UNIVERSIDADE ESTADUAL PAULISTA “Júlio de Mesquita Filho”
Campus - Bauru
Faculdade de Ciências
Bacharelado em Sistemas de Informação

Giovana Carolini Reinaldo da Silva

WHAT TO WATCH: SISTEMA WEB DE RECOMENDAÇÃO DE FILMES E SÉRIES

BAURU
2022

Giovana Carolini Reinaldo da Silva

WHAT TO WATCH: SISTEMA WEB DE RECOMENDAÇÃO DE FILMES E SÉRIES

Trabalho de Conclusão de Curso do Curso de Bacharelado em Sistemas de Informação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, Campus Bauru.

Orientadora: Profª. Drª. Simone das Graças Domingues Prado

BAURU
2022

S586w

Silva, Giovana Carolini Reinaldo da
What to watch : sistema web de recomendação de filmes e séries / Giovana Carolini Reinaldo da Silva. -- Bauru, 2023
83 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Sistemas de Informação) - Universidade Estadual Paulista (Unesp), Faculdade de Ciências, Bauru
Orientadora: Simone das Graças Domingues Prado

1. Algoritmos de computador. 2. Matemática. 3. Técnica. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Ciências, Bauru. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Giovana Carolini Reinaldo da Silva

WHAT TO WATCH: SISTEMA WEB DE RECOMENDAÇÃO DE FILMES E SÉRIES

Trabalho de Conclusão de Curso do Curso de Bacharelado em Sistemas de Informação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, Campus Bauru.

Orientadora: Profª. Drª. Simone das Graças Domingues Prado

BANCA EXAMINADORA

Prof.ª Dr.ª Simone das Graças Domingues Prado
Orientadora
Departamento de Computação
Faculdade de Ciências
Unesp - Campus Bauru

Prof. Dr. José Remo Ferreira Brega
Departamento de Computação
Faculdade de Ciências
Unesp - Campus Bauru

Prof. Dr. Kelton Augusto Pontara da Costa
Departamento de Computação
Faculdade de Ciências
Unesp - Campus Bauru

AGRADECIMENTOS

Agradeço inicialmente a minha mãe, por me apoiar sempre que necessitei. Sempre com empenho máximo em me ajudar o máximo que pode.

Agradeço também a minha melhor amiga que sempre me ajudou com ideias e incentivos não só nesse trabalho como durante todos esses anos de amizade.

Aos meus professores e colegas da Unesp, por todo conhecimento compartilhado e, especialmente, à minha prezada e querida orientadora Prof.ª Dr.ª Simone das Graças Domingues Prado, pela dedicação, compreensão e paciência.

RESUMO

A tecnologia mostra-se mais uma vez presente na vida das pessoas. Uma das atividades que exerce é o lazer, agora principalmente por meio dela. Antes veio a TV aberta, depois surgiu a TV por assinatura, agora tem-se consolidado no cotidiano os serviços de *streaming* de vídeo. Com a era de empresas interessadas em lançar suas próprias plataformas, o conteúdo que se encontrava nos poucos serviços foi gradativamente fragmentado. Dessa forma, o trabalho reduz essa experiência de fragmentação ao disponibilizar todo catálogo de entretenimento brasileiro numa única plataforma com um sistema de recomendação de modo a ajudar as pessoas a encontrarem com facilidade o que assistir sem precisar entrar em todos os serviços assinados.

Palavras-chave: *Streaming* de vídeo. Sistema de Recomendação. Filmes. Séries.

ABSTRACT

Technology shows itself once again present in people's lives. One of the activities it performs is leisure, now mainly through it. Before came the open TV, then came the cable TV, now it has been consolidated in everyday life the video streaming services. With the era of companies interested in launching their own platforms, the content that could be found in the few services has been gradually fragmented. Thus, the work reduces this fragmentation experience by making the entire Brazilian entertainment catalog available on a single platform with a recommendation system in order to help people easily find what to watch without having to enter all the subscribed services.

Keywords: Video Streaming. Recommendation System. Movies. TV Show.

LISTA DE FIGURAS

Figura 1 - Modelo Cliente-Servidor 18
Figura 2 - Modelo de dados relacional 19
Figura 3 - Modelo de dados não relacional 20
Figura 4 - Abordagem tradicional de se resolver problemas computacionais 21
Figura 5 - Abordagem de aprendizado de máquina de se resolver problemas 22
Figura 6 - Classificação dos sistemas de recomendação 25
Figura 7 - Exemplo de fatoração de matriz das avaliações de filmes 31
Figura 8 - Exemplo de algoritmo ALS 34
Figura 9 - Representação da estrutura do projeto 37
Figura 10 - Logo do projeto TMDB 38
Figura 11 - Exemplo de um documento HTML 39
Figura 12 - Exemplo de uma folha de estilo CSS 40
Figura 13 - Exemplo de script Javascript 41
Figura 14 - Exemplo de código Tailwind CSS 41
Figura 15 - Exemplo de código Vue.js 42
Figura 16 - Exemplo de código Node.js 42
Figura 17 - Exemplo de código Java 43
Figura 18 - Exemplo de código com uso do *framework* Spring Boot 44
Figura 19 - Exemplo de código em Python 45
Figura 20 - Exemplo de código com pandas 46
Figura 21 - Exemplo de código com scikit-learn 46
Figura 22 - Exemplo de documento MongoDB 47
Figura 23 - Diagrama de Contexto 48
Figura 24 - Diagrama de Fluxo de Dados Nível 1 49
Figura 25 - Diagrama de Fluxo de Dados Nível 2 49
Figura 26 - Página inicial 51
Figura 27 - Página de *login*/entrar na conta 51
Figura 28 - Página de cadastro 52
Figura 29 - Página de filmes 52
Figura 30 - Visualização do filme pelo usuário visitante 54
Figura 31 - Visualização do filme pelo usuário conectado em sua conta 55
Figura 32 - Página de séries 56

Figura 33 - Visualização da série pelo usuário visitante 57
Figura 34 - Visualização da série pelo usuário conectado em sua conta 58
Figura 35 - Funcionalidade de pesquisa e seus filtros na página de filmes 59
Figura 36 - Funcionalidade de pesquisa na barra de navegação 59
Figura 37 - Recomendações personalizadas para o usuário conectado 61
Figura 38 - Visualização das listas 62
Figura 39 - Modo visualização do perfil do usuário 63
Figura 40 - Fluxograma do algoritmo de recomendação baseada no conteúdo 64
Figura 41 - Fluxograma do algoritmo de recomendação colaborativa 65

LISTA DE TABELAS

Tabela 1 - Avaliação dos filmes agrupados por gêneros 69
Tabela 2 - Avaliação das séries agrupadas por gêneros 70
Tabela 3 - Especificação técnica da máquina pessoal 72

LISTA DE QUADROS

Quadro 1 - Serviços de *streaming* disponíveis para filtragem de filmes 59
Quadro 2 - Serviços de *streaming* disponíveis para filtragem de séries 60
Quadro 3 - Gêneros disponíveis para filtragem de filmes 60
Quadro 4 - Gêneros disponíveis para filtragem de séries 60
Quadro 5 - Classificação de idade disponível para filtragem de filmes e séries 60

LISTA DE ABREVIATURAS E SIGLAS

ALS	*Alternating Least Square*
AWS	*Amazon Web Services*
API	*Application Programming Interface*
BD	Banco de Dados
CSS	*Cascading Style Sheets*
DFD	Diagrama de Fluxo de Dados
GCP	*Google Cloud Platform*
HTML	*HyperText Markup Language*
IA	Inteligência Artificial
JSON	*Javascript Object Notation*
ML	*Machine Learning*
NLP	*Natural Language Processing*
RAM	*Random Access Memory*
RC	Recomendação Colaborativa
RMSE	*Root Mean Squared Error*
SVD	Decomposição em valores singulares
TCC	Trabalho de Conclusão de Curso
TF-IDF	*Term Frequency-Inverse Document Frequency*
TMDB	*The Movie Database*
WWW	*World Wide Web*

SUMÁRIO

1. **INTRODUÇÃO** 12

1.1 OBJETIVOS 13

1.1.2 **Objetivo geral** 13

1.1.3 **Objetivos específicos** 13

1.2 ORGANIZAÇÃO DA MONOGRAFIA 13

2. **REFERENCIAL TEÓRICO** 15

2.1 REDES SOCIAIS 15

2.2 SERVIÇOS DE STREAMING DE VÍDEO 16

2.3 APLICAÇÕES WEB 17

2.3.1 **API** 17

2.3.2 **Front-end** 18

2.3.3 **Back-end** 18

2.4 BANCO DE DADOS 19

2.4.1 **Banco de dados relacional** 19

2.4.2 **Banco de dados não-relacional** 19

2.5 COMPUTAÇÃO EM NUVEM 20

2.6 APRENDIZADO DE MÁQUINA 20

2.7 PROCESSAMENTO DE DADOS 23

2.7.1 **Em lote** 23

2.7.2 **Em tempo real** 23

2.8 SISTEMAS DE RECOMENDAÇÃO 24

2.9 CLASSIFICAÇÃO DOS SISTEMAS DE RECOMENDAÇÃO 24

2.9.1 **Recomendação baseada em conteúdo** 25

2.9.1.1 Somente analisa a descrição do conteúdo 25

2.9.1.2 Construção do perfil do usuário e do item dado sua avaliação 26

2.9.2 **Recomendação colaborativa** 26

2.9.2.1 Recomendação colaborativa baseada em memória 27

2.9.2.1.1 *Recomendação baseado em itens* 27

2.9.2.1.2 *Recomendação baseado no usuário* 28
2.9.2.2 Recomendação colaborativa baseada em modelo 28
2.9.3 **Recomendação não personalizada** 29
2.10 ALGORITMOS DE SISTEMAS DE RECOMENDAÇÃO 29
2.10.1 **Vetorização de contagem** 29
2.10.2 **TF-IDF** 30
2.10.3 **Fatoração de matriz** 31
2.10.4 **Algoritmo de mínimos quadrados alternados** 33
2.11 MÉTRICA DE COMPARAÇÃO 34
2.11.1 **Similaridade de cosseno** 35
2.12 MÉTRICAS DE AVALIAÇÃO 35
2.12.1 **Raiz do erro quadrático médio** 36
2.12.2 **Coeficiente de determinação** 36
3. **METODOLOGIA** 37
3.1 ESTRUTURA DO PROJETO 37
3.2 OBTENÇÃO DOS DADOS 38
3.3 FERRAMENTAS UTILIZADAS 39
3.3.1 **HTML** 39
3.3.2 **CSS** 40
3.3.3 **Javascript** 40
3.3.4 **Tailwind CSS** 41
3.3.5 **Vue.js** 42
3.3.6 **Node.js** 42
3.3.7 **Java** 43
3.3.8 **Spring Boot** 43
3.3.9 **Python** 45
3.3.10 **Pandas** 45
3.3.11 **Scikit-learn** 46
3.3.12 **Apache Spark** 47

3.3.13 **MongoDB** 47

4. **ESPECIFICAÇÃO** 48

4.1 DIAGRAMA DE CONTEXTO 48

4.2 DIAGRAMA DE FLUXO DE DADOS NÍVEL 1 - VISÃO GERAL DETALHADA 48

4.3 DIAGRAMA DE FLUXO DE DADOS NÍVEL 2 - 4 CRIAR LISTAS DE TÍTULOS 49

5. **DESENVOLVIMENTO** 50

5.1 DETALHAMENTO DA APLICAÇÃO 50

5.1.1 **Página inicial** 50

5.1.2 **Cadastro e login** 51

5.1.3 **Filmes** 52

5.1.3.1 Visualização de um filme 53

5.1.4 **Séries** 56

5.1.4.1 Visualização de uma série 56

5.1.5 **Pesquisa** 59

5.1.6 **Recomendações personalizadas** 61

5.1.7 **Listas** 61

5.1.8 **Perfil** 62

5.2 SISTEMA DE RECOMENDAÇÃO 63

5.3 TESTES REALIZADOS 67

5.3.1 **Recomendação baseada em conteúdo** 67

5.3.2 **Recomendação colaborativa** 67

5.4 LIMITAÇÕES DO TRABALHO 71

6. **CONCLUSÃO** 74

REFERÊNCIAS 76

1. INTRODUÇÃO

O entretenimento encontra-se presente na vida das pessoas há muito tempo. É algo tão antigo que precede a criação da tecnologia como é conhecida atualmente. No mundo moderno, faz-se presente seu uso nas seguintes vertentes: através da moda, das compras, música, paladar, ilusão e, principalmente, dos jogos.

