos processos de ensino, aprendizagem e avaliação, a busca por conhecimentos e capacitação (MACIEL, 2018).

Com tantas informações à disposição, é cada vez mais comum que estudantes aprendam através da Internet. No entanto, isto não é garantia de qualidade do aprendizado. Mesmo tendo inúmeros conteúdos à disposição, muitos fatores podem fazer com que um estudante não consiga encontrar um material de qualidade, como: a inexistência de conteúdos disponíveis para a linguagem nativa da pessoa; o uso inadequado das palavras-chave em um site de buscas, impedindo que o algoritmo do buscador encontre o conteúdo desejado; e as informações de um assunto muito específico serem escassas, desatualizadas ou até inexistentes.

Com isso, diversas ferramentas tecnológicas foram surgindo, muitas delas buscando solucionar esse problema através de conteúdos próprios, por meio de plataformas com acesso exclusivo à pessoas que fazem uma assinatura paga. Apesar de serem bastante úteis para determinados públicos-alvo, elas acabam limitando seus conteúdos apenas para essas pessoas dispostas a investirem em suas assinaturas. Além disso, deixam de se beneficiar de um fator interessantíssimo: o poder colaborativo da Internet. *GitHub*<sup>1</sup>, *StackOverflow*<sup>2</sup> e *Reddit*<sup>3</sup> são fortes exemplos de ferramentas que sabem utilizar muito bem esse fator de colaboratividade. Através de suas soluções, conseguem fornecer aos usuários um ambiente que incentiva fortemente o compartilhamento de informações entre pessoas, fazendo com que as informações mais relevantes e úteis cheguem ao maior número possível de interessados.

---

<sup>1</sup><https://github.com/>

<sup>2</sup><https://stackoverflow.com/>

<sup>3</sup><https://www.reddit.com/>

Uma estratégia bastante utilizada com o objetivo de facilitar a usuários encontrarem conteúdos em uma ferramenta é o uso de técnicas de **Sistemas de Recomendação**. Basicamente, essas técnicas reconhecem as preferências de seus usuários e, com isso, fazem recomendações de conteúdos potencialmente relevantes aos mesmos. Uma das técnicas conhecidas de recomendação é a **Filtragem Colaborativa**, que se baseia em conteúdos acessados por usuários que tenham interesses compartilhados.

### 1.1 OBJETIVO GERAL

Buscando facilitar aos estudantes universitários a descoberta de conteúdos com maior relevância e, conseqüentemente, levar a eles uma melhor experiência de aprendizado, decidiu-se pelo desenvolvimento de uma aplicação *web* de compartilhamento de conteúdos relacionados à disciplinas de graduação. A aplicação terá como foco a recomendação de conteúdos por meio da técnica de filtragem colaborativa.

### 1.2 OBJETIVOS ESPECÍFICOS

Com o objetivo geral em mente, tem-se os seguintes objetivos específicos:

- Estudar técnicas de Sistemas de Recomendação.
- Estudar as ferramentas a serem utilizadas no desenvolvimento do sistema.
- Desenvolver um sistema tendo como principais funcionalidades:
  - Criação de conteúdos de graduação na plataforma.
  - Avaliação de conteúdos existentes na plataforma.
  - Visualização de conteúdos existentes na plataforma.
  - Recomendação de novos conteúdos por meio da filtragem colaborativa.

### 1.3 ESTRUTURAÇÃO DO TRABALHO

Este trabalho está estruturado da seguinte forma: o Capítulo 2 apresenta uma revisão da literatura, trazendo os conceitos e conhecimentos necessários para melhor compreensão do projeto; o Capítulo 3 tem como foco demonstrar uma visão geral do projeto.

Esse capítulo demonstra uma pesquisa de obtenção de opiniões a respeito do projeto e o levantamento dos principais requisitos; o Capítulo 4 trata do desenvolvimento do *software* através da apresentação de sua arquitetura e das tecnologias definidas. Além disso, o capítulo apresenta questões referentes ao desenvolvimento do sistema, dando ênfase ao tratamento referente à filtragem colaborativa. Por fim, é realizada uma demonstração do protótipo desenvolvido, onde pode-se ver como o sistema de recomendação de conteúdo funciona; e o Capítulo 5 apresenta as considerações finais, juntamente com sugestões para trabalhos futuros.

## 2 REVISÃO DA LITERATURA

Neste capítulo são apresentados os conceitos e conhecimentos necessários para melhor compreensão do projeto. Inicialmente, a Seção 2.1 traz uma introdução sobre Sistemas de Recomendação. Na sequência, a Seção 2.2 enumera conhecimentos à respeito da Recomendação Baseada em Conteúdo. Prosseguindo, a Seção 2.3 aborda sobre Recomendação Colaborativa. Já a Seção 2.4 traz conceitos relacionados à Recomendação Híbrida. Por fim, a Seção 2.5 apresenta trabalhos relacionados.

### 2.1 INTRODUÇÃO À SISTEMAS DE RECOMENDAÇÃO

**Sistemas de Recomendação** são sistemas responsáveis por gerar sugestões de itens úteis para usuários. Um item é a designação do que o sistema recomenda para um usuário, tal como um produto em sistemas de vendas *online*, um filme em sistemas de *streaming* de vídeo ou uma notícia em plataformas de notícias. Exemplos de relevância na indústria que utilizam sistemas de recomendação são a *Amazon*<sup>1</sup> (LINDEN; SMITH; YORK, 2003) e a *Netflix*<sup>2</sup> (GOMEZ-URIBE; HUNT, 2015).

As recomendações geradas por sistemas de recomendação visam auxiliar os usuários em seus processos de tomada de decisão durante a utilização de um *software*, facilitando a busca por conteúdos de interesse em grandes quantidades de informação, melhorando a experiência do usuário e estimulando o uso contínuo do sistema. Consequentemente, é possível obter um aumento no faturamento, seja através da venda de produtos recomendados em lojas *online* (*e-commerces*), ou através de acessos dos usuários em anúncios recomendados de terceiros.

Diversas técnicas foram propostas para a geração de recomendações em Sistemas de Recomendação. As mais comuns são baseadas em algum tipo de filtragem: baseada em conteúdo e colaborativa. É possível combinar duas ou mais técnicas, tendo assim um sistema de recomendação híbrido. Essas técnicas serão esclarecidas com maiores detalhes nas próximas seções.

### 2.2 RECOMENDAÇÃO BASEADA EM CONTEÚDO

**Filtragem Baseada em Conteúdo** é uma técnica de recomendação de itens focada em analisar avaliações de itens feitas por usuários, juntamente com características

---

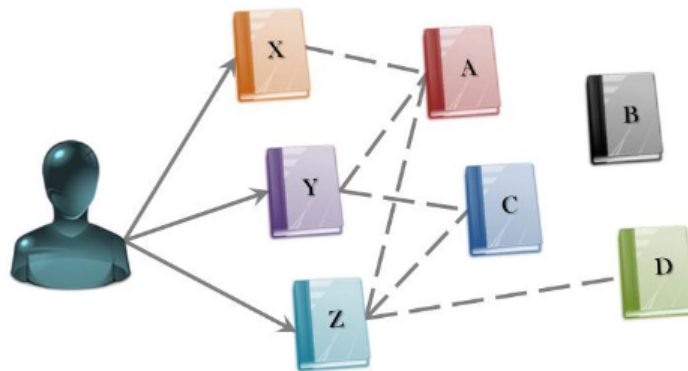
<sup>1</sup><https://www.amazon.com/>

<sup>2</sup><https://www.netflix.com>

associadas a esses itens. Características são particularidades que classificam itens, com o objetivo de agrupá-los em grupos de itens com propriedades similares. Por exemplo, plataformas de *streaming* de música, cujos itens seriam as músicas, utilizam diversos aspectos como características, como gênero, artista e ano de lançamento. Em suma, um sistema de recomendação baseado em conteúdo analisa itens com as mesmas características e recomenda-os a usuários que gostaram de outros itens parecidos.

Na Figura 2.1, há um exemplo de representação da Recomendação Baseada em Conteúdo, elaborado por Costa, Aguiar e Magalhães (2013). Dado um usuário que avaliou positivamente 3 livros X, Y e Z, deseja-se recomendar livros que ele provavelmente se interessaria. Inicia-se o processo analisando os livros que possuem características similares aos livros que o usuário gostou. No exemplo da figura, os livros A e C são os que possuem mais características parecidas com os livros avaliados positivamente pelo usuário. Portanto, os livros A e C serão recomendados ao usuário.

Figura 2.1 – Representação da Recomendação Baseada em Conteúdo.



Fonte: Retirado de Costa, Aguiar e Magalhães (2013).

Apesar de serem bastante utilizados, sistemas de recomendação baseados em conteúdo possuem certas limitações, conforme pontuam Adomavicius e Tuzhilin (2005). Devido ao fato das técnicas baseadas em conteúdo necessitarem da existência de características associadas a itens para serem realizadas, elas acabam sendo limitadas pela explicitação das próprias características. Então, para se obter uma quantidade de características suficiente para a aplicação da recomendação baseada em conteúdo, é necessário que elas possam ser definidas automaticamente por um computador, caso contrário devem ser assinaladas de forma manual. Esse problema é chamado de **Análise de conteúdo limitado**. Embora a extração automática de informações funcione bem em documentos textuais, não se pode dizer o mesmo para a extração de características em dados multimídia, por exemplo vídeos, imagens e áudios. Além disso, a definição manual de características associadas a itens geralmente não é prática, devido a limitação de recursos e a enorme quantidade de itens diferentes. Outro problema decorrente da análise de conteúdo limitado é que, caso dois itens diferentes sejam representados pelo mesmo conjunto de

características, eles tornam-se indistinguíveis. Como documentos textuais são geralmente representados pelas suas palavras-chave mais importantes, sistemas baseados em conteúdo acabam não conseguindo distinguir entre um artigo bem escrito e um mal escrito, se eles usarem os mesmos termos.

Em casos de itens com avaliação alta em relação a um perfil muito específico de usuário, o sistema de recomendação está limitado a recomendar ao usuário itens similares aos que ele já avaliou. Por exemplo, em um sistema de recomendação de restaurantes, uma pessoa inexperiente em culinária grega nunca receberia uma recomendação para um restaurante de culinária grega, mesmo que este seja o melhor restaurante grego da cidade do usuário. De maneira oposta, há casos em que itens não devem ser recomendados se eles são muito similares a algo que o usuário já viu, como por exemplo diferentes artigos de notícias que descrevem o mesmo acontecimento. Essa limitação denomina-se **Super-especialização**. Ela ocasiona uma recomendação de pequenas variedades de opções de itens diferentes aos usuários, dificultando assim a existência de uma diversidade nas recomendações, aspecto geralmente desejado em sistemas de recomendação.

A limitação mais comum é a denominada **Problema do novo usuário**. Um usuário precisa avaliar uma quantidade significativa de itens para que a técnica de recomendação baseada em conteúdo consiga realmente compreender as preferências do usuário e, então, apresentar a ele recomendações confiáveis. Devido a isso, novos usuários, que recentemente começaram a utilizar um sistema e, portanto, realizaram poucas avaliações, receberão recomendações com pouca precisão.

### 2.3 RECOMENDAÇÃO COLABORATIVA

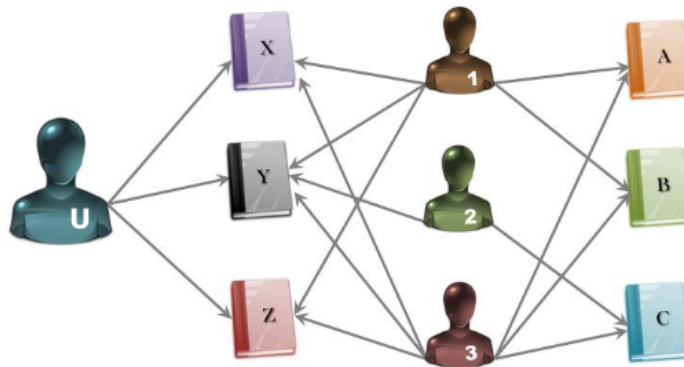
**Filtragem Colaborativa** é uma técnica de recomendação de itens focada em informações obtidas das avaliações de itens realizadas por usuários e das comparações entre eles, cujas semelhanças são estabelecidas através de seus históricos de avaliações. Avaliações são a maneira que o usuário consegue classificar se gosta ou não de um determinado item. Essa classificação pode ser definida de diferentes maneiras, seja através de uma classificação binária, onde o usuário apenas indica seu gosto com um sim ou um não, ou através de uma classificação utilizando valores reais, onde o usuário indica seu gosto através de um valor numérico que está contido dentro de um intervalo de números definido anteriormente.

Na Figura 2.2, há um exemplo de representação da Recomendação Colaborativa, adaptado de Costa, Aguiar e Magalhães (2013). Dado um usuário  $U$  que avaliou positivamente 3 livros  $X$ ,  $Y$  e  $Z$ , deseja-se recomendar livros que ele provavelmente se interessaria. Inicia-se o processo analisando os usuários que avaliaram de maneira positiva pelo menos um dos livros que o usuário  $U$  também avaliou. Assim, deve-se analisar os usuários  $1$ ,  $2$



e 3, encontrando os mais semelhantes ao usuário  $U$ . Desses usuários, os mais similares são os usuários 1 e 3, pois ambos avaliaram positivamente os 3 livros  $X$ ,  $Y$  e  $Z$ . Finalizando, deve-se verificar outros livros que os usuários 1 e 3 avaliaram de forma positiva e, então, recomendá-los ao usuário  $U$ . Nesse exemplo, como ambos avaliaram positivamente os livros  $A$  e  $B$ , estes então serão recomendados ao usuário  $U$ .

Figura 2.2 – Representação da Recomendação Colaborativa.



Fonte: Adaptado de Costa, Aguiar e Magalhães (2013).

Sistemas de recomendação colaborativos não possuem algumas das limitações que os sistemas baseados em conteúdo possuem. Como a filtragem colaborativa utiliza avaliações de outros usuários, ela consegue lidar com qualquer tipo de conteúdo e recomendar quaisquer itens, inclusive os que são diferentes dos já avaliados anteriormente.

No entanto, a recomendação colaborativa possui suas próprias limitações, conforme pontuam Adomavicius e Tuzhilin (2005). Uma delas é a mesma que ocorre na filtragem baseada em conteúdo, o **Problema do novo usuário**: com a finalidade de realizar recomendações efetivas, o sistema deve primeiro entender as preferências do usuário através de suas avaliações de itens. Enquanto não houver um número substancial de avaliações de um usuário, ele não conseguirá receber recomendações confiáveis. Esse problema pode ser resolvido através de várias técnicas, sendo que a maioria delas utiliza a abordagem de Recomendação Híbrida, a qual combina mais de uma técnica de recomendação em um mesmo sistema.

Para que consiga realizar recomendações, a filtragem colaborativa depende exclusivamente das preferências dos usuários. Como novos itens são adicionados de maneira regular à sistemas de recomendação, a filtragem colaborativa poderá não ser capaz de recomendar estes itens enquanto eles não tiverem sido avaliados por uma quantidade significativa de usuários. Essa limitação, denominada de **Problema do novo item**, pode ser resolvida utilizando a abordagem de Recomendação Híbrida, que será explicada com maiores detalhamentos na próxima seção.

O sucesso de um sistema de recomendação colaborativo depende da disponibilidade de uma grande quantidade de usuários. Por exemplo, em um sistema de reco-

mendação de filmes, podem existir vários filmes que foram avaliados por uma pequena quantidade de pessoas. Esses filmes seriam recomendados muito raramente, mesmo que esses poucos usuários dessem avaliações altas. Além disso, para o usuário cujo gosto é incomum comparado ao resto da população, não haverá outros usuários que sejam particularmente semelhantes, levando-o a recomendações ruins. Essa limitação é denominada **Escassez de avaliações**. Uma forma de resolver esse problema é usar informações do perfil do usuário ao calcular sua similaridade com outros usuários. Desta forma, dois usuários podem ser considerados similares não apenas se eles avaliaram os mesmos itens de maneira similar, mas também se possuem características similares entre si.

## 2.4 RECOMENDAÇÃO HÍBRIDA

Conforme pontuado por Burke (2007) e descrito nas seções anteriores, cada técnica de recomendação possui certas limitações. Com o objetivo de evitar essas limitações e, conseqüentemente, tornar as recomendações mais precisas, começou-se a utilizar duas ou mais técnicas em conjunto, combinando-as através de um **Sistema de Recomendação Híbrido**, para que a desvantagem de uma técnica seja amenizada pela vantagem de outra. Por exemplo, recomendações colaborativa e baseada em conteúdo podem ser combinadas para que a técnica baseada em conteúdo possa compensar o problema do novo usuário da técnica colaborativa, provendo assim recomendações para novos usuários cujos perfis possuem poucas informações suficientes para que a recomendação colaborativa consiga realizar recomendações assertivas.

Para combinar duas ou mais técnicas de recomendação em um sistema híbrido, diferentes métodos de combinação foram propostos por Burke (2002): Ponderado, Comutação, Misto, Combinação de características, Cascata, Aumento de características e Meta-nível.

No método de recomendação **ponderado** definem-se, separadamente para cada técnica de recomendação presente no sistema, pontuações para os itens recomendados, e então realiza-se uma combinação linear entre essas pontuações. Em alguns casos, atribuem-se pesos inicialmente iguais para cada técnica de recomendação e, conforme os usuários vão utilizando o sistema, os pesos são adaptados através do feedback dos usuários.

Um sistema de recomendação híbrido que utiliza o método de **comutação** é aquele que utiliza algum critério de decisão para alternar entre técnicas de recomendação. Diante de uma certa situação, uma determinada técnica de recomendação é selecionada. Em outros casos, diferentes técnicas podem ser escolhidas.

Através do método de recomendação **misto**, um sistema apresenta recomendações de diferentes técnicas em uma lista combinada, sem nenhuma tentativa de combi-

nar evidências entre os recomendadores. O ranqueamento das recomendações pode ser realizado através da fusão com base em uma classificação prevista ou na confiança do recomendador (BURKE, 2007).

A ideia do método **combinação de características** é inserir características de uma técnica de recomendação em um algoritmo designado para processar dados utilizando outra técnica (BURKE, 2007). Dessa forma, as características que normalmente seriam processadas por um determinado recomendador, são usadas como entrada para outro recomendador.

A recomendação híbrida em **cascata** envolve um processo em etapas. Primeiramente, uma técnica de recomendação é utilizada para produzir uma grande quantidade de candidatos. Então, uma segunda técnica é utilizada para refinar a recomendação a partir desses candidatos (BURKE, 2002). Dessa forma, pode ser considerada uma recomendação estritamente hierárquica, na qual uma técnica de recomendação fraca não pode anular decisões tomadas por uma mais forte, somente refiná-las (BURKE, 2007).

**Aumento de características** é uma estratégia para recomendação híbrida que é similar em alguns aspectos a combinação de características. Ao invés de usar diretamente as características utilizadas em uma segunda técnica de recomendação, o método de aumento de características gera uma nova característica para cada item usando a lógica de recomendação dessa técnica (BURKE, 2007).

O método **Meta-nível** é aquele que usa um modelo aprendido por uma técnica de recomendação como entrada para outra (BURKE, 2007). Esse processo é diferente do que ocorre no método de aumento de características: nele, é utilizado um modelo aprendido para gerar características como entrada para um segundo algoritmo. Já no meta-nível, o modelo inteiro se torna a entrada (BURKE, 2002).

## 2.5 TRABALHOS RELACIONADOS

Diversos trabalhos desenvolvidos no meio acadêmico abordam o desenvolvimento de um *software* com o objetivo de melhorar a experiência do aprendizado de estudantes. Esta seção apresenta alguns destes trabalhos, sendo todos relacionados à educação e um deles que faz uso de Sistemas de Recomendação. Ao final da seção, é feita uma relação dos trabalhos relacionados com o presente trabalho.

O trabalho elaborado por Torrado (2016) tem como propósito reunir diversas informações referentes a estudo em apenas um local. Essas informações geralmente estão espalhadas em inúmeras ferramentas na Internet, o que dificulta a sua organização por parte dos estudantes. Além disso, o trabalho busca facilitar o uso e o acesso, diferentemente das plataformas existentes que geralmente são exclusivas para determinados grupos (por exemplo, para a comunidade de uma universidade específica) ou possuem alguma limita-

ção de acesso (por exemplo, recursos exclusivos para membros assinantes).

A plataforma de Torrado (2016) é organizada em diversos componentes, sendo os primários: Grupos, Usuários e Conteúdos. Os principais são os grupos, componentes que representam os espaços virtuais nos quais ocorrem a interação entre os utilizadores da plataforma. Dentro de cada grupo constam os conteúdos (informações criadas ou compartilhadas pelos usuários) e os usuários (responsáveis pelo gerenciamento de grupos e conteúdos). Ademais, existem outros componentes secundários, como os Controles de Acesso (definição de papéis e permissões dos usuários), Ações (representação das ações dos usuários na plataforma) e Avaliações (quantificação da relevância de um conteúdo para um determinado usuário).