Segundo Johnson (2017), novas tecnologias ou formas de entretenimento popular mudaram o mundo de maneira direta: deram origem a indústrias, possibilitaram formas de escapismo inéditas, às vezes até promoveram novos tipos de opressão e danos físicos ao ambiente. Mas também mudaram o mundo de uma maneira mais conceitual. Todas as tecnologias emergentes significativas inevitavelmente entram no universo da linguagem como uma nova metáfora, uma forma de estruturar ou realçar algum aspecto da realidade difícil de compreender antes que a metáfora começasse a circular.

Atualmente, um dos entretenimentos mais populares são os *streamings*. Eles surgiram como evolução do que se conhece por TV a cabo. No caso, é pago uma assinatura mensal para consumir filmes, séries e documentários sob demanda.

De início, o pioneiro no Brasil era a Netflix, mas agora os próprios estúdios de cinema e empresas como Amazon e Apple resolveram entrar na briga ao lançarem as suas próprias plataformas para focar nas produções originais - uma tendência de fragmentação de conteúdo.

Hoje as pessoas se veem no dilema de quais serviços priorizar as assinaturas por conta do custo que o pacote gera. Assim, ficam indecisas com a abundância de conteúdo disponível para assistir.

Com a fragmentação dos serviços de *streaming*, o foco deste trabalho de conclusão de curso (TCC) envolveu desenvolver uma plataforma de recomendação de filmes e séries ao centralizar o catálogo de todos os *streamings* disponíveis no mercado brasileiro. De modo a auxiliar as pessoas a encontrarem rapidamente o que consumir, além de priorizar qual serviço assinar sem recorrer aos serviços de pirataria.

1.1 Objetivos

A seguir, serão apresentados os objetivos do trabalho.

1.1.2 Objetivo geral

Desenvolver uma aplicação *web* com algumas características de uma rede social, com um sistema de recomendação para auxiliar cinéfilos a encontrarem filmes ou séries de interesse com facilidade.

1.1.3 Objetivos específicos

- Definir uma aplicação *web* padrão para exibição de filmes e séries;
- Implementar um sistema de recomendação para ajudar os usuários a encontrarem informações relevantes;
- Permitir que usuários façam comentários do conteúdo assistido;
- Permitir que usuários compartilhem listas e;
- Permitir que usuários disponibilizem um meio de comunicação - redes sociais que possuem.

1.2 Organização da monografia

Esta monografia está organizada em seis capítulos, sendo cinco além deste, da seguinte forma:

- Capítulo 2 - REFERENCIAL TEÓRICO: revisão de literatura sobre temas relevantes para o desenvolvimento do trabalho como: redes sociais, aplicações *web*, sistemas de recomendação, classificação dos sistemas de recomendação e métrica de comparação;
- Capítulo 3 - METODOLOGIA: definição e apresentação das ferramentas utilizadas e das pré-condições realizadas para desenvolver a aplicação;
- Capítulo 4 - ESPECIFICAÇÃO: representação formal da aplicação;
- Capítulo 5 - DESENVOLVIMENTO: apresentação do desenvolvimento do projeto, bem como as dificuldades encontradas, limitações impostas e os resultados obtidos;

- Capítulo 6 - CONCLUSÃO: apresenta uma descrição geral dos resultados e os trabalhos futuros.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta a revisão de literatura sobre os tópicos importantes para o trabalho.

2.1 Redes sociais

De acordo com Rosen (2022), a Internet, desde seu início, foi uma ferramenta desenvolvida para permitir a comunicação digital entre os indivíduos e para auxiliar a transferência eletrônica de informações entre as organizações. A evolução das interações sociais viabilizadas digitalmente acabou levando às plataformas densas de informação que hoje são conhecidas como "redes sociais". O negócio de fornecer e apoiar aplicações de rede social é uma indústria multibilionária impulsionada por algumas das empresas mais rentáveis e poderosas do mundo. Cerca da metade da população total do mundo utiliza as redes sociais, uma proporção que está se expandindo rapidamente. O Facebook, por exemplo, tem conquistado mais de 2,85 bilhões de usuários globais, uma saturação alcançada em apenas 17 anos (ROSEN, 2022).

As redes sociais tornaram-se parte da experiência cotidiana das pessoas. A partir do Facebook, Twitter, e Instagram para Snapchat e TikTok, a possibilidade de opções cresce com cada geração de novos usuários. Entretanto, sua proliferação trouxe uma miríade de preocupações sérias sobre os efeitos negativos que as redes sociais têm sobre nossos padrões de socialização, influência social, disseminação de desinformação e governança.

Há também preocupações crescentes sobre os impactos a longo prazo do uso das redes sociais no bem-estar, incluindo o vício das mesmas, preocupações com a privacidade e uma série de efeitos negativos sobre a saúde mental. Além disso, as redes sociais apresentam a rede social mais abrangente e navegável que já existiu, permitindo que os usuários localizem e interajam uns com os outros e por grupos o mais rápido e global do que qualquer comunicação anterior e tecnologia da informação. Ela pode dar voz àqueles que antes eram silenciados, ao mesmo tempo, em que auxilia na disseminação de informações distorcidas com consequências desastrosas.

2.2 Serviços de *streaming* de vídeo

Partindo de um nível técnico, é possível definir o *streaming* - comumente usado em relação à Internet e aos novos serviços de mídia - como a transmissão e recuperação de conteúdo digital que é armazenado e processado em um servidor remoto. Ao contrário de baixar o conteúdo digital, o mesmo é apenas temporariamente retido no cache, não se mantendo armazenado permanentemente no disco rígido do dispositivo do usuário (SPILKER; COLBJØRNSEN, 2020).

Em relação ao contexto histórico, de início, a internet era um ambiente muito limitado. Seu propósito era compartilhar informações sobre as primeiras guerras internacionais de maneira conectada, estando distante dos atuais serviços oferecidos. Essa conexão usou de cabos de rede que eram ligados somente às universidades e dado a evolução natural, a internet chegou na casa da população.

Outro aspecto que evoluiu quase em conjunto foi a transmissão de vídeos. Antes o processo era oneroso devido à qualidade de conexão de rede, mas depois, serviços como YouTube, popularizaram o acesso aos vídeos para todas as pessoas. Inicialmente, o conteúdo dessa plataforma envolveu em parte pirataria do que já existia da indústria do cinema e, posteriormente, surgiram os criadores de conteúdo que passaram a abordar temas variados e inéditos.

A partir do momento em que a indústria cinematográfica percebeu o interesse dos internautas em suas obras, concebeu-se o surgimento dos primeiros serviços de *streaming*, posto o caso da plataforma Netflix lançada em 2007. Devido ao seu sucesso, a evolução desse percurso foi o despertar do interesse de cada estúdio de entretenimento para lançar seu serviço de *streaming*. Hoje as mais populares no território brasileiro são Netflix, HBO Max, Amazon Prime Video, Telecine Play, Crunchyroll, Apple TV+, Disney+, Star+ e Globoplay.

Com essa fragmentação dos serviços de *streaming* surgiram duas formas de conteúdo: o catálogo original e de outros estúdios de entretenimento. O catálogo original, visa despontar cada serviço como referência no mercado ao oferecer um conteúdo fixo e relativamente diferenciado. Já o de outros estúdios, consiste no empréstimo dos direitos de exibição das obras negociadas. Como resultado, esse comportamento contribuiu para a ampliação dos títulos disponíveis para consumo.

2.3 Aplicações *web*

Rossi et al. (2010) definem a *World Wide Web* (*Web)* e aponta que, a sua participação em todos os aspectos da vida das pessoas. Desde sua criação, há 15 anos, as vidas e o trabalho foram inexoravelmente mudados por ela. Ela influenciou a sociedade dramaticamente de diversas maneiras e amadureceu para se tornar uma plataforma atraente e dominante para a implantação de aplicações comerciais, sociais e sistemas de informação organizacionais. Também se tornou uma plataforma de interface universal de usuário para aplicações comerciais, sistemas de informação, bancos de dados e sistemas legados. Suporta gestão de documentos e fluxo de trabalho, trabalho cooperativo, e compartilhamento de conhecimento distribuído e mídia (foto, áudio e vídeo).

As principais vantagens em adotar a *Web* para o desenvolvimento de produtos de *software* incluem (1) nenhum custo de instalação, (2) atualização automática com novos recursos para todos os usuários, e (3) acesso universal a partir de qualquer máquina conectada à Internet. Agora pelo lado negativo, o uso de tecnologias de servidor e navegador tornam as aplicações *web* particularmente propensas a erros e desafiadoras de testar, causando sérias ameaças à confiabilidade. Além dos custos financeiros, os erros nas aplicações *web* podem resultar na perda de receita e credibilidade (GAROUSI *et al.*, 2013).

2.3.1 API

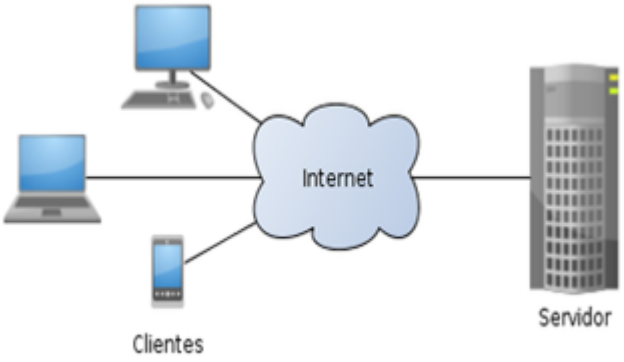
Application Programming Interface (API), são mecanismos que permitem que dois componentes de software se comuniquem usando um conjunto de definições e protocolos. Por exemplo, o sistema de *software* do instituto meteorológico contém dados meteorológicos diários. A aplicação para a previsão do tempo presente nos telefones “fala” com esse sistema por meio de APIs e mostra atualizações meteorológicas diárias no telefone.