Outro exemplo de trabalho relacionado à educação é o de Calle-Alonso et al. (2017), cujo objetivo central é abordar a educação *online* na perspectiva das novas tendências em *e-learning* e princípios de neurodidática, através de uma plataforma denominada NeuroK. Diferentemente da maioria das ferramentas existentes, cujo foco principal é nos conteúdos disponibilizados em suas plataformas, a NeuroK foca no estudante em si, em melhorar a qualidade do seu aprendizado, através de técnicas como gamificação, aprendizado cooperativo e tutoria de pares. A ideia é gerar engajamento e motivação para facilitar o aprendizado a longo prazo, ao invés de apenas apresentar novas informações aos estudantes.

Para atingir os objetivos propostos, Calle-Alonso et al. (2017) construíram sua solução como sendo uma rede social, permitindo que participantes compartilhem documentos, façam comentários, marquem itens como favoritos, realizem avaliações e solucionem problemas de forma cooperativa, tudo isso com uma geração de estatísticas de uso para a gamificação da plataforma. Por exemplo, através das denominadas Unidades de Aprendizado, conteúdos para debate são propostos, juntamente com atividades, questões e problemas práticos sobre esses conteúdos. Assim, usuários podem solucioná-los de forma individual ou colaborativa, aprendendo de forma prática e conseqüentemente absorvendo mais facilmente o conteúdo.

Taveira-Gomes et al. (2014) pontuam que tecnologias de aprendizado emergentes centradas na educação médica focam principalmente em maximizar a eficiência do ensino administrativo e pouco em tarefas de aprendizagem voltadas a alunos. Com o objetivo de auxiliar os estudantes da área médica na tomada de decisões sobre o seu processo de aprendizagem pessoal, propuseram uma plataforma colaborativa de *e-learning* para estudantes de medicina, denominada *ALERT STUDENT*.

O conteúdo da plataforma *ALERT STUDENT* foi estruturado através de blocos reutilizáveis que permitem a construção de blocos de aprendizagem. O menor bloco de aprendizagem, chamado de *Flashcard*, é composto por informações em um lado e questões abertas no outro. Cada *flashcard* contém até 8 peças de conhecimento denominadas Fato, Descrição e Imagem. Essas peças podem ser associadas a questões, servindo assim como resposta a suas respectivas questões.

Para atingir determinados objetivos de aprendizado, *flashcards* de categorias diferentes podem ser agregados entre si, formando assim blocos de aprendizado maiores denominados Cadernos. Esses conteúdos podem ser criados e compartilhados por professores e estudantes através de Grupos, que são conjuntos de usuários agrupados de forma global ou separados por instituições de ensino.

O trabalho de Lichtnow et al. (2006), diferentemente dos demais, utiliza técnicas de recomendação em sua solução. O sistema proposto por eles, denominado *SisRecCol*, busca apoiar o processo de aprendizagem colaborativa, através de recomendações feitas a partir da análise do comportamento de usuários em um *web chat* e do acesso a uma biblioteca digital.

Conforme um usuário utiliza o *chat*, ocorre um processo de mineração de texto para acompanhar as trocas de mensagens e identificar os assuntos tratados a partir de consultas feitas na biblioteca. Finalizada a identificação dos assuntos tratados, os mesmos são enviados ao módulo de recomendação, que é responsável por consultar a base de perfis e então gerar as recomendações para cada um dos usuários participantes.

Como já citado, a maioria dos trabalhos relacionados não aplicou técnicas de recomendação em suas soluções. Torrado (2016) até pontuou que seu sistema poderia ter sido mais útil para o usuário caso fosse combinado com algoritmos de recomendação, porém, por questões de simplicidade, optou por não desenvolvê-los. Em contrapartida, este trabalho propõe a implementação de um sistema de recomendação para possibilitar a recomendação de tópicos potencialmente relevantes aos usuários. Com isso, a tendência é que usuários se sintam mais incentivados a seguir interagindo com a plataforma.

O sistema de Lichtnow et al. (2006) faz uso de recomendações através da filtragem baseada em conteúdo. Isto faz com que sua aplicação dependa da interação com o *chat* para funcionar (problema da análise do conteúdo limitado). Já o presente trabalho utiliza filtragem colaborativa, fazendo com que o sistema não dependa de instrumentos adicionais para que sejam feitas as recomendações.

### 3 ANÁLISE DO PROJETO

Este capítulo visa realizar uma análise do projeto proposto pelo presente trabalho da seguinte forma: a Seção 3.1 apresenta uma pesquisa de opinião a respeito do sistema proposto; a Seção 3.2 apresenta uma visão geral do projeto; e na Seção 3.3 tem-se a definição dos requisitos do sistema, onde são demonstrados o módulo Usuários (Subseção 3.3.1) e o módulo Tópicos (Subseção 3.3.2).

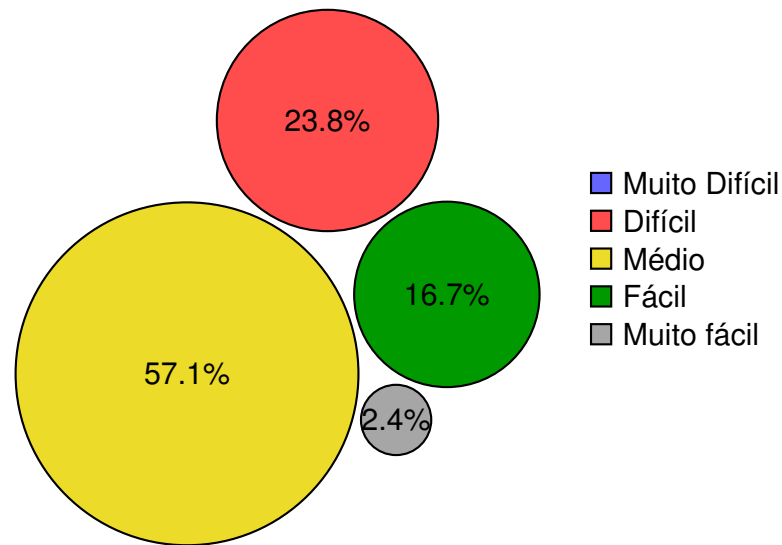
#### 3.1 PESQUISA DE OPINIÃO

Com a finalidade de obter validações e sugestões a respeito do sistema e suas funcionalidades, foi realizada uma pesquisa em que estudantes universitários ou já graduados responderam à perguntas e puderam trazer sugestões a respeito da proposta do projeto. Ao total foram cinco perguntas, respondidas por 42 alunos e graduados, sendo 69% dos cursos de Sistemas de Informação, Ciência da Computação, Análise e Desenvolvimento de Sistemas e Sistemas para Internet; 12% de cursos de Engenharias; 9,5% dos cursos de Administração e Ciências Contábeis; e 9,5% de outros cursos (Jornalismo, Farmácia, Gestão da Qualidade e Letras).

As duas primeiras perguntas da pesquisa foram: “Você utiliza frequentemente a Internet para seus estudos?”, que foi respondida como “Sim” por 100% dos participantes; e “Já teve dificuldades ao encontrar algum material específico na Internet sobre uma determinada disciplina de graduação?”, a qual teve 86% das respostas como sendo “Sim”.

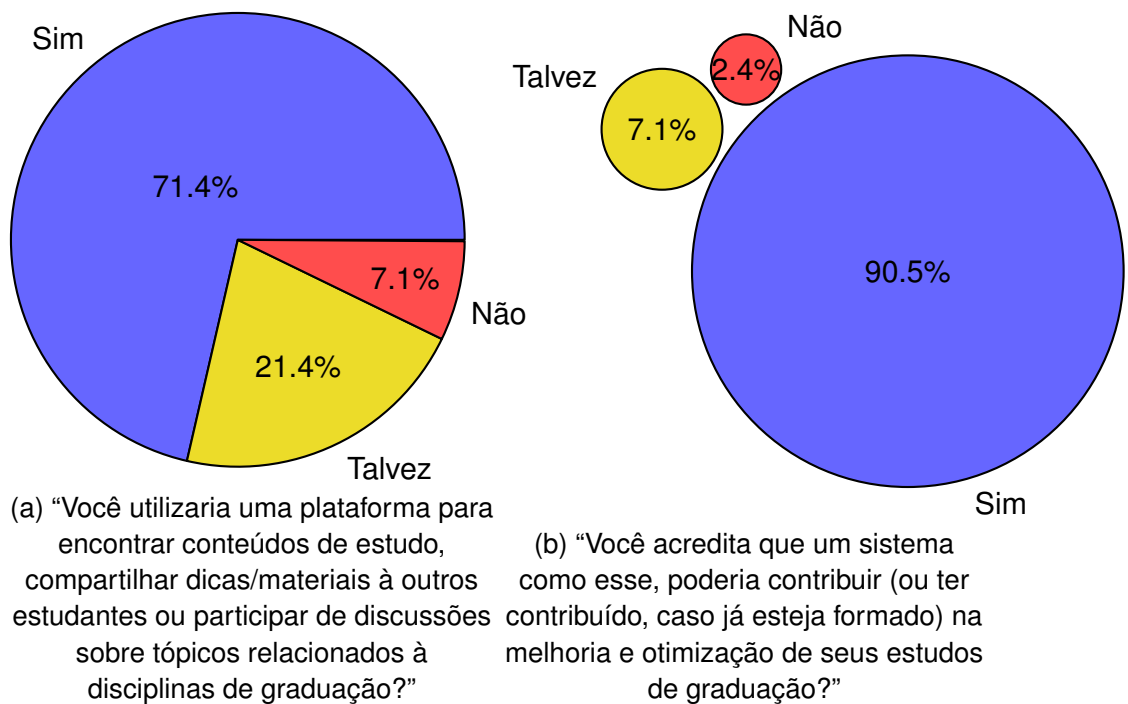
A terceira pergunta foi: “O quão difícil você considera encontrar conteúdos de disciplinas de graduação gratuitos e de boa qualidade na Internet?”. Suas respostas estão demonstradas no gráfico da Figura 3.1.

Figura 3.1 – “O quão difícil você considera encontrar conteúdos de disciplinas de graduação gratuitos e de boa qualidade na Internet?”



A Figura 3.2 apresenta as respostas para as últimas perguntas: “Você utilizaria uma plataforma para encontrar conteúdos de estudo, compartilhar dicas/materiais à outros estudantes ou participar de discussões sobre tópicos relacionados à disciplinas de graduação?” (Figura 3.2a); e “Você acredita que um sistema como esse, poderia contribuir (ou ter contribuído, caso já esteja formado) na melhoria e otimização de seus estudos de graduação?” (Figura 3.2b).

Figura 3.2 – Respostas para as 4<sup>a</sup> e 5<sup>a</sup> perguntas da pesquisa



Analisando o resultado da pesquisa, percebe-se que as respostas obtidas reforçam a necessidade de sistemas como este que está sendo proposto.

### 3.2 VISÃO GERAL DO PROJETO

O presente trabalho propõe uma plataforma *web* de compartilhamento de conteúdos referentes à disciplinas de graduação, com uso de um sistema de recomendação desses conteúdos. Através dessa plataforma, busca-se fornecer um ambiente virtual que auxilie universitários em seus estudos durante o período em que estão cursando uma graduação.

Usuários poderão acessar o sistema por meio de um navegador *web*. Em seu primeiro acesso, precisarão criar uma conta na plataforma. Tendo uma conta criada, será necessário logarem na aplicação. Após o *login*, os usuários serão direcionados à página principal da aplicação, onde será possível encontrar e utilizar as funcionalidades da plataforma (que serão demonstradas com maiores detalhes na Seção 3.3).

Tendo acesso ao sistema, estudantes interagirão na plataforma majoritariamente através de **Tópicos**, conteúdos referentes à disciplinas de graduação. Será possível visualizar e avaliar tópicos criados por outros estudantes, além de criar novos. Essas ações, com o tempo, tornarão a plataforma cada vez mais eficiente e útil aos seus usuários.

Para auxiliar os estudantes à encontrarem conteúdos que lhe sejam mais relevantes, o sistema irá realizar recomendações de tópicos para os usuários. Estes tópicos a serem recomendados serão definidos a partir da avaliação de outros usuários da plataforma (Filtragem Colaborativa). Dessa forma, estudantes com gostos parecidos serão recomendados à tópicos semelhantes. A opção pela técnica da filtragem colaborativa deveu-se ao alto custo da busca por itens relacionados da abordagem baseada em conteúdo (LINDEN; SMITH; YORK, 2003) e um maior nível de complexidade da abordagem híbrida.

### 3.3 DEFINIÇÃO DOS REQUISITOS

De acordo com Sommerville (2011), os requisitos de um *software* representam as descrições do que o mesmo deve fazer, os serviços que oferece e as restrições a seu funcionamento. Em outras palavras, eles refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada.

Casos de uso são uma forma de especificar os requisitos funcionais de um sistema. Eles servem para descrever as interações típicas entre os usuários de um sistema e o próprio sistema, fornecendo uma narrativa sobre como o mesmo é utilizado (FOWLER, 2003).



Um diagrama de caso de uso possui três componentes básicos: **Caso de Uso**, representado por uma elipse; **Ator**, representado através de um boneco palito; e **Relacionamento**, representado por uma linha.

O Quadro 3.1 demonstra os atores do sistema e suas descrições. O ator **Visitante** representa o usuário que não possui uma conta cadastrada na plataforma ou que não realizou o *login*. Quando este ator criar uma conta e prosseguir com o processo de *login*, ele deixará de ser um Visitante e será representado pelo ator **Usuário**.

Quadro 3.1 – Atores do sistema e suas descrições.

<b>Ator</b>	<b>Descrição</b>
Visitante	Visitante da plataforma, acessada através de um navegador <i>web</i> . Possui permissão para visualizar tópicos e criar uma conta.
Usuário	Usuário com conta criada e que está logado na aplicação. Pode colaborar criando novos tópicos e curtindo os existentes.

Os diagramas de caso de uso da aplicação foram definidos separadamente para componentes chamados de **Módulos**. Cada módulo representa uma abstração da aplicação e contém apenas as funcionalidades que são relacionadas a seu domínio. Dessa forma, será possível ter um melhor entendimento do funcionamento do sistema, aprimorando sua organização e tendo uma melhor separação de responsabilidades. Os principais módulos da aplicação são: **Usuários** e **Tópicos**. As subseções a seguir demonstram estes módulos com mais detalhes, o que inclui a apresentação de protótipos de tela (*wireframes*) relacionados às principais funcionalidades.

### 3.3.1 Módulo Usuários

A Figura 3.3 demonstra os casos de uso referentes ao módulo Usuários. Este módulo é o responsável pelas funcionalidades mais básicas de um sistema *web*, como criação de conta (Figura 3.4a), realização de *login* (Figura 3.4b) e *logout*.

Figura 3.3 – Diagrama de casos de uso do módulo Usuários.

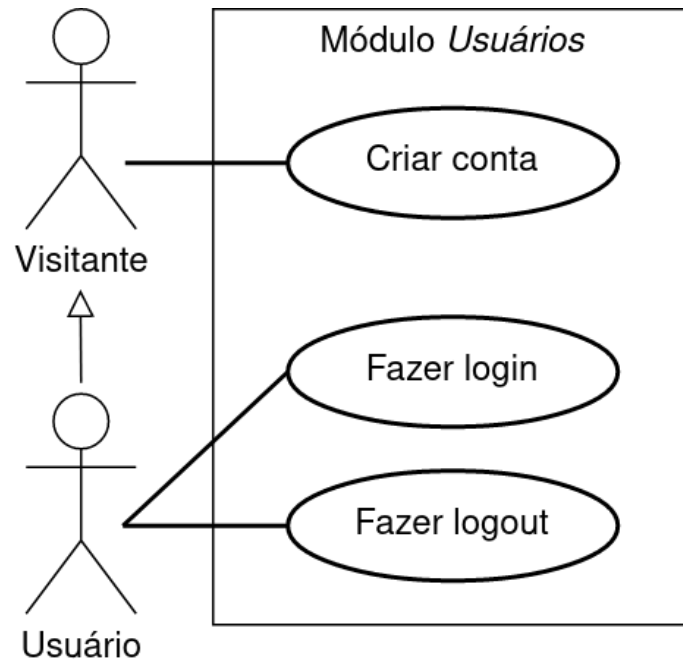


Figura 3.4 – Protótipos visuais das funcionalidades “Criar conta” e “Fazer login”.

### Criar conta

Nome

E-mail

Senha

Universidade ▼

Curso ▼

**Cadastrar**

Já tem uma conta?  
[Faça Login](#)

(a) Criar conta

### Fazer login

E-mail

Senha

**Entrar**

---

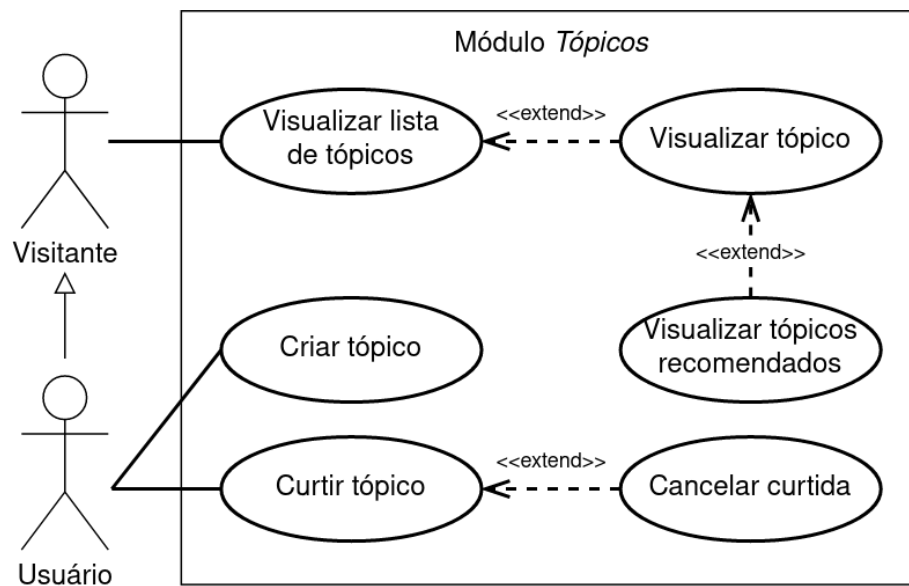
Não tem uma conta?  
[Criar conta](#)

(b) Fazer login

### 3.3.2 Módulo Tópicos

O módulo Tópicos tem seus casos de uso descritos no diagrama da Figura 3.5. **Tópico** é a principal forma de interação entre usuários na plataforma. É composto por informações como título, descrição, disciplina, curso de graduação e universidade. A descrição é o conteúdo do tópico, podendo assumir várias formas, como um resumo, uma lista de exercícios ou uma dúvida. Um tópico deve estar imprescindivelmente vinculado a uma específica disciplina de um determinado curso de graduação, o qual pertence a uma determinada universidade.

Figura 3.5 – Diagrama de casos de uso do módulo Tópicos.



#### 3.3.2.1 Visualizar lista de tópicos

Ao acessar o sistema por um navegador *web*, o visitante (ou usuário) é levado à página principal da aplicação (Figura 3.6). Esta página traz à ele uma lista de tópicos criados na plataforma. Esta lista é ordenada, de forma que os tópicos são sempre apresentados dos mais curtidos até os menos curtidos.

Figura 3.6 – Protótipo visual da página principal.



Como já mencionado, esta é a tela principal, a partir da qual o usuário tem acesso à outras funcionalidades. As subseções a seguir descrevem essas funcionalidades.

### 3.3.2.2 Criar tópico

Ao clicar no botão "Criar tópico" da página principal (Figura 3.6), é aberta uma nova página com uma seção (Figura 3.7) de criação de um novo tópico. O usuário deve selecionar a universidade, o curso e a disciplina referentes ao tópico a ser criado. Após isso, deve informar um título e uma descrição, que pode ser tanto um conteúdo próprio como um *link* para conteúdos externos. O usuário não necessariamente é o criador do conteúdo. Ele pode estar simplesmente compartilhando conteúdos que julgue serem relevantes.

Figura 3.7 – Protótipo visual da funcionalidade “Criar tópico”.

O protótipo visual da funcionalidade “Criar tópico” apresenta um formulário com o seguinte layout:

- Um cabeçalho com o título “Criar tópico” em negrito.
- Quatro campos de entrada de texto com bordas arredondadas e ícones de seta para baixo no canto inferior direito, indicando que são menus suspensos:
  - Universidade
  - Curso de Graduação
  - Disciplina
- Um campo de entrada de texto simples para o “Título”.
- Um campo de entrada de texto maior para a “Descrição”.
- Um botão azul com o texto “Criar” em branco, centralizado na base do formulário.

### 3.3.2.3 Visualizar tópico, Curtir tópico e Cancelar curtida

Ao clicar em um tópico listado na página principal (Figura 3.6), é aberta uma nova página de visualização de tópico contendo maiores detalhes do tópico em questão (Figura 3.8). Ao visualizar um tópico, o usuário pode demonstrar que gostou dele clicando no botão “Curtir”, que então mudará seu texto para “Curtido”. Caso queira cancelar a curtida, basta clicar no botão “Curtido” que o processo será desfeito. Cada tópico possui um total de curtidas, que é atualizado toda vez que usuários curtem o tópico ou cancelam curtidas realizadas ao mesmo.