No contexto de APIs, a palavra aplicação refere-se a qualquer *software* com uma função distinta. A interface pode ser pensada como um contrato de serviço entre duas aplicações. Esse contrato define como as duas se comunicam usando solicitações e respostas. A documentação de suas respectivas APIs contém

informações sobre como os desenvolvedores devem estruturar essas solicitações e suas respostas.

A arquitetura da API é geralmente explicada em termos de cliente e servidor (Figura 1). A aplicação que envia a solicitação é chamada de cliente e a aplicação que envia a resposta é chamada de servidor. Então, no exemplo do clima, o banco de dados meteorológico do instituto é o servidor e o aplicativo móvel seria o cliente (AWS, 2023b).

Figura 1 - Modelo Cliente-Servidor



Fonte: Mendes (2002)

2.3.2 *Front-end*

Front-end pode ser definido como a parte visual de um sistema *web*, aquilo que os usuários conseguem visualizar e interagir através de um navegador. Dessa forma, representa o lado do cliente em uma aplicação *web*.

2.3.3 *Back-end*

Já o *Back-end* lida com as tecnologias responsáveis por armazenar e de maneira segura, manipular os dados do usuário. Ele também é comumente associado à lógica escondida que dá vida às aplicações no qual os usuários interagem. Sendo assim, o *Back-end* é considerado o lado do servidor de uma aplicação *web*.

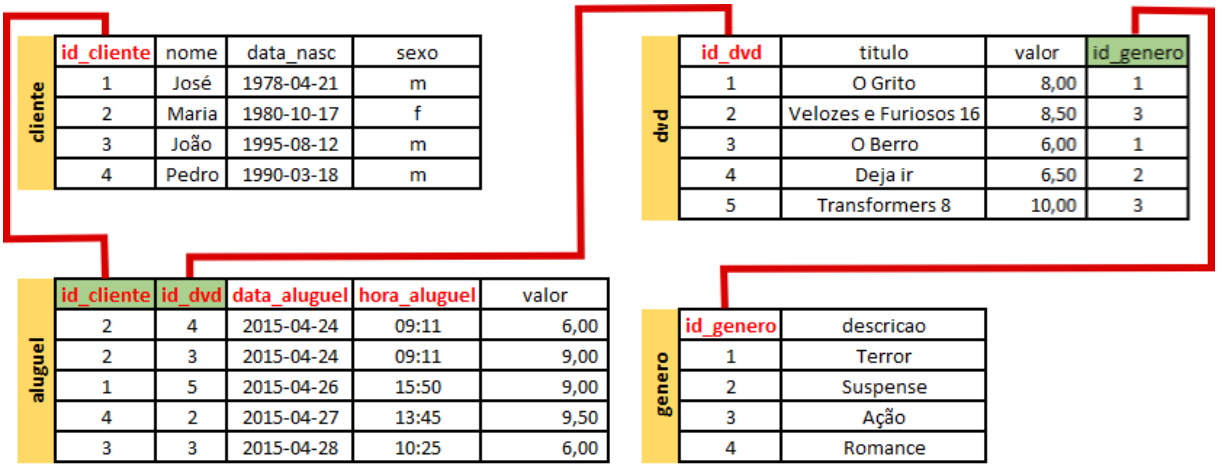
2.4 Banco de dados

Banco de dados trata-se de uma coleção de dados estruturados e organizados - que possuem uma relação entre si. A partir desses dados armazenados, é possível inferir informações sobre o mundo real, como vendas de produtos, dados pessoais e informações sobre os produtos. Por fim, um banco de dados possibilita ajudar as empresas a gerenciar seus negócios ao gerar a rastreabilidade das informações, podendo também armazenar essas informações com segurança.

2.4.1 Banco de dados relacional

No modelo relacional, os dados são representados sob a forma de tabelas (Figura 2). Cada tabela tem várias colunas, e cada coluna tem um nome único. Cada linha da tabela representa um pedaço de informação (SILBERSCHATZ; KORTH; SUDARSHAN, 2019).

Figura 2 - Modelo de dados relacional



Fonte: Cury (2015)

2.4.2 Banco de dados não-relacional

Os bancos de dados não-relacionais podem ser referidos como "NoSQL", que significa Não apenas SQL. Um banco de dados não relacional armazena dados de forma não tabular (Figura 3), e tende a ser mais flexível do que as estruturas tradicionais de banco de dados relacionais, baseadas em SQL. Em vez disso, bancos de dados não-relacionais podem ser baseados em estruturas de dados

como documentos. Um documento pode ser altamente detalhado e conter uma gama de diferentes tipos de informações em diferentes formatos. Esta capacidade de digerir e organizar vários tipos de informações lado a lado torna as bases de dados não-relacionais muito mais flexíveis do que as bases de dados relacionais (MONGODB, 2023b).

Figura 3 - Modelo de dados não relacional

Fonte: MongoDB (2023b)

2.5 Computação em nuvem

Segundo Chee e Franklin (2019), a computação em nuvem é um modelo de processamento de informações no qual capacidades computacionais administradas são centralmente entregues como serviços, conforme a necessidade, em toda a rede para uma variedade de dispositivos voltados para o usuário.

As empresas fornecedoras desse tipo de serviço são conhecidas como Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), Alibaba Cloud, Oracle Cloud, IBM Cloud, Tencent Cloud, entre outros.

2.6 Aprendizado de máquina

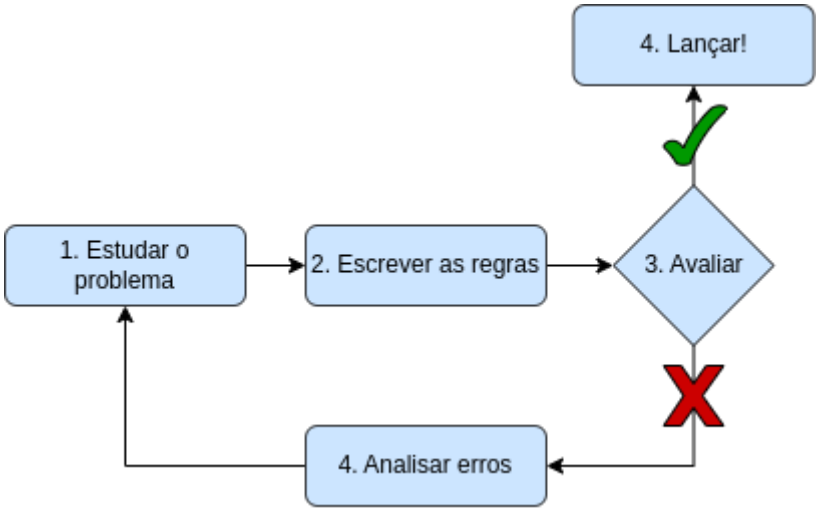
Géron (2022) define aprendizado de máquina (ML - do inglês, *Machine Learning*) como a ciência (e a arte) da programação de computadores para que eles possam aprender com os dados. Em relação a essa aprendizagem, Huyen (2022) postula que o conhecimento é adquirido através dos complexos padrões dos dados existentes e usa esses padrões para realizar predições em dados não analisados.

Na abordagem tradicional da utilização de algoritmos, o processo para resolver um problema pode ser trabalhoso. No cenário em que se queira filtrar

conteúdo textual que não respeita as políticas de direitos humanos de um fórum, por exemplo, pode seguir as seguintes etapas também visto na Figura 4.

1. É necessário **estudar o problema** para entender os padrões de comportamentos;
2. Com os cenários definidos, são **escritas as regras** necessárias para resolver o problema;
3. De modo a analisar o sucesso do algoritmo, são aplicados **testes**;
4. Se os resultados forem positivos dado uma métrica de comparação, o algoritmo pode ser **lançado** em produção;
5. Caso contrário, **os erros são analisados** de modo a reavaliar o processo realizado - volta-se à etapa 1.

Figura 4 - Abordagem tradicional de se resolver problemas computacionais



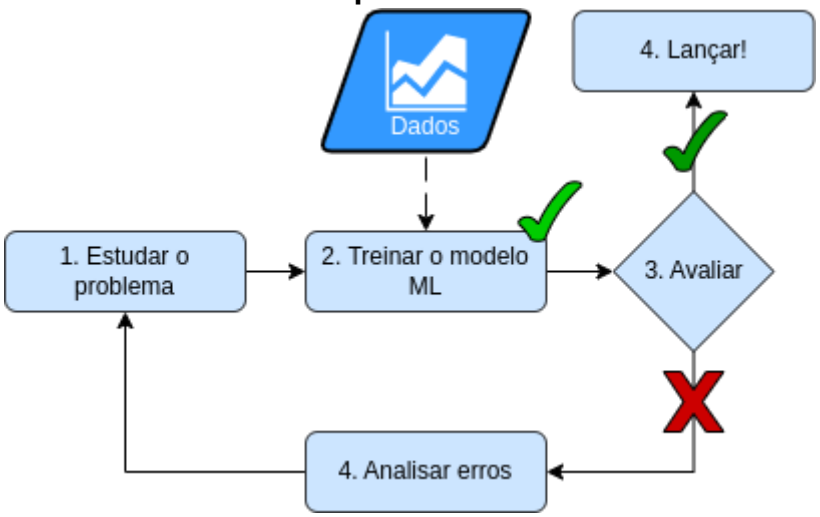
Fonte: Géron (2022)

Ao longo do tempo, torna-se difícil manter o algoritmo do cenário apresentado dado inúmeras regras criadas. Em contrapartida, se usar algum algoritmo de ML para aprender automaticamente quais palavras e frases são ótimas para prever o descumprimento das políticas de interação entre os usuários do fórum, será possível detectar padrões de palavras invulgarmente frequentes de maneira muito mais rápida, tornando fácil de manter o código e além de ser muito mais preciso ao reduzir os erros manuais e não mapeados por humanos.

Com essa nova abordagem de se resolver problemas, suas etapas consiste em (também visto na Figura 5):

1. Na realização do **estudo do problema** para entender os padrões de comportamentos;
2. **Treina-se um modelo** (o algoritmo escrito) de aprendizado de máquina o qual faz o uso de uma coleção dados para se extrair os padrões de conhecimento;
3. De modo a analisar o sucesso do modelo, são aplicados **testes**;
4. Se os resultados forem positivos dado uma métrica de comparação, o modelo pode ser **lançado** em produção;
5. Caso contrário, **os erros são analisados** de modo a reavaliar o processo realizado - volta-se à etapa 1.

Figura 5 - Abordagem de aprendizado de máquina de se resolver problemas computacionais



Fonte: Géron (2022)

Em resumo, Géron (2022) indica a utilização de aprendizado de máquina para:

- Problemas para os quais as soluções existentes requerem muitos ajustes no algoritmo ou possui uma longa lista de regras (um modelo de aprendizagem de máquina pode muitas vezes simplificar o código e possuir melhor desempenho do que a abordagem tradicional);
- Problemas complexos para os quais o uso de uma abordagem tradicional não produz boa solução (as melhores técnicas de aprendizagem de máquinas talvez possam encontrar uma solução);
- Ambientes flutuantes (um sistema de aprendizagem de máquinas pode ser facilmente reutilizado sobre novos dados, mantendo-os sempre atualizados);

- Para obter soluções sobre problemas complexos e que recorra a abundantes dados.