Figura 3.8 – Protótipo visual de um tópico.

O protótipo visual de um tópico apresenta o seguinte layout:

- Um cabeçalho azul com o perfil do usuário (Matheus Einloft), a data e hora (01/12/20 às 14:01), o curso (Ciência da Computação (UFSM)) e o número de curtidas (10 curtidas).
- Um subtítulo “Laços de Repetição” em negrito.
- Um texto explicativo sobre laços de repetição: “Laços de repetição são um dos conceitos de programação mais básicos e poderosos. Um laço de repetição, ou loop, é uma instrução que se repete até que uma determinada condição seja alcançada. Em uma estrutura de loop, o loop faz uma pergunta. Se a resposta exigir ação, ela será executada. A mesma pergunta é feita repetidamente até que nenhuma ação adicional seja necessária. Cada vez que a pergunta é feita, é chamada de iteração.”
- Um texto introdutório: “Existem diversos tipos de laços de repetição:”
- Uma lista de itens:
  - **for**: é um loop que executa por um número determinado de vezes.
  - **while**: é um loop que é repetido enquanto uma expressão é verdadeira.
  - **do while**: é um loop que executa uma primeira vez, e então é repetido enquanto uma expressão é verdadeira.
- Um botão azul com o ícone de uma mão dando um like e o texto “Curtir” em branco, centralizado na base do tópico.

### 3.3.2.4 Visualizar tópicos recomendados

Na mesma página em que um tópico é visualizado, também pode-se acessar uma listagem de tópicos recomendados pela plataforma ao usuário. Esses tópicos aparecem em uma seção logo abaixo do tópico sendo visualizado (Figura 3.9).

Figura 3.9 – Protótipo visual da funcionalidade “Visualizar tópicos recomendados”.



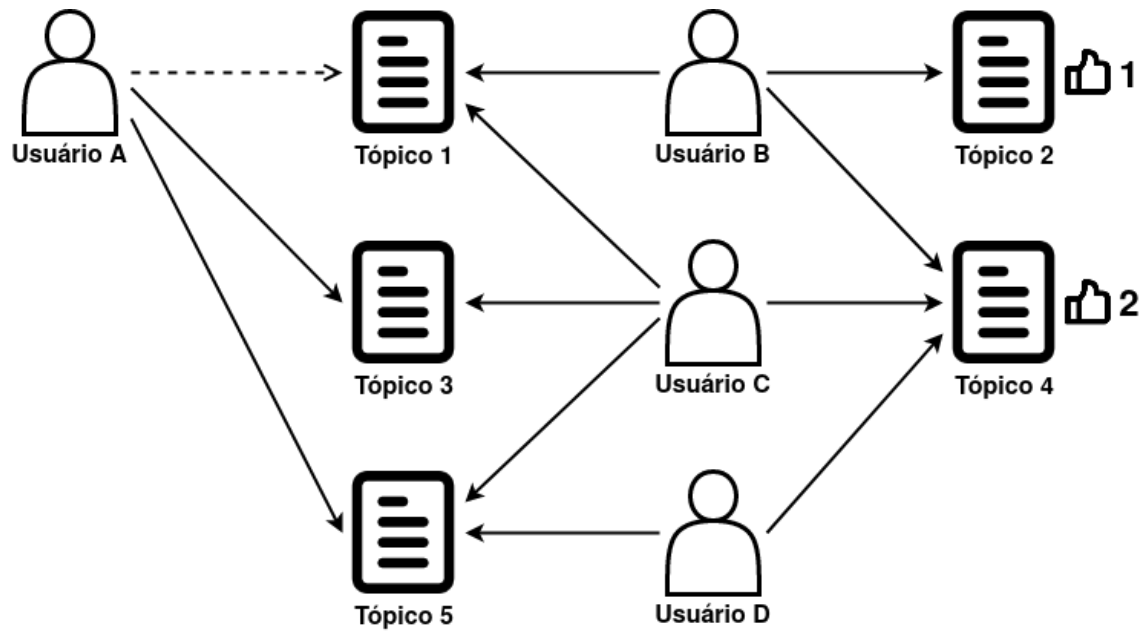
A definição desses tópicos é realizada por um sistema de recomendação baseado em Filtragem Colaborativa. Considerando que  $A$  representa o usuário a ser recomendado novos tópicos e que  $T$  representa o tópico que está sendo visualizado por  $A$ , o processo de recomendação funciona da seguinte forma:

1. observa-se outros usuários que curtiram o tópico  $T$ ;
2. verifica-se quais outros tópicos foram curtidos por esses usuários;
3. obtém-se todos os tópicos que o usuário  $A$  já curtiu, para então compará-los com os tópicos curtidos pelos outros usuários, removendo os iguais;
4. obtém-se o número de curtidas de cada um destes tópicos resultantes (considerando apenas os usuários que curtiram o tópico  $T$ );
5. apresenta-se os tópicos ordenados pelos maiores valores destes totais de curtidas.

Para ilustrar, considere o exemplo demonstrado na Figura 3.10. No exemplo, a linha tracejada demonstra que o usuário  $A$  está visualizando o tópico  $1$ . Já as linhas cheias indicam a relação entre usuário e tópicos que foram curtidos.

Observa-se que os usuários *B* e *C* também curtiram o tópico *1*; Logo, as curtidas destes dois usuários serão avaliadas. A figura mostra que *B* e *C* curtiram os tópicos *2*, *3*, *4* e *5*. Como o usuário *A* também curtiu os tópicos *3* e *5*, eles são desconsiderados. Analisando os demais tópicos (*2* e *4*), percebe-se que o tópico *4* será o primeiro a ser recomendado, pois possui o maior número de curtidas, dentre os usuários selecionados (duas curtidas para o tópico *4* e uma curtida para o tópico *2*).

Figura 3.10 – Exemplo de uma recomendação de tópicos.



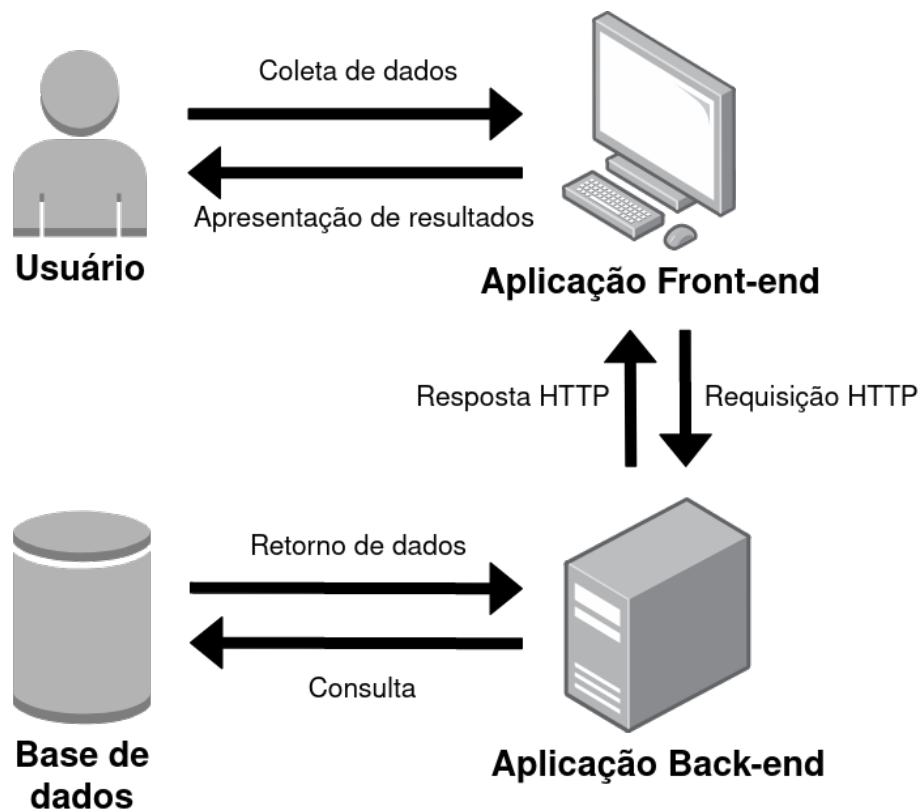
## 4 DESENVOLVIMENTO DO SISTEMA

Este capítulo elenca definições referentes ao desenvolvimento da aplicação do trabalho em questão, e está estruturado da seguinte forma: a Seção 4.1 apresenta uma visão geral da arquitetura da aplicação; a Seção 4.2 explica a respeito da base de dados; a Seção 4.3 demonstra o *back-end* do sistema; a Seção 4.4 aborda o *front-end*; e a Seção 4.5 demonstra como foi implementada a recomendação de tópicos.

### 4.1 VISÃO GERAL DA ARQUITETURA

A arquitetura da plataforma é estruturada conforme demonstra a Figura 4.1: uma aplicação *back-end*, uma aplicação *front-end* e uma base de dados. Por meio de um navegador *web*, o usuário interage com o *front-end*; o *front-end* envia a requisição do usuário ao *back-end* através do protocolo HTTP; o *back-end* processa a requisição, faz consultas na base de dados e envia uma resposta HTTP ao *front-end*; *front-end* recebe os dados de resposta, organiza-os em uma interface gráfica e então apresenta o resultado ao usuário.

Figura 4.1 – Arquitetura da aplicação.

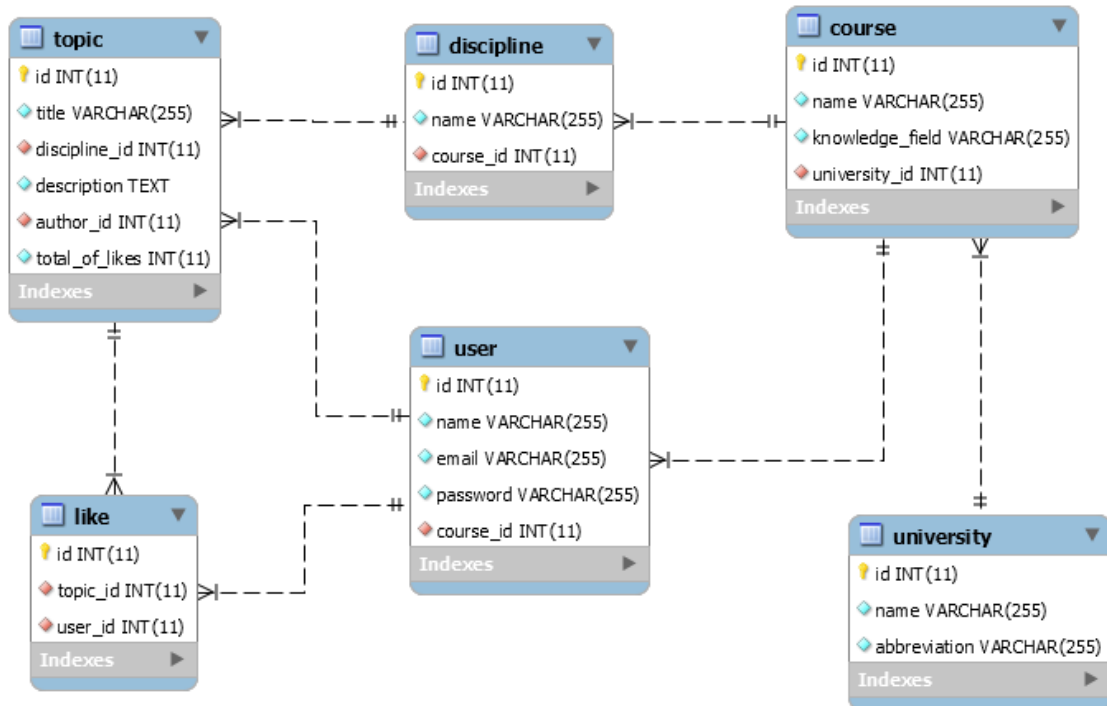




## 4.2 BASE DE DADOS

A base de dados da aplicação foi modelada para atender aos requisitos definidos no capítulo anterior (Capítulo 3). A Figura 4.2 apresenta o Diagrama Entidade-Relacionamento definido para o sistema. Cada entidade do diagrama busca representar uma abstração dos módulos definidos na Seção 3.3.

Figura 4.2 – Diagrama Entidade-Relacionamento.



A descrição geral de cada tabela é destacada a seguir:

- **University**: armazena dados de uma universidade.
- **Course**: armazena dados de um curso. Um curso está vinculado a uma universidade.
- **Discipline**: armazena dados de uma disciplina. Uma disciplina está vinculada a um curso.
- **User**: armazena dados de um usuário. Um usuário está vinculado ao seu curso de origem.
- **Topic**: armazena dados de um tópico, incluindo o total de curtidas que ele recebeu. Um tópico está vinculado a uma disciplina e ao usuário que o criou.
- **Like**: armazena a curtida que um usuário deu a um tópico.

### 4.2.1 Tecnologias utilizadas

Para este projeto, foi necessário tomar uma decisão quanto ao paradigma de banco de dados a ser usado. Duas abordagens principais foram analisadas: os **Bancos Relacionais** e os **Bancos Orientados a Documentos**.

Por um lado, os bancos orientados a documentos são adequados para registros longos, como é o caso dos tópicos. Por outro lado, o sistema de recomendação exige que sejam feitos cruzamentos e agrupamentos (como será visto mais adiante), e os bancos de dados relacionais são conhecidamente eficientes para esse tipo de consulta.

Desse modo, optou-se pelo SGBD PostgreSQL<sup>1</sup>, um banco de dados relacional desenvolvido em código aberto, de alta performance, de fácil administração e utilização em projetos (AMARAL; LIZARDO; SOUZA, 2011).

As consultas à base de dados foram realizadas por meio do TypeORM<sup>2</sup>, um ORM que permite ao desenvolvedor criar os modelos de dados como se fossem classes orientadas à objetos, sem a necessidade de escrever consultas SQL. E mesmo assim, caso o desenvolvedor necessite utilizar uma consulta SQL, o próprio ORM permite isso.

Utilizou-se o Docker<sup>3</sup>, plataforma aberta de criação, execução e publicação de *containers*, para a execução da instância da base de dados. O PostgreSQL foi empacotado em um *container* através de uma imagem fornecida pelo próprio Docker. Dessa forma, com apenas um comando foi possível instalar todas as dependências necessárias para a execução de uma instância do PostgreSQL.

## 4.3 BACK-END

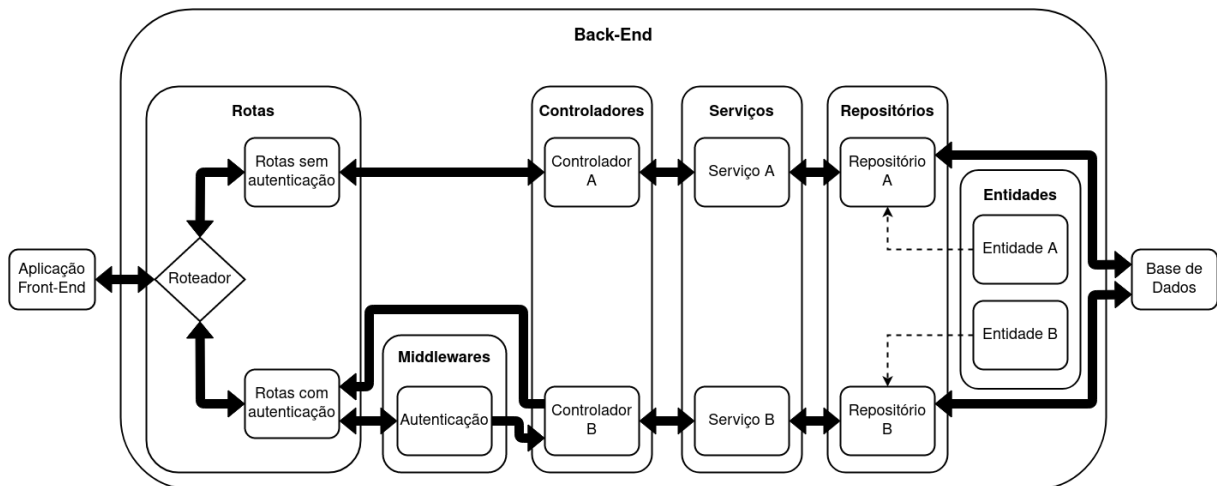
O *back-end* da aplicação consiste em uma API responsável por processar os dados dos usuários coletados pelo *front-end*, aplicar as regras de negócio da plataforma, comunicar-se com a base de dados e retornar respostas ao *front-end*, que então apresentará os resultados ao usuário. Os recursos da API são fornecidos através de rotas seguindo o protocolo HTTP e utilizando os padrões definidos pela arquitetura REST, como a padronização do uso do JSON para comunicação da API com seus clientes. A Figura 4.3 demonstra a arquitetura da aplicação *back-end*.

---

<sup>1</sup><https://www.postgresql.org/>

<sup>2</sup><https://typeorm.io>

<sup>3</sup><https://www.docker.com/>

Figura 4.3 – Arquitetura do *back-end*.

Após o *front-end* realizar uma requisição HTTP ao *back-end*, ela será enviada para o **Roteador**, que é responsável por analisar se a requisição está definida na aplicação e, caso esteja, repassá-la à sua rota específica. O Quadro 4.1 apresenta as rotas desenvolvidas para a API, juntamente com a descrição de cada uma.

Quadro 4.1 – Rotas da API e suas descrições.

Rota	Método	Descrição
:host/users	POST	Criação de um usuário
:host/sessions	POST	Criação de uma sessão no sistema
:host/topics	POST	Criação de um tópico
:host/topics	GET	Listagem de tópicos
:host/topics/:id	GET	Exibição de um tópico
:host/likes	POST	Criação de uma curtida
:host/likes/topic/:topic_id	DELETE	Remoção de uma curtida
:host/likes/topic/:topic_id	GET	Exibição de uma curtida
:host/recommended-topics/:topic_id	GET	Listagem de tópicos recomendados

Existem 2 categorias de Rotas: **Rotas sem autenticação**, onde qualquer cliente pode acessar os seus recursos; e **Rotas com autenticação**, cujos recursos só podem ser consumidos caso a requisição envie um *token* de autenticação, que é validado por uma lógica de autenticação através de um **Middleware**. Esse processo foi implementado para impedir o acesso a recursos privados da aplicação por usuários indevidos.

A rota então envia a requisição ao seu **Controlador**. O controlador é responsável por verificar se a requisição possui todas as informações necessárias para prosseguir. Por exemplo, caso esteja faltando o campo “*e-mail*” no corpo da requisição ao recurso que realiza o *login* de um usuário na aplicação, o controlador deve retornar uma resposta ao cliente, informando que o campo específico não foi informado. Caso contrário, o controlador encaminha a requisição ao **Serviço** respectivo. É nessa camada que estão as regras

de negócio do *software*, ou seja, toda a lógica ligada ao funcionamento das normas da aplicação.

Toda vez que o serviço precisa fazer uma criação, leitura, atualização ou remoção de registros da base de dados, essas operações são feitas pelos **Repositórios**. Com o auxílio de um ORM, eles fazem uso das **Entidades**, classes orientadas a objeto, para realizarem consultas à tabela desejada na base de dados. Após receber o resultado destas consultas e finalizar seus processos, o serviço retorna uma resposta ao controlador, que é responsável por receber os dados desta resposta e retorná-los ao *front-end* através de um objeto no formato JSON.

### 4.3.1 Tecnologias utilizadas

Para o desenvolvimento do código do *back-end*, foi usado o Node.js<sup>4</sup>, plataforma de código aberto que permite o uso da linguagem JavaScript<sup>5</sup> no lado do servidor. Porém, JavaScript tem diversas limitações, como: não possui os confortos da programação baseada em classes e orientada a objetos como o C# ou o Visual Basic; não é fortemente tipada; e não possui *interfaces* e *namespaces* (MAHARRY, 2013). Por este motivo, optou-se por utilizar o TypeScript<sup>6</sup>, linguagem de programação baseada em JavaScript com recursos extras, como a tipagem estática.

Com a finalidade de estruturar a API, organizar o roteamento e processamento de requisições e respostas HTTP, utilizou-se o Express.js<sup>7</sup>, *framework web* para Node.js. Juntamente, foram utilizadas as seguintes bibliotecas: cors<sup>8</sup> para habilitar o CORS e então permitir que as rotas da API possam ser acessadas pelo *front-end*; celebrate<sup>9</sup> para validar os parâmetros e o corpo das requisições HTTP; e class-transformer<sup>10</sup> para transformar objetos que serão utilizados nas respostas HTTP.

As senhas criadas pelos usuários foram criptografadas através do método bcrypt (PROVOS; MAZIERES, 1999), com auxílio da biblioteca bcrypt.js<sup>11</sup>. Já para a autenticação dos usuários, fez-se uso de JWT (JONES; BRADLEY; SAKIMURA, 2015) por meio da biblioteca jsonwebtoken<sup>12</sup>.