2.7 Processamento de dados

Em um sistema, a maneira que os dados são processados pode ser importante na tomada de decisões de uma empresa. No caso, existem duas possibilidades: em lote e em tempo real.

2.7.1 Em lote

No processamento de dados em lote, envolve armazenar todos os dados recebidos até que uma certa quantidade seja coletada, para depois serem processados em lote. Isso significa que:

1. Esse processamento de dados é agendado em um horário específico;
2. É processado uma quantidade de dados suficiente para que o sistema esteja funcional.

Nessa abordagem, menos recursos computacionais são empregados e é mais fácil de implementar. O que significa ser menos custoso para o dono do sistema e também para os usuários no quesito de velocidade para recuperar as informações processadas.

2.7.2 Em tempo real

O processamento de dados em tempo real consiste em processar os dados o mais rápido possível e em pequenas quantidades. Isso significa que:

1. Esse processamento de dados é imediato e constantemente atualizado;
2. O processamento é realizado no momento do evento acontecido (solicitação de execução de uma funcionalidade de um sistema, por exemplo).

Nessa abordagem, emprega-se um alto custo computacional e de alta complexidade de implementação para garantir que o sistema processe os dados em tempo hábil e retorne seu resultado ao usuário.

2.8 Sistemas de recomendação

Segundo S e John (2016), os sistemas de recomendação são ferramentas de *software* que fornecem sugestões sobre itens ao usuário. Este sistema fornece sugestões para o usuário na tomada de decisões, tais como onde planejar uma viagem, quais itens podem ser trazidos, que notícias interessantes estão lá para ler, etc. Consegue lidar com o problema de sobrecarga de informações. Há muitos métodos para lidar com os desafios do sistema de recomendação como esparsidade de dados, escalabilidade, problema de cauda longa, problema de inicialização gelada (não gera recomendações relevantes quando não há relacionamentos suficientes entre os dados), etc.

Os sistemas de recomendação são úteis tanto para usuários como para os fornecedores.

Valor para os usuários:

- Recomendar itens que sejam interessantes para eles;
- Diminuir o número de escolhas;
- Explorar o espaço de opções;
- Encontrar novos itens;
- Entretenimento.

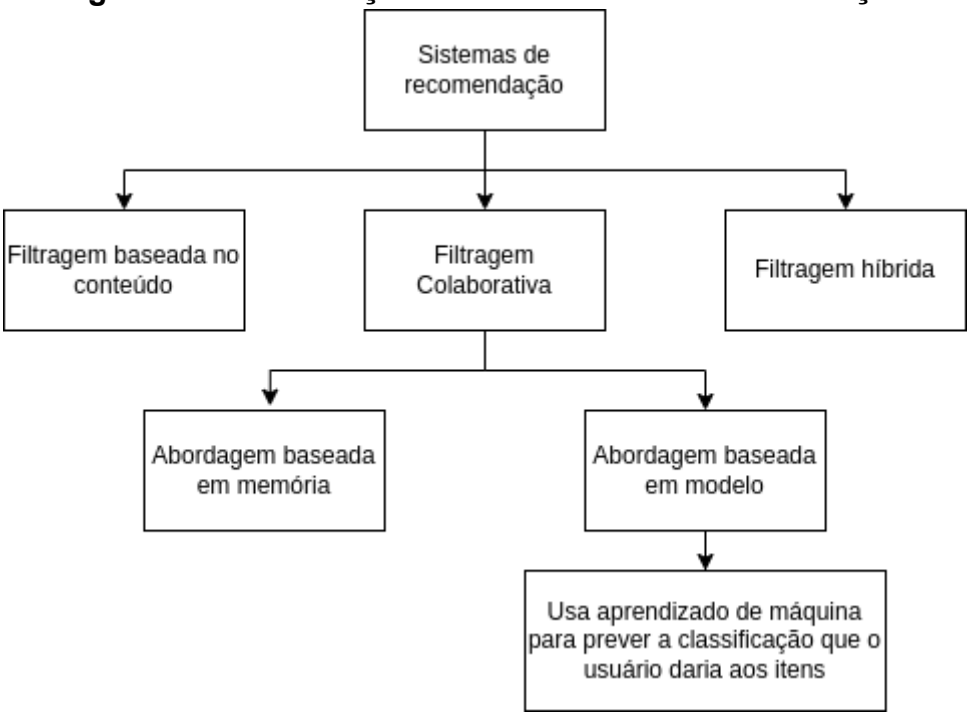
Valor para os fornecedores:

- Assegurar um serviço personalizado único para os usuários;
- Aumentar a confiança e lealdade dos usuários;
- Aumentar as vendas de itens;
- Possibilidades de promoção;
- Ganhar conhecimento sobre os usuários.

2.9 Classificação dos sistemas de recomendação

Com base em como os sistemas analisam e filtram as informações conforme as exigências do usuário, há muitos sistemas de recomendação, como visto na Figura 6 (S; JOHN, 2016).

Figura 6 - Classificação dos sistemas de recomendação



Fonte: Elaborado pela autora

2.9.1 Recomendação baseada em conteúdo

O sistema de recomendação baseado em conteúdo, depende da análise da descrição do item o qual encontra os seus similares. Essa descrição pode ser composta pelo maior número de atributos que diferenciam um item do outro, como: título, sinopse, elenco, ano de lançamento, diretor, etc. Há dois tipos de abordagens na recomendação baseada em conteúdo: somente considera a análise da descrição do conteúdo e no segundo, constrói-se o perfil do usuário e do item dado a avaliação do usuário sobre o conteúdo.

2.9.1.1 Somente analisa a descrição do conteúdo

Nesse sistema, somente é considerada a descrição dos itens. Encontrado uma similaridade entre os mesmos, é gerado a lista de recomendação.

Vantagens:

- Não depende de avaliações do usuário ou de outros usuários para gerar a recomendação - inicialização gelada.

Desvantagens:

- Se a base de dados dos itens não mudar conforme o tempo, a recomendação pode trazer os mesmos resultados na mesma ordem;
- Não filtra o que o usuário já viu.

2.9.1.2 Construção do perfil do usuário e do item dado sua avaliação

S e John (2016) postulam que o sistema de recomendação é baseado na descrição do item e no perfil do usuário, o qual é construído de acordo com suas preferências. Os itens são descritos usando palavras-chave e o perfil do usuário contém todos os itens que o usuário gosta. Este sistema de recomendação recomenda itens ao usuário de acordo com seu perfil. Os perfis do usuário são atualizados automaticamente com base em seu *feedback*.

Utiliza-se técnicas de filtragem para a recuperação de informações, nas quais alguns itens candidatos são combinados com aqueles itens que o usuário havia avaliado previamente e recomenda o melhor item a ele. O resultado é uma pontuação de desempenho baseada no grau de correspondência entre a descrição dos itens e a avaliação do usuário.

Vantagens:

- Gera recomendação com base nas preferências do usuário;
- Sem problemas de esparsidade;
- Sem problema de inicialização gelada;
- Atualização automática.

Desvantagens:

- Depende do conhecimento preexistente do usuário para gerar a recomendação.

2.9.2 Recomendação colaborativa

Segundo Isinkaye, Folajimi e Ojokoh (2015), a recomendação colaborativa (RC) é uma previsão independente de domínio técnica para o conteúdo que não

pode ser facilmente descrito por metadados, como filmes e músicas. A técnica de RC funciona criando um banco de dados (matriz usuário-item) de preferências para itens por usuários. Em seguida, combina os usuários com interesses e preferências relevantes, calculando semelhanças entre os seus perfis para fazer recomendações. Esses usuários constroem um grupo chamado de vizinhança.

Um usuário recebe recomendações para os itens que ele não avaliou antes, mas que já foram avaliados positivamente pelos usuários em sua vizinhança. Recomendações produzidas pela RC podem ser de previsão ou recomendação. A previsão é um valor numérico, Rij, expressando a avaliação prevista do item *j* para o usuário *i*, enquanto a recomendação é uma lista dos principais N itens que o usuário gostará mais. A técnica de recomendação colaborativa pode ser dividida em duas categorias: baseada em memória e baseada em modelo.

2.9.2.1 Recomendação colaborativa baseada em memória

Os itens que já foram avaliados pelo usuário desempenham um papel relevante na busca por um vizinho que compartilhe apreço com ele. Uma vez que um vizinho de um usuário é encontrado, diferentes algoritmos podem ser usados para combinar as preferências dos vizinhos para gerar as recomendações. Devido à eficácia dessas técnicas, elas alcançaram um sucesso generalizado em aplicações da realidade (ISINKAYE; FOLAJIMI; OJOKOH, 2015). Há dois tipos de abordagens de recomendação colaborativa baseada em memória: baseada no item e baseada no usuário.

2.9.2.1.1 Recomendação baseado em itens

Neste sistema, S e John (2016), dizem que as relações entre itens são identificadas usando uma matriz de itens-usuário, o qual é usado para calcular indiretamente as recomendações para os usuários. Ele fornece aos usuários um item como recomendação, dependendo de outros itens com altas correlações. Um conjunto de itens similares para cada item é primeiro encontrado, ou seja, o conjunto de seus vizinhos. Este sistema prevê as avaliações do usuário para um determinado item, dependendo da avaliação dada ao item similar pelo mesmo usuário.

Vantagens:

- Alto desempenho
- Boa qualidade de predição

2.9.2.1.2 Recomendação baseado no usuário

No sistema colaborativo baseado no usuário, recomenda-se um item para o usuário com base na opinião de outros usuários de mente semelhante para aquele item. Ele encontra os usuários com interesses comuns e os considera como os vizinhos mais próximos. Então a previsão desse usuário para o item será feita com base na avaliação que os vizinhos do usuário dão para aquele item (S; JOHN, 2016).

Vantagens:

- Simples
- Eficiente
- Preciso

Desvantagens:

- Problema de inicialização gelada
- Problema de escalabilidade
- Problema de esparsidade de dados

2.9.2.2 Recomendação colaborativa baseada em modelo

Essa técnica emprega as avaliações anteriores para aprender um modelo, a fim de melhorar o desempenho da técnica de recomendação colaborativa. O processo de construção do modelo pode ser feito usando ML ou técnicas de mineração de dados. Essas técnicas podem recomendar rapidamente um conjunto de itens pelo fato de usarem um modelo pré-calculado e provado produzir resultados de recomendação semelhantes às técnicas de recomendação baseadas em vizinhança.

Exemplos dessas técnicas incluem a técnica de redução de dimensionalidade, como decomposição em valores singulares (SVD), técnica de

completação de matriz, métodos semânticos latentes, regressão e *clustering*. Recomendações baseadas em modelos analisam a matriz usuário-item para identificar relações entre itens; eles usam essas relações para comparar a lista de recomendações mais populares. As técnicas baseadas em modelos resolvem os problemas de esparsidade associados aos sistemas de recomendação (ISINKAYE; FOLAJIMI; OJOKOH, 2015).

2.9.3 Recomendação não personalizada

Por fim, existe a recomendação não personalizada que não necessariamente faz uso do ML, entretanto sua abordagem incentiva as pessoas usarem o sistema até que se forme uma base de dados adequada para melhorar as outras recomendações. Um exemplo de recomendação não personalizada é fornecer a listagem dos produtos recém lançados, os mais populares e até mesmo os mais vendidos.

2.10 Algoritmos de sistemas de recomendação

Nesta seção, serão apresentados os algoritmos vetorização de contagem e TF-IDF que podem ser utilizados na recomendação baseada no conteúdo. Eles são métodos do campo de estudo de processamento de linguagem natural (NLP - do inglês, *Natural Language Processing*). NLP refere-se ao ramo da ciência da computação – e, mais especificamente, ao ramo da inteligência artificial ou IA – preocupado em dar aos computadores a capacidade de entender texto e palavras faladas da mesma maneira que os seres humanos (IBM, 2023c).

Outros algoritmos também apresentados são a fatoração de matriz e os mínimos quadrados alternados que é uma variação do algoritmo fatoração de matriz. Eles podem ser aplicados na recomendação colaborativa.

2.10.1 Vetorização de contagem

Vetorização de contagem (tradução literal de *Count vectorizing*) é um dos métodos NLP que consiste em contar o número de ocorrências que cada palavra aparece em um documento (qualquer descrição textual que pode ser diferenciada

por outra descrição, o escopo pode ser desde frases, parágrafos, artigos ou até mesmo livros).

O resultado desta contagem consiste numa matriz esparsa que pode ser utilizada em algoritmos de aprendizado de máquina. Posteriormente, é necessário calcular o grau de similaridade entre as matrizes (seção 2.11.1) para realizar a recomendação baseada no conteúdo.

Matriz esparsa ou dados esparsos, diz respeito a uma predominância de elementos com valor zero ou ausentes.

2.10.2 TF-IDF

Segundo Casarotto (2019), a Frequência do Termo – Frequência Inversa dos Documentos (TF-IDF - do inglês, *Term Frequency-Inverse Document Frequency*) é um cálculo estatístico utilizado para medir quais termos são mais relevantes dado um documento, analisando a frequência com que aparecem, em comparação à sua frequência em um conjunto maior de documentos.

O cálculo (equação 1) serve como fator de ponderação de termos, ou seja, para entender a importância de um termo ou frase específica para determinado documento.

TF se refere à “frequência do termo”. Essa parte do cálculo responde à pergunta: com que frequência o termo aparece nesse documento? Quanto maior for a frequência no documento, maior será a importância do termo.

Já o IDF significa “frequência inversa dos documentos”. Nessa parte, a ferramenta responde: com que frequência o termo aparece em todos os documentos da coleção? Quanto maior for a frequência nos documentos, menor será a importância do termo.

O cálculo do IDF considera que termos que se repetem frequentemente nos textos — como artigos e conjunções (a, o, e, mas, que, etc.) — não têm relevância para os documentos.

Então, quando o fator IDF é incorporado, o cálculo diminui o peso dos termos que ocorrem com muita frequência no conjunto de documentos e aumenta o peso dos termos que ocorrem raramente.

$$W_{x,y} = tf_{x,y} \times log\left(\frac{N}{df_x}\right) \quad (1)$$

Onde:

Termo *x* dentro do documento *y*;

$tf_{x,y}$ = frequência de *x* em *y*;

df_x = número de documentos que contém *x*;

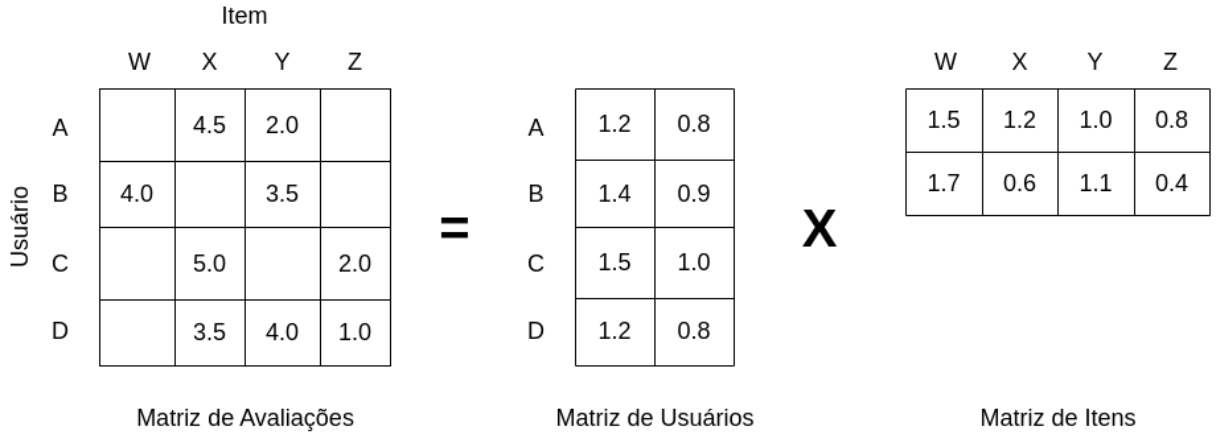
N = número total de documentos.

No quesito de sua aplicabilidade, quando selecionado os atributos de interesse do item, o TF-IDF calcula a frequência desses atributos e transforma esse resultado em uma matriz esparsa.

2.10.3 Fatoração de matriz

Segundo Liao (2018), na recomendação colaborativa, a fatoração de matriz (Figura 7) é uma solução de última geração para problemas de dados esparsos, embora tenha se tornado amplamente conhecida desde o Desafio do Prêmio Netflix[1].

Figura 7 - Exemplo de fatoração de matriz das avaliações de filmes



Fonte: Liao (2018)

A fatoração matricial é simplesmente uma família de operações matemáticas para matrizes em álgebra linear. De maneira mais específica, uma fatoração de

[1] Em 2006, a Netflix, prestes a entrar no ramo de streaming de vídeos, lançou um desafio à comunidade para melhorar seu sistema de recomendação em ao menos 10%. Como recompensa, os vencedores ganhariam algo em torno de $1.000.000 de dólares.

matriz é uma fatoração de uma matriz em um produto de matrizes. No caso da recomendação colaborativa, os algoritmos de fatoração matricial funcionam decompondo a matriz de interação usuário-item no produto de duas matrizes retangulares de menor dimensionalidade. Uma matriz pode ser vista como a matriz do usuário, o qual as linhas representam os usuários e as colunas são fatores latentes. A outra matriz é a matriz de item onde as linhas são fatores latentes e as colunas representam itens.

A fatoração de matriz, resolve:

1. O modelo aprende a fatorar a matriz de avaliações nas representações do usuário e do item, o que permite que o modelo preveja melhores avaliações personalizadas de itens para os usuários;
2. Com a fatoração matricial, os itens menos conhecidos podem ter representações latentes ricas tanto quanto os itens populares, melhorando a capacidade do recomendador em recomendar itens menos conhecidos.

Na matriz de interação usuário-item esparsa, a avaliação prevista do usuário *u* dará ao item *i* calculado como mostrado na equação 2:

$$\bar{r}_{ui} = \sum_{f=0}^{n\,fatores} H_{u,f} W_{f,i} \quad (2)$$

Onde H é a matriz do usuário, W é a matriz do item.

A avaliação do item *i* dado pelo usuário *u* pode ser expressa como um produto escalar do vetor latente do usuário e do vetor latente do item.

Dado a equação acima, o número de fatores latentes pode ser ajustado via validação cruzada. Fatores latentes são as características no espaço latente de dimensão inferior projetadas a partir da matriz de interação usuário-item. A ideia por trás da fatoração matricial é usar fatores latentes para representar as preferências do usuário ou tópicos dos itens em um espaço de dimensão muito menor. A fatoração de matriz é uma das técnicas de redução de dimensão muito eficaz no aprendizado de máquina.

O número de fatores latentes determina a quantidade de informações abstratas que se quer armazenar em um espaço de dimensão inferior. Uma fatoração matricial com um fator latente é equivalente a um recomendador mais

popular (por exemplo, recomenda os itens com mais interações sem qualquer personalização). Aumentar o número de fatores latentes melhora a personalização, até que o número de fatores se torne muito alto, momento em que o modelo começa a se sobre ajustar (o modelo consegue prever as avaliações, mas não consegue reproduzir o comportamento para novos dados, dá impressão que o modelo decora os resultados). Uma estratégia comum para evitar o sobreajuste é adicionar regularização à função objetivo.

O objetivo da fatoração matricial é minimizar o erro entre a avaliação verdadeira e a avaliação prevista visto na equação 3:

$$arg\,min_{H,W} \|R - \overline{R}\|_F + \alpha \|H\| + \beta \|W\| \quad (3)$$

Onde H é a matriz do usuário, W é a matriz do item.

Uma vez que se tem uma função objetiva, só precisa de uma rotina de treinamento para concluir a implementação de um algoritmo de fatoração de matriz (LIAO, 2018).

2.10.4 Algoritmo de mínimos quadrados alternados

Mínimos quadrados alternados (tradução literal de ALS - do inglês, *Alternating Least Square*) é um eficiente algoritmo (Figura 8) de fatoração de matriz para recomendação colaborativa que pode ser executado paralelamente e consegue trabalhar com um grande conjunto de dados. Serve para prever avaliações da matriz esparsa da relação usuário-item. Matematicamente falando (equação 4), encontra duas matrizes, P e Q, de modo que seu produto seja aproximadamente igual à matriz original de usuários e itens. Uma vez que tais matrizes tenham sido encontradas, é possível prever qual avaliação do usuário *u* do item *i* multiplicando a linha *u* de P pela linha *i* de Q (SOPHIE, 2018).

Função objetiva

$$minimize_{p,q} \sum_{(u,i) \in S} \left(r_{ui} - < p_u, q_i >\right)^2 + \lambda \left[\|p\|^2_{Frob} + \|q\|^2_{Frob}\right] \quad (4)$$

Mínimos quadrados alternados

$$p_u \leftarrow \left[\lambda 1 \ + \sum_{i|(u,i)\in S} q_i q_i^{\top}\right]^{-1} \sum_i q_i r_{ui} \tag{4.1}$$

$$q_i \leftarrow \left[\lambda 1 \ + \sum_{u|(u,i)\in S} p_u p_u^{\top}\right]^{-1} \sum_i p_u r_{ui} \tag{4.2}$$

O método é chamado de “alternado”, uma vez que as duas etapas a seguir são repetidas:

1. Corrija a matriz P e encontre a matriz ideal Q;
2. Corrija a matriz Q e encontre a matriz ideal P.

O bit “mínimos quadrados” de ALS entra em jogo quando se encontram as matrizes "ideais", porque "ideal", neste caso, é entendido como "a minimização do erro dos mínimos quadrados" (além da regularização que impede o sobreajuste) (SOPHIE, 2018).