---

<sup>4</sup><https://nodejs.org>

<sup>5</sup><https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

<sup>6</sup><https://www.typescriptlang.org/>

<sup>7</sup><https://expressjs.com>

<sup>8</sup><https://github.com/expressjs/cors>

<sup>9</sup><https://github.com/arb/celebrate>

<sup>10</sup><https://github.com/typestack/class-transformer>

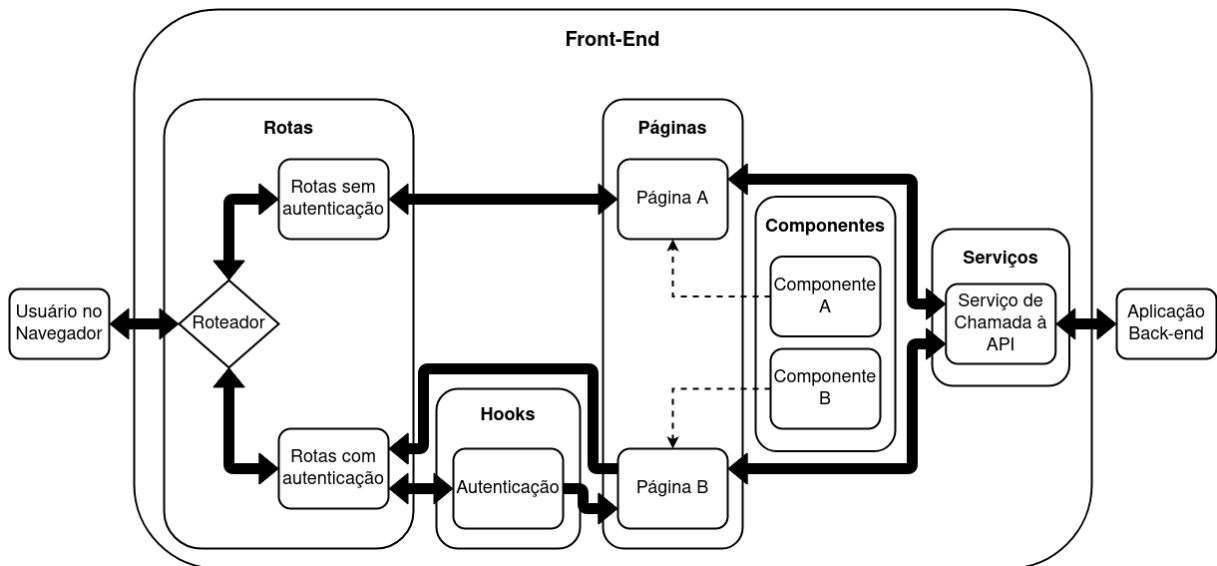
<sup>11</sup><https://github.com/dcodeIO/bcrypt.js>

<sup>12</sup><https://github.com/auth0/node-jwt-token>

#### 4.4 FRONT-END

O *front-end* consiste em uma aplicação *web* cujo acesso ocorre por meio de navegadores *web*. É a camada do sistema em que ocorre a interação direta com os usuários. É responsável por: receber dados dos usuários e processá-los; comunicar-se com o *back-end* quando necessário; organizar visualmente os resultados; e apresentá-los na interface do navegador. A Figura 4.4 demonstra a arquitetura da aplicação *front-end*.

Figura 4.4 – Arquitetura do *front-end*.



Quando o usuário tenta acessar uma página do sistema, o *front-end* recebe essa requisição pelo **Roteador**, que é responsável por analisar se a rota que o usuário deseja acessar está definida na aplicação e, caso esteja, repassá-la à sua rota específica. Existem 2 categorias de Rotas: **Rotas sem autenticação**, onde qualquer cliente pode acessar os seus recursos; e **Rotas com autenticação**, cujos recursos só podem ser consumidos caso o usuário esteja logado na aplicação (que é determinado através de um **Hook** de autenticação, o qual verifica se existe um *token* de autenticação válido no armazenamento local do navegador). Esse processo foi implementado para impedir o acesso a recursos privados da aplicação por usuários indevidos.

Finalizando o processo de roteamento, o usuário é enviado à **Página** que solicitou para acessar. Cada página tem sua estrutura própria, sendo alterada em partes que tem-se variáveis definidas. As páginas fazem de uso de **Componentes**, conjuntos isolados de código que podem ser reaproveitados em diversas páginas. Quando a página necessita de alguma informação do *back-end*, é executado um **Serviço** que fará uma requisição HTTP à API.

#### 4.4.1 Tecnologias utilizadas

Para o desenvolvimento do projeto *front-end*, foi usado o React<sup>13</sup>, biblioteca da linguagem JavaScript para construção de interfaces de usuário. Como ela utiliza como base tecnologias básicas de desenvolvimento *web*, como HTML5<sup>14</sup>, CSS3<sup>15</sup> e JavaScript, estas também são utilizadas. A única exceção é o JavaScript que, pelos mesmos motivos pontuados na Subseção 4.3.1 e pela possibilidade de reaproveitamento do conhecimento utilizado no *back-end*, optou-se por utilizar o TypeScript em seu lugar.

A comunicação de informações do *front-end* com o *back-end* ocorre através de chamadas HTTP. Essa comunicação foi implementada utilizando o axios<sup>16</sup>, cliente HTTP baseado em *Promises* que pode ser utilizado tanto no lado do cliente (navegador por exemplo) como no servidor (Node.js por exemplo). Para validação dos parâmetros e do corpo das requisições HTTP a serem enviadas ao *back-end* foi utilizado o yup<sup>17</sup>.

Demais aspectos do projeto foram desenvolvidos com auxílio das seguintes bibliotecas: react-router-dom<sup>18</sup> para gerenciamento das rotas do navegador; styled-components<sup>19</sup> para estilização de componentes React e páginas da aplicação; uniform<sup>20</sup> para criação de formulários; react-select<sup>21</sup> para criação de botões *dropdown*; date-fns<sup>22</sup> para formatação de datas; e react-icons<sup>23</sup> para adição de ícones no formato SVG<sup>24</sup>.

#### 4.5 IMPLEMENTAÇÃO DA RECOMENDAÇÃO DE TÓPICOS

Conforme já demonstrado no capítulo anterior (Subseção 3.3.2.4), a recomendação de tópicos foi desenvolvida por meio de abordagem de filtragem colaborativa.

Dois parâmetros são importantes durante a filtragem colaborativa:

- **:current\_user**: identifica o usuário principal que está em busca de recomendações.
- **:current\_topic**: identifica o tópico que o usuário principal está acessando.

A recomendação é realizada por meio de uma consulta SQL, que retorna os tópicos a serem recomendados já na ordem esperada. A consulta é composta por sub-

<sup>13</sup><https://reactjs.org/>

<sup>14</sup><https://developer.mozilla.org/pt-BR/docs/Web/HTML>

<sup>15</sup><https://developer.mozilla.org/pt-BR/docs/Web/CSS>

<sup>16</sup><https://github.com/axios/axios>

<sup>17</sup><https://github.com/jquense/yup>

<sup>18</sup><https://github.com/ReactTraining/react-router>

<sup>19</sup><https://github.com/styled-components/styled-components>

<sup>20</sup><https://github.com/uniform/uniform>

<sup>21</sup><https://github.com/JedWatson/react-select>

<sup>22</sup><https://github.com/date-fns/date-fns>

<sup>23</sup><https://github.com/react-icons/react-icons>

<sup>24</sup><https://developer.mozilla.org/pt-BR/docs/Web/SVG>

consultas. Para compreendê-la, cada sub-consulta é apresentada individualmente, para então se apresentar a consulta completa.

A Figura 4.5 apresenta a primeira sub-consulta. Esta consulta seleciona usuários que curtiram o tópico sendo acessado pelo usuário principal. É a partir destes usuários selecionados que será realizada a recomendação.

Figura 4.5 – Primeira sub-consulta.

```
1 SELECT user_id
2 FROM Like
3 WHERE topic_id = :current_topic
```

A Figura 4.6 apresenta a segunda sub-consulta. Esta consulta retorna os tópicos curtidos pelo usuário principal. Estes tópicos devem ser descartados, pois eles já foram acessados.

Figura 4.6 – Segunda sub-consulta.

```
1 SELECT topic_id
2 FROM Like
3 WHERE user_id = :current_user
```

Por fim, a Figura 4.7 demonstra a consulta SQL completa. O objetivo é selecionar as curtidas em tópicos curtidos pelos usuários selecionados (linha 3), desde que os tópicos já não tenham sido curtidos pelo usuário principal (linha 8). Uma última filtragem exclui as curtidas referentes ao tópico sendo atualmente acessado pelo usuário principal (linha 13). As curtidas resultantes são então agrupadas por tópico (linha 14), para se obter a contagem de curtidas por tópico. Os resultados são exibidos por ordem de contagem, de modo a exibir os tópicos mais curtidos primeiro (linha 15).

Figura 4.7 – Consulta SQL - Recomendação de tópicos.

```

1 SELECT topic_id, COUNT(*) as total_of_related_likes
2 FROM Like
3 WHERE user_id IN (
4   SELECT user_id
5   FROM Like
6   WHERE topic_id = :current_topic
7 )
8 AND topic_id NOT IN (
9   SELECT topic_id
10  FROM Like
11  WHERE user_id = :current_user
12 )
13 AND topic_id != :current_topic
14 GROUP BY topic_id
15 ORDER BY total_of_related_likes DESC

```

#### 4.5.1 Demonstração da recomendação de tópicos

Esta subseção traz uma demonstração da recomendação de tópicos através de um exemplo prático no sistema. As telas implementadas são baseadas nos *wireframes* demonstrados na Seção 3.3, tendo pequenas diferenças entre si, o que é comum devido à tecnologia de formatação utilizada.

Para exemplificar, será considerado o mesmo exemplo usado na Figura 3.10, que descreve as curtidas de cada usuário. O Quadro 4.2 exhibe essas curtidas.

Quadro 4.2 – Curtidas dos usuários no exemplo da Figura 3.10

Usuário	Tópicos curtidos
Usuário A	Tópico 3, Tópico 5
Usuário B	Tópico 1, Tópico 2, Tópico 4
Usuário C	Tópico 1, Tópico 3, Tópico 4, Tópico 5
Usuário D	Tópico 4, Tópico 5

A fim de possibilitar o uso demonstrativo, foram criados diretamente no banco de dados da aplicação: uma universidade, um curso e uma disciplina. Já os usuários, tópicos e curtidas apresentados no Quadro 4.2 foram criados manualmente através da aplicação.

Todos os tópicos foram criados por um usuário chamado “Admin”. A Figura 4.8 ilustra a criação do tópico 1 no sistema. Os itens referentes aos cursos e disciplinas são ajustados quando o usuário escolhe a universidade e o curso, respectivamente.