Figura 8 - Exemplo de algoritmo ALS

Algorithm 1 The ALS algorithm

```
procedure ALS(R, f, λ; X, Y)
    X ← 0, Y ← random initial guess
    repeat
        for row u ← 1, m do
            x_u ← (Y^T Y + λI)^{-1} Y^T r_u by solving the linear system
                (Y^T Y + λI) x_u = Y^T r_u
        end for
        for column i ← 1, n do
            y_i ← (X^T X + λI)^{-1} X^T r_i by solving the linear system
                (X^T X + λI) y_i = X^T r_i
        end for
    until reaching the maximum iterations
end procedure
```

Fonte: Chen *et al* (2020)

2.11 Métrica de comparação

Na recomendação baseada em conteúdo, é necessário aplicar uma métrica de comparação para validar o quão a descrição de um item é similar ao outro e para esse trabalho, foi utilizado a similaridade de cosseno, apesar de ter muitas outras.

2.11.1 Similaridade de cosseno

De acordo com Alake (2020), a similaridade de cosseno é uma medida que quantifica a semelhança entre dois ou mais vetores. A similaridade de cosseno é o cosseno do ângulo entre os vetores. Os vetores são tipicamente não zero e estão num espaço de produto interno.

A similaridade de cosseno é descrita matematicamente (equação 5) como a divisão entre o produto escalar dos vetores e o produto das normas euclidianas ou magnitude de cada vetor.

$$similaridade\ de\ cosseno\ = S_C(A,\ B) := cos(\theta)\ = \frac{A \cdot B}{\|A\|\|B\|}\ = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (5)$$

A quantificação da similaridade entre dois documentos pode ser obtida convertendo as palavras ou frases do documento, ou frases em uma forma vetorizada de representação.

As representações vetoriais dos documentos podem então ser usadas na equação de similaridade de cosseno para obter uma quantificação da similaridade.

No cenário descrito acima, a similaridade de cosseno de 1 implica que os dois documentos são exatamente iguais e uma similaridade de cosseno de 0 apontaria para a conclusão de que não há semelhanças entre os dois documentos (ALAKE, 2020).

2.12 Métricas de avaliação

Quando se trabalha com algoritmos de aprendizado de máquina, é importante avaliar o quão bem eles desempenham. No caso, da recomendação colaborativa, o quão assertivo foi a previsão da avaliação do usuário dado um título selecionado. Como o algoritmo da recomendação pode ser classificado como regressão, escolheu-se trabalhar com as métricas raiz do erro quadrático médio e coeficiente de determinação para avaliar seu desempenho. Decidiu-se trabalhar com essas duas métricas, pois sua avaliação individual pode resultar em uma conclusão equivocada.

Os algoritmos de aprendizado de máquina podem ser classificados em dois tipos: regressão e classificação. Fundamentalmente, Brownlee (2017) postula que a

classificação é sobre a previsão de um rótulo (característica do item de análise) e a regressão é sobre a previsão de um valor numérico.

2.12.1 Raiz do erro quadrático médio

A raiz do erro quadrático médio (RMSE - do inglês, *Root Mean Squared Error*) é uma das métricas mais comumente usadas para avaliar a qualidade das previsões (equação 6). Ela mostra quão longe (interpretado como erro) as previsões caem dos valores reais medidos usando a distância euclidiana (C3.AI, 2023). É uma métrica que varia de zero a infinito. Quanto mais próximo de zero, melhor.

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \quad (6)$$

2.12.2 Coeficiente de determinação

Segundo Turney (2022), o coeficiente de determinação (ou R²) mede o quão bem um modelo estatístico prevê um resultado (equação 7). O resultado é representado pela variável dependente do modelo.

Apesar de o nome R² conter "ao quadrado", os valores R² variam de infinito negativo a 1. Simplificando, quanto melhor um modelo estiver fazendo previsões, mais próximo o R² estará de 1 (TURNEY, 2022). Se o resultado for negativo, é porque o modelo precisa ser revisto.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n} \left(y_i - \hat{y}\right)}, \bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i \quad (7)$$

3. METODOLOGIA

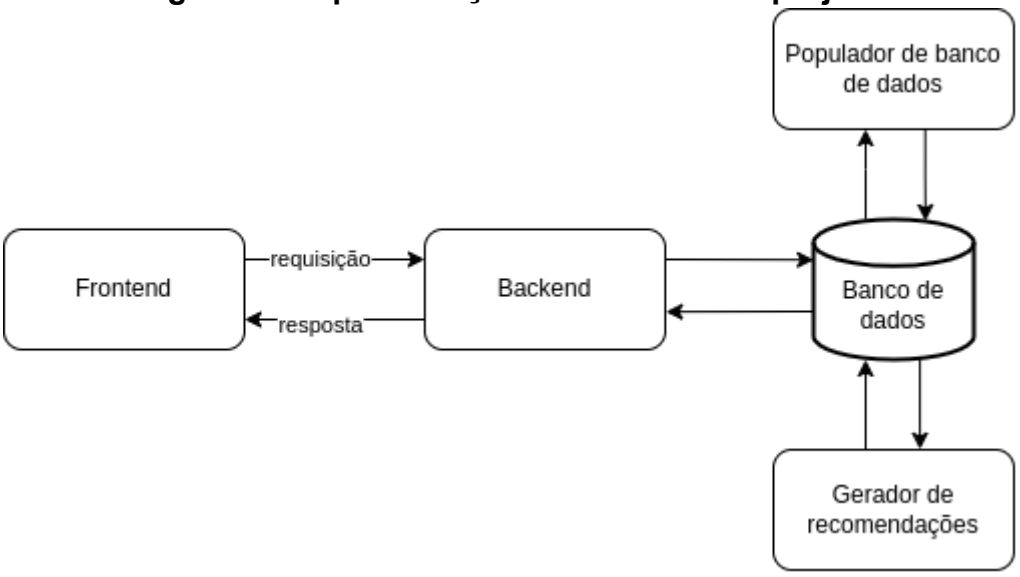
Neste capítulo, serão apresentadas as ferramentas e os métodos utilizados pelo projeto desenvolvido.

3.1 Estrutura do projeto

A estrutura do projeto foi dividida em cinco partes: *front-end*, *back-end*, *scripts* com os sistemas de recomendação, o banco de dados (BD) e automação que popula o BD. Estrutura também representada pela Figura 9.

O *front-end* disponibiliza as telas que o usuário poderá interagir pelo navegador, o qual se comunica com o *back-end* através de sua API. O *back-end* tem todas as funcionalidades que não estejam relacionadas com o sistema de recomendação e fornece uma API. Os *scripts* com os sistemas de recomendação persistem as recomendações geradas no banco de dados, e o *back-end* acessa as informações pelo mesmo banco. Por fim, o BD armazena os dados sobre os filmes, as séries, os usuários e suas interações com o sistema.

Figura 9 - Representação da estrutura do projeto



Fonte: Elaborado pela autora.

3.2 Obtenção dos dados

O sistema é abastecido através de uma automação com os dados do catálogo provenientes do projeto *The Movie Database* (TMDB) via API (Figura 10). Conforme a página do projeto[2], TMDB "é um banco de dados de filmes e séries construído pela comunidade. Cada pedaço de dado foi adicionado por nossa incrível comunidade que data desde 2008." (TMDB, 2023, tradução nossa). Embora apoiemos oficialmente 39 idiomas, também temos amplos dados regionais. Todos os dias o TMDB é utilizado em mais de 180 países. (TMDB, 2023).

Figura 10 - Logo do projeto TMDB

Fonte: TMDB (2023)

Com data de acesso de 3 janeiro de 2023, estatisticamente a base do projeto consiste de:

- 768.777 filmes;
- 142.266 séries;
- 229.201 temporadas;
- 3.529.079 episódios;
- 2.678.721 pessoas (atores e equipe de gravação);
- 4.072.984 imagens;

O processo para obter os dados necessários constituiu-se em criar uma conta na plataforma, obter acesso à API deles e usar a documentação para selecionar quais recursos consumir e informar o que precisou ser parametrizado.

Os dados são retornados no formato notação de objetos Javascript (JSON - do inglês, *Javascript Object Notation*) que depois são salvos pela automação no

[2] TMDB. **Let's talk about TMDB**. 2023. Disponível em <https://www.themoviedb.org/about>. Acesso em 3 de janeiro de 2023.

banco de dados. Os dados disponíveis foram salvos tanto em português brasileiro quanto em inglês.

3.3 Ferramentas utilizadas

Apesar de usar *frameworks* para acelerar o desenvolvimento do projeto, foi utilizado os conhecimentos sobre HTML, CSS e Javascript. Elas são ferramentas que viabilizam o desenvolvimento básico de um website.

3.3.1 HTML

O HTML (Linguagem de Marcação de HiperTexto) é o bloco de construção mais básico da *web* (Figura 11). Define o significado e a estrutura do conteúdo da *web* (MDN CONTRIBUTORS, 2022a).

Segundo Mdn Contributors (2022a), "Hipertexto" refere-se aos *links* que conectam páginas da *Web* entre si, seja num único site ou entre sites. *Links* são um aspecto fundamental da web. Ao carregar conteúdo na Internet e vinculá-lo a páginas criadas por outras pessoas, você se torna um participante ativo na *web*.

O HTML usa "Marcação" para anotar texto, imagem e outros conteúdos para exibição em um navegador da *Web* (MDN CONTRIBUTORS, 2022a).

Figura 11 - Exemplo de um documento HTML

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Example</title>
5          <link rel="stylesheet" href="sty
6      </head>
7      <body>
8          <h1>
9              <a href="/">Header</a>
10         </h1>
11         <nav>
12             <a href="one/">One</a>
13             <a href="two/">Two</a>
14             <a href="three/">Three</a>
15         </nav>
```

Fonte: Commons (2023)

3.3.2 CSS

O CSS (do inglês, *Cascading Style Sheets* ou Folhas de Estilo em Cascata) é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML ou em XML (Figura 12). O CSS descreve como elementos são mostrados na tela, no papel, na fala ou em outras mídias (MDN CONTRIBUTORS, 2022b).

Figura 12 - Exemplo de uma folha de estilo CSS