Figura 4.8 – Página de criação de tópico.

←

→

### Criar tópico

Tópico 1

Universidade A (UA)

Curso A

Disciplina A

Descrição do tópico 1

Criar

Após a criação do tópico, o sistema foi acessado a partir do usuário A. Após o *login*, o usuário é levado à página principal da aplicação, conforme demonstrado na figura abaixo (Figura 4.9).

Figura 4.9 – Página principal do sistema.

→

### Página Principal

Criar tópico

Admin • 16/08/2021 às 20:57

Curso A (UA)

4

**Tópico 4**

Descrição do tópico 4

Admin • 16/08/2021 às 20:57

Curso A (UA)

3

**Tópico 5**

Descrição do tópico 5

Admin • 16/08/2021 às 20:57

Curso A (UA)

2

**Tópico 3**

Descrição do tópico 3

Admin • 16/08/2021 às 20:56

Curso A (UA)

2

**Tópico 1**

Descrição do tópico 1

Após visualizar a lista de tópicos na página principal, o usuário clica no tópico 1 para visualizá-lo. Então, ele é redirecionado à página de visualização do tópico 1 (Figura 4.10).

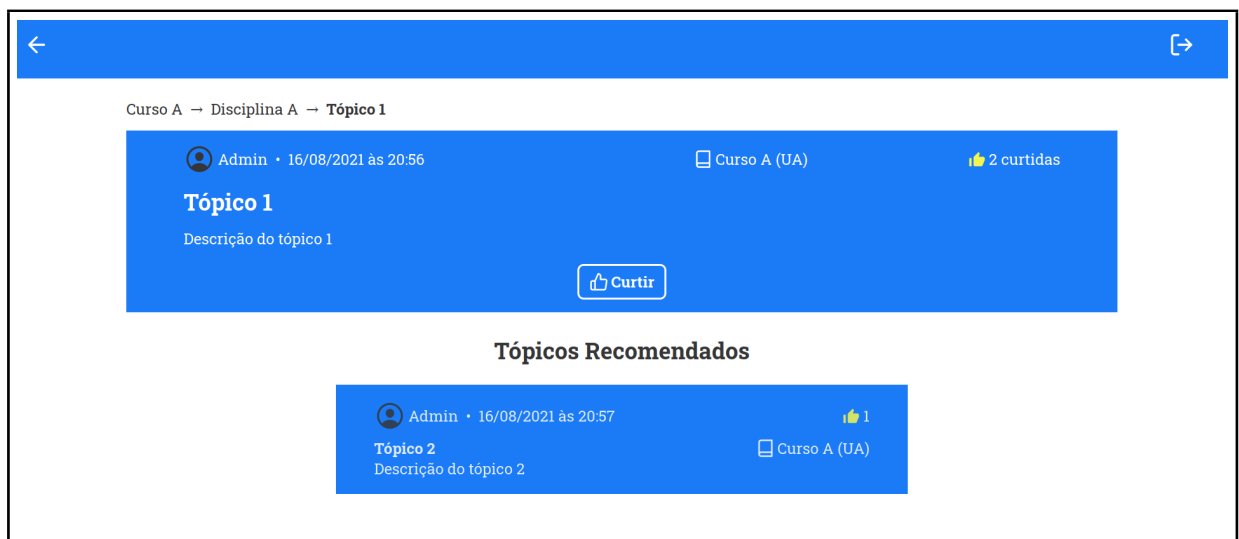
Como esperado, além das informações do tópico 1, a tela apresenta os tópicos 4 e 2 na seção “Tópicos recomendados”. É possível observar que a ordem de exibição dos tópicos é consistente com o critério de ordenação escolhido, priorizando tópicos que tenham recebido mais curtidas pelos usuários que também acessaram o tópico 1.

Figura 4.10 – Página do Tópico 1.



Para demonstrar a lista de tópicos sendo atualizada, considera-se que neste momento o usuário A acessou o tópico 4 e clicou no botão “Curtir”, ou seja, curtiu o tópico 4. Assim, ao acessar novamente o tópico 1, pode-se perceber que o tópico 4 não está mais presente na seção “Tópicos Recomendados” (conforme demonstrado na figura 4.11).

Figura 4.11 – Página do Tópico 1 (considerando que o usuário A curtiu o tópico 4).



## 5 CONCLUSÃO

O presente trabalho teve como objetivo a implementação de uma plataforma *web* de compartilhamento de conteúdos de disciplinas de graduação, chamados de tópicos, juntamente com a aplicação de uma recomendação desses tópicos aos estudantes. Esta recomendação foi desenvolvida através do método de filtragem colaborativa, utilizando o “gosto” de outros usuários (representados através de curtidas) para determinar outros tópicos a serem recomendados.

As principais funcionalidades levantadas foram implementadas, permitindo ao usuário utilizar a aplicação conforme esperado. Através das funcionalidades ‘criar conta’ e “fazer *login*”, é possível acessar a plataforma. Na página principal, o estudante consegue criar um tópico, visualizar a lista de tópicos criados e visualizar um tópico em específico, sendo redirecionado à página do tópico em questão. Conforme visualiza esta página, o usuário pode curtir o tópico para demonstrar seu interesse nele e visualizar tópicos recomendados. Ele pode também acessar estes tópicos e seguir navegando pelo sistema.

Para trabalhos futuros, destaca-se a possibilidade de realizar alterações no sistema de recomendação, aprimorando a definição dos tópicos a serem recomendados e, consequentemente, melhorando a experiência do estudante ao utilizar a plataforma. Devido à presença dos problemas do novo item e da escassez de avaliações, sugere-se o uso de uma abordagem híbrida para a recomendação. Além disso, propõe-se outras possíveis alterações: recomendação de tópicos da mesma disciplina; recomendação de tópicos que possuem palavras-chave iguais ou relacionadas; e recomendação de tópicos pertencentes à mesma categoria (conjunto de tópicos com uma característica em comum).

Além disso, destaca-se a adição de outras funcionalidades que acabaram não entrando no escopo do sistema desenvolvido, mas que possuem potencial de aprimorá-lo: funcionalidade de adicionar comentários em tópicos, possibilitando diálogos e uma maior interação entre estudantes; a possibilidade de usuários reportarem outros usuários ou tópicos por julgar serem inadequados (por exemplo, caso um usuário tenha desrespeitado outro através da plataforma); um novo tipo de usuário responsável por administrar a plataforma, analisando se conteúdos e usuários estão de acordo com as normas da plataforma e excluindo-os caso necessário; e busca de tópicos por texto informado pelo usuário.

Outra oportunidade de melhoria para a aplicação é no formato dos conteúdos dos tópicos. O sistema, da forma que está desenvolvido atualmente, não permite que seja escrito hipertexto, ou seja, um texto com formatação e imagens. Caso isto fosse desenvolvido, poderia permitir por exemplo a adição de *links*, imagens e vídeos aos tópicos da plataforma.

De modo geral, a aplicação foi idealizada, organizada e implementada buscando contribuir com os estudos de alunos universitários. O resultado é um ambiente de compar-

tilhamento de conteúdos estruturado em universidades, cursos e disciplinas, melhorando a organização dos tópicos na plataforma. Esta organização, juntamente com a recomendação de tópicos, facilitam aos estudantes a busca por conteúdos relevantes. Devido a tudo isso, acredita-se que o sistema tem potencial para ser uma ferramenta de grande auxílio aos estudantes universitários.

## REFERÊNCIAS BIBLIOGRÁFICAS

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. **IEEE transactions on knowledge and data engineering**, IEEE, v. 17, n. 6, p. 734–749, 2005.

AMARAL, H. R.; LIZARDO, L. E. O.; SOUZA, A. C. V. de. Postgresql: uma alternativa para sistemas gerenciadores de banco de dados de código aberto. In: **Anais do Congresso Nacional Universidade, EAD e Software Livre**. [S.l.: s.n.], 2011. v. 2, n. 2.

BURKE, R. Hybrid recommender systems: Survey and experiments. **User modeling and user-adapted interaction**, Springer, v. 12, n. 4, p. 331–370, 2002.

\_\_\_\_\_. Hybrid web recommender systems. **The adaptive web**, Springer, p. 377–408, 2007.

CALLE-ALONSO, F. et al. Neurok: A collaborative e-learning platform based on pedagogical principles from neuroscience. In: **CSEDU (1)**. [S.l.: s.n.], 2017. p. 550–555.

COSTA, E.; AGUIAR, J.; MAGALHÃES, J. Sistemas de recomendação de recursos educacionais: conceitos, técnicas e aplicações. **Jornada de Atualização em Informática na Educação**, v. 1, n. 1, 2013.

FOWLER, M. **UML Distilled: A Brief Guide to the Standard Object Modeling Language**. 3. ed. USA: Addison-Wesley Longman Publishing Co., Inc., 2003. ISBN 0321193687.

GOMEZ-URIBE, C. A.; HUNT, N. The netflix recommender system: Algorithms, business value, and innovation. **ACM Transactions on Management Information Systems (TMIS)**, ACM New York, NY, USA, v. 6, n. 4, p. 1–19, 2015.

JONES, M.; BRADLEY, J.; SAKIMURA, N. RFC, **JSON Web Token (JWT)**. Fremont, CA, USA: RFC Editor, 2015. RFC 7519 (Proposed Standard). (Internet Request for Comments, 7519). Updated by RFCs 7797, 8725. Disponível em: <<https://www.rfc-editor.org/rfc/rfc7519.txt>>.

LICHTNOW, D. et al. O uso de técnicas de recomendação em um sistema para apoio à aprendizagem colaborativa. **Revista brasileira de informática na educação**, v. 14, n. 3, 2006.

LINDEN, G.; SMITH, B.; YORK, J. Amazon. com recommendations: Item-to-item collaborative filtering. **IEEE Internet computing**, IEEE, v. 7, n. 1, p. 76–80, 2003.

MACIEL, C. Educação a distância: ambientes virtuais de aprendizagem. Universidade Federal de Mato Grosso, 2018.

MAHARRY, D. **TypeScript revealed**. [S.l.]: Apress, 2013.

PROVOS, N.; MAZIERES, D. Bcrypt algorithm. In: **USENIX**. [S.l.: s.n.], 1999.

SOMMERVILLE, I. Software engineering 9th edition. **ISBN-10**, v. 137035152, p. 18, 2011.

TAVEIRA-GOMES, T. et al. A novel collaborative e-learning platform for medical students-alert student. **BMC medical education**, Springer, v. 14, n. 1, p. 1–14, 2014.

TORRADO, F. R. O. **Collaborative Study Web Platform**. 2016. Tese (Doutorado) — Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa, 2016.