```
h1 {
    font-family: courier, courier-new, serif;
    font-size: 20pt;
    color: blue;
    border-bottom: 2px solid blue;
}
p {
    font-family: arial, verdana, sans-serif;
    font-size: 12pt;
    color: #6B6BD7;
}
.red_txt {
    color: red;
}
```

Fonte: Eletronics (2013)

3.3.3 Javascript

O Javascript é uma linguagem de *script* que possibilita a criação de páginas dinâmicas (Figura 13). Nativamente ela é executada no navegador, contudo existe a sua versão para servidores chamada Node.js. Com Javascript é possível criar elementos HTML na página, mudar o conteúdo existente, modificar os estilos CSS; reagir às interações do usuário (*clicks* do mouse, movimentos do ponteiro e pressionamento das teclas do teclado); disparar requisições entre o servidor remoto, baixar e subir arquivos; obter e definir os *cookies*, perguntar questões ao visitante, mostrar mensagens; lembrar dados no lado do cliente (o navegador do usuário) - (KANTOR, 2022).

Figura 13 - Exemplo de script Javascript

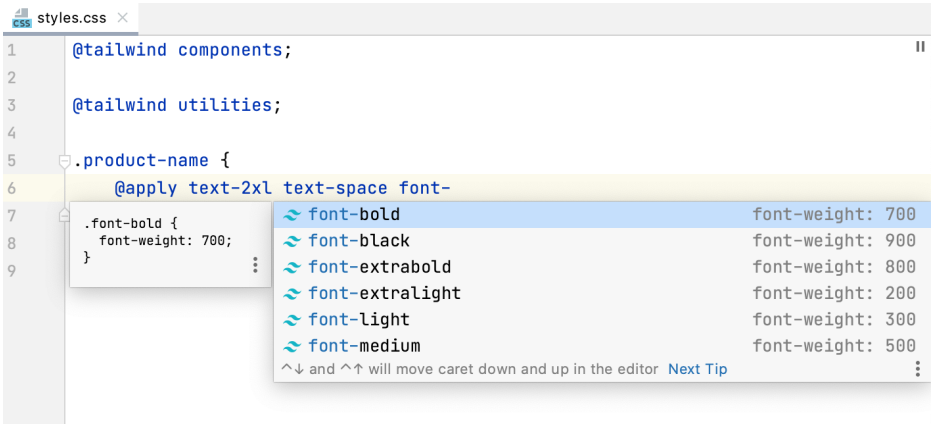
```
<HTML>
<script type="text/javascript">
/**
 * This is a multiple-
 * line comment
 */
var index = 0;
var arr = [];

function push(elem) {
    // This comment may span only this line
    arr[index++] = elem;
}
</script>
</HTML>
```

Fonte: Eclipse (2023)

3.3.4 Tailwind CSS

É um *framework* CSS que ajuda a desenvolver *websites* modernos com uso de uma vasta coleção de classes de estilos (Figura 14). É uma ferramenta extremamente customizável ao permitir que os desenvolvedores criem suas próprias classes utilitárias. A principal desvantagem está em usar a ferramenta sem um *framework* Javascript, porque como existem diversas combinações de estilos, até acertar um padrão que queira usar nos elementos, pode deixar o código verboso e gerar duplicidade. Contudo, ao usar uma *framework* Javascript esse problema pode ser contornado ao criar componentes e com o *tail*, fazer a referência do componente a uma classe customizada.

Figura 14 - Exemplo de código Tailwind CSS

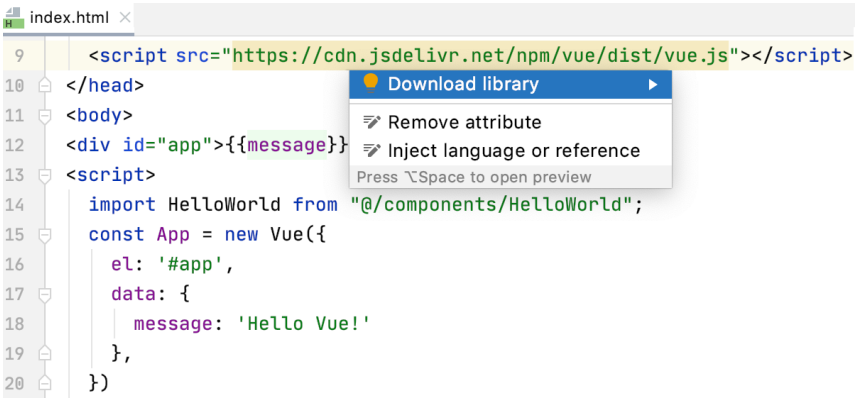


Fonte: Jetbrains (2023b)

3.3.5 Vue.js

Em conjunto ao Javascript, foi utilizado o *framework* Vue.js. Ele é usado para construir, principalmente, interfaces de usuário (Figura 15). Ele se baseia no HTML, CSS e JavaScript e fornece um modelo de programação declarativo e baseado em componentes que ajuda a desenvolver interfaces de usuário eficientemente, seja ela simples ou complexa (INTRODUCTION, 2022).

Figura 15 - Exemplo de código Vue.js



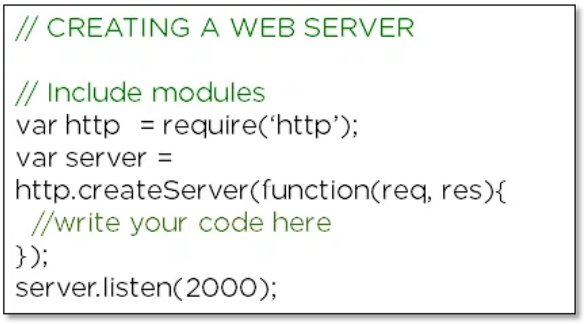
Fonte: Jetbrains (2022)

3.3.6 Node.js

Node.js é um ambiente em que é possível executar código Javascript. Ele representa o lado do servidor, onde se encontra a lógica do funcionamento do sistema *web*. Node.js é de código aberto, multiplataforma com foco em desenvolver aplicações servidoras e de rede (Figura 16).

Com Node.js os desenvolvedores conseguem desenvolver aplicações rápidas, escalonáveis, com código de fácil compreensão e também adota o estilo de programação assíncrona.

Figura 16 - Exemplo de código Node.js



Fonte: Sufiyan (2023)

3.3.7 Java

Java é uma linguagem de programação amplamente usada para codificar aplicações *Web*. Ela tem sido uma escolha popular entre os desenvolvedores há mais de duas décadas, com milhões de aplicações Java em uso hoje. Java é uma linguagem multiplataforma, orientada a objetos e centrada em rede que pode ser usada como uma plataforma em si (Figura 17). É uma linguagem de programação rápida, segura e confiável para codificar tudo, desde aplicações móveis e software empresarial até aplicações de *big data* e tecnologias do servidor (AWS, 2023a).

Possui funcionalidades que permitem trabalhar com programação funcional caso necessário. Por ser uma das linguagens predominantes no mercado, com uma vasta comunidade e atualizações recentes, ela desbanca gradualmente o estigma de ser uma linguagem pesada que consome muita memória RAM.

Figura 17 - Exemplo de código Java

```
package rentalStore;
import java.util.Enumeration;
import java.util.Vector;

class Customer {
    private String _name;
    private Vector<Rental> _rentals = new Vector<Rental>();

    public Customer(String name) {
        _name = name;
    }
    public String getMovie(Movie movie) {
        Rental rental = new Rental(new Movie("", Movie.NEW_RELEASE), 10);
        Movie m = rental._movie;
        return movie.getTitle();
    }
    public void addRental(Rental arg) {
        _rentals.addElement(arg);
    }
    public String getName() {
        return _name;
    }
```

Fonte: Rahman (2023)

3.3.8 Spring Boot

Segundo a Ibm (2023a), Java *Spring Framework* é uma estrutura popular, de código aberto, de nível empresarial, para criar aplicações autônomas, de produção, que rodam na Máquina Virtual Java (JVM).

Java *Spring Boot* (Figura 18) é uma ferramenta que torna o desenvolvimento de aplicações web e micro serviços com *Spring Framework* mais rápido e fácil através de três capacidades principais:

- Autoconfiguração
- Uma abordagem com opinião sobre a configuração
- A capacidade de criar aplicações autônomas

Além de fornecer todo o aparato necessário para escalar a aplicação, essa ferramenta trabalha com segurança, interceptadores, *cache*, banco de dados, entre outras funcionalidades de modo a executar uma aplicação com um mínimo de configuração possível.

Figura 18 - Exemplo de código com uso do framework Spring Boot

```
public class Greeting {

    private final long id;
    private final String content;

    public Greeting(long id, String content) {
        this.id = id;
        this.content = content;
    }

    public long getId() {
        return id;
    }

    public String getContent() {
        return content;
    }
}
```

```
@RestController
public class GreetingController {

    private static final String template = "Hello, %s!";
    private final AtomicLong counter = new AtomicLong();

    @RequestMapping("/greeting")
    public Greeting greeting(@RequestParam(value="name",
defaultValue="World") String name) {
        return new Greeting(counter.incrementAndGet(),
                            String.format(template, name));
    }
}
```

```
@SpringBootApplication
public class Application {

    public static void main(String[] args) {
        SpringApplication.run(Application.class, args);
    }
}
```

Fonte: Hora (2023)

3.3.9 Python

Segundo a descrição da página do projeto, Python (2023) é uma linguagem de programação interpretada, orientada a objetos, de alto nível com semântica dinâmica (Figura 19). Seu alto nível construído em estruturas de dados, combinado com digitação dinâmica e encadernação dinâmica, o torna muito atraente para o desenvolvimento rápido de aplicações, bem como para uso como uma linguagem de *script*. A sintaxe simples e fácil de aprender do Python enfatiza a legibilidade e reduz o custo de manutenção do programa. Python suporta módulos e pacotes, encorajando a modularidade do programa e a reutilização do código. O intérprete Python e a extensa biblioteca padrão estão disponíveis em formato fonte ou binário sem custo para todas as principais plataformas, e podem ser distribuídos livremente.

Ela também é muito usada nas aplicações de ML e ciências de dados. Suas bibliotecas de ML são pioneiras nesse quesito e como a linguagem desempenha bem, ela também é fácil de usar.

Figura 19 - Exemplo de código em Python

```
class Car:

    def __init__(self, speed=0):
        self.speed = speed
        self.odometer = 0
        self.time = 0

    def say_state(self):
        print(f"I'm going {my_car}kph!".format(self.speed))

    def accelerate(self):
        self.speed += 5
```

Fonte: Jetbrains (2023a)

3.3.10 Pandas

Segundo McKinney (2022), o pandas fornece estruturas de dados de alto nível e funções projetadas para fazer funcionar com dados estruturados ou tabulares intuitivos e flexíveis (Figura 20). Desde seu surgimento em 2010, ele ajudou a permitir que a Python fosse um ambiente poderoso e produtivo de análise de dados.

Figura 20 - Exemplo de código com pandas

```
In [25]: import pandas as pd
         import pandasql as pdsql
         pysql = lambda q: pdsql.sqldf(q, globals())

         mydataframe = pd.DataFrame([[1,2],[3,4],[5,6]], columns=['uid', 'test'])
         pysql("select uid, (test /2) as new_val  from mydataframe;")
```

Out[25]:

	uid	new_val
0	1	1
1	3	2
2	5	3

Fonte: Kervizic (2023)

3.3.11 Scikit-learn

Scikit-learn é uma popular e robusta biblioteca de ML que possui uma vasta gama de algoritmos, bem como ferramentas para visualização ML, pré-processamento, ajuste de modelos, seleção e avaliação (Figura 21).

Com base no NumPy, SciPy e matplotlib, scikit-learn apresenta uma série de algoritmos eficientes para classificação, regressão e agrupamento. Estes incluem suporte a *vector machine*, *rain forests*, *gradient boosting*, *k-means*, and DBSCAN.

Scikit-learn apresenta relativa facilidade de desenvolvimento devido a suas APIs consistentes e eficientes, ampla documentação para a maioria dos algoritmos e numerosos tutoriais online (NVIDIA, 2023).

Figura 21 - Exemplo de código com scikit-learn

```
Out[16]: array([2])

In [17]: X_new = [[3,5,4,2],[5,4,3,2]]
         knn.predict(X_new)

Out[17]: array([2, 1])

In [18]: from sklearn.linear_model import  LogisticRegression
         logreg = LogisticRegression()
         # fit the model
         logreg.fit(X,y)
         #predict the response
         logreg.predict(X_new)

Out[18]: array([2, 0])
```

Fonte: Soppin (2018)

3.3.12 Apache Spark

O Apache Spark é um mecanismo de análise unificado para processamento de dados em larga escala com módulos integrados para SQL, *streaming*, aprendizado de máquina e processamento de gráficos. O Spark pode ser executado no Apache Hadoop, Apache Mesos, Kubernetes, por conta própria, na nuvem e em diversas fontes de dados (CLOUD, 2023).

3.3.13 MongoDB

Segundo a Ibm (2023b), o MongoDB é um sistema de gerenciamento de banco de dados não-relacional de código aberto que utiliza documentos flexíveis ao invés de tabelas e linhas para processar e armazenar várias formas de dados. Como uma solução de banco de dados NoSQL, o MongoDB não requer um sistema de gerenciamento de banco de dados relacional, portanto, ele fornece um modelo elástico de armazenamento de dados que permite aos usuários armazenar e consultar tipos de dados multivariados com facilidade. Isto não apenas simplifica o gerenciamento do banco de dados para desenvolvedores, mas também cria um ambiente altamente escalável para aplicações e serviços multiplataforma.

Conforme destacado na Figura 22, os documentos MongoDB ou coleções de documentos são as unidades básicas de dados. Formatados como *Binary* JSON (Javascript Object Notation), estes documentos podem armazenar vários tipos de dados e ser distribuídos por vários sistemas. Uma vez que o MongoDB emprega um projeto de esquema dinâmico, os usuários têm flexibilidade inigualável ao criar registros de dados, consultar coleções de documentos através da agregação do MongoDB e analisar abundantes informações.

Figura 22 - Exemplo de documento MongoDB

```
{
  "student": {
    "name": "John",
    "class": "Intermediate",
    "address": {
      "street": "2293 Example Street",
      "City": "Chicago",
      "State": "IL"
    }
  }
}
```

Fonte: Mongodb (2023a)

4. ESPECIFICAÇÃO

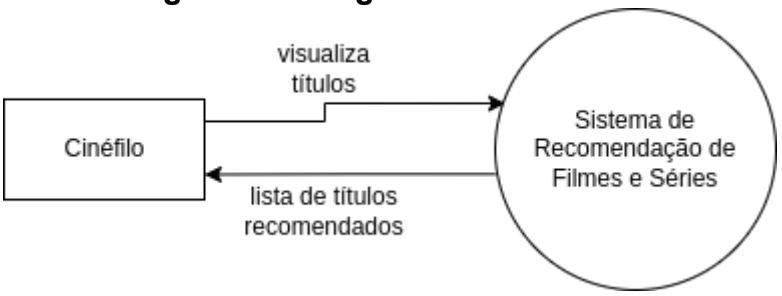
De modo a apresentar melhor a modelagem do projeto, foi desenvolvido o diagrama de fluxo de dados (DFD).

4.1 Diagrama de contexto

O diagrama de contexto é um gráfico, composto por um fluxo de dados que mostra as interfaces entre o projeto e a sua relação com o ambiente em que será desenvolvido (CAMARGO, 2018).

No contexto do trabalho desenvolvido, a Figura 23 representa a interação do usuário com o sistema. O cinéfilo pode solicitar a visualização de títulos e o sistema por sua vez, retorna uma lista de recomendação dos mesmos.

Figura 23 - Diagrama de Contexto



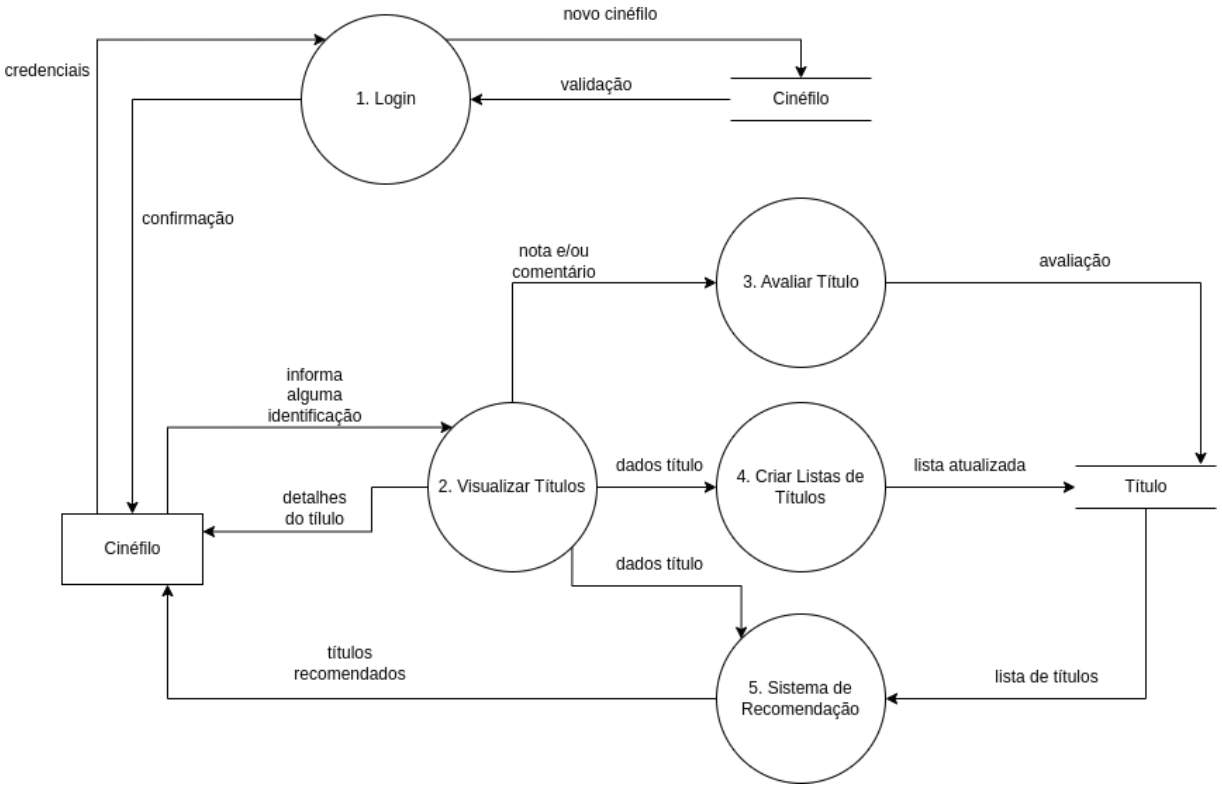
Fonte: Elaborado pela autora

4.2 Diagrama de fluxo de dados Nível 1 - Visão geral detalhada

O DFD nível 1 oferece maiores detalhes de peças do diagrama de contexto. Ele destaca as principais funções desempenhadas pelo sistema enquanto você expande o processo de alto nível do diagrama de contexto em subprocessos (O QUE... 2022).

Na Figura 24, o cinéfilo consegue realizar seu cadastro e login, visualizar os títulos pelo qual consegue avaliá-lo, adicionar na lista “assistir mais tarde” e até mesmo receber recomendações de títulos relacionados dado suas características.

Figura 24 - Diagrama de Fluxo de Dados Nível 1



Fonte: Elaborado pela autora

4.3 Diagrama de fluxo de dados Nível 2 - 4 Criar Listas de Títulos

O DFD nível 2 aprofunda ainda mais partes do nível 1. Pode ser preciso mais texto para chegar ao nível necessário de detalhes em relação ao funcionamento do sistema (O QUE... 2022).

Na Figura 25, está detalhado a quarta funcionalidade de criação de listas de títulos. Esse processo consiste em: iniciar a criação da lista ao definir uma temática (título e descrição), selecionar os títulos interessados e, por fim, configurar o modo de visualização desta lista (se qualquer pessoa pode ver ou somente a pessoa autora da lista).

Figura 25 - Diagrama de Fluxo de Dados Nível 2

Fonte: Elaborado pela autora

5. DESENVOLVIMENTO

Neste trabalho, foi desenvolvido uma aplicação *web* que possui alguns elementos de uma rede social, denominada *WatchToWatch*, voltada para telespectadores de filmes e séries. A aplicação permite que o usuário visualize o catálogo de títulos, realize comentários, veja recomendações, receba recomendações personalizadas dado o histórico de títulos marcados como assistidos, crie listas e caso disponível, contatar outras pessoas por meio da rede social fornecida. O sistema de recomendação fornecido pela aplicação possibilita que os usuários encontrem conteúdo de interesse rapidamente.

Apesar de a aplicação ser desenvolvida em português, alguns elementos visuais do título selecionado, como sinopse e elenco, podem estar em inglês para garantir que a base de dados tenha um tamanho considerável devido a limitações de dados encontrados em português pelo serviço utilizado conforme mencionado na seção 3.2. Mais limitações são descritas na seção 5.4.

As tabelas de banco de dados utilizadas nesta aplicação foram modeladas em documentos que possuem uma estrutura dinâmica.

Essa aplicação foi desenvolvida utilizando o *framework* Spring Boot com a linguagem Java e o sistema gerenciador de banco não relacional MongoDB para a parte do servidor. Já o *framework* Vue.js, o Tailwind CSS, o servidor Node.js e além do HTML, CSS e Javascript para a parte do cliente.

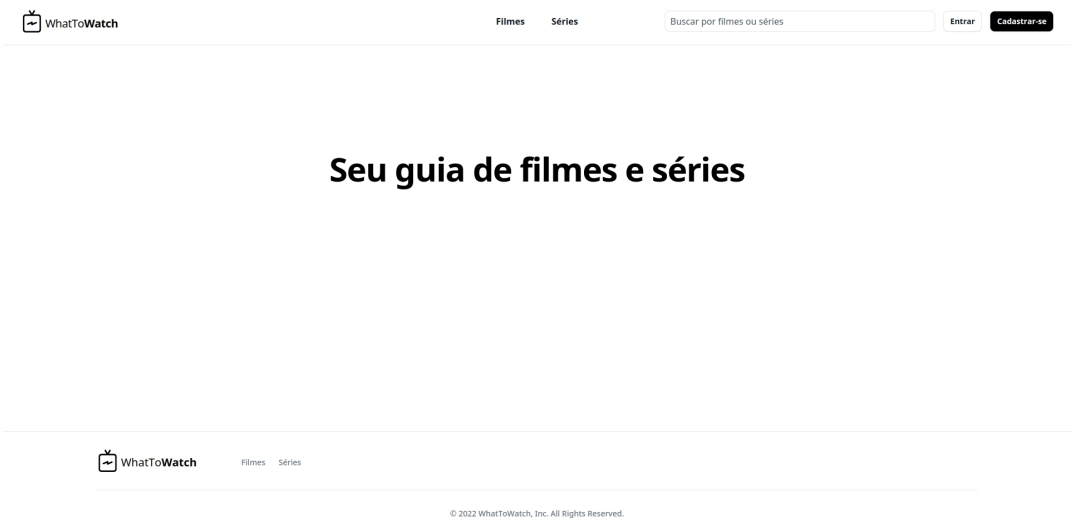
5.1 Detalhamento da aplicação

Nessa seção, serão apresentadas as funcionalidades da aplicação.

5.1.1 Página inicial

A primeira página que o usuário poderá interagir é a inicial (Figura 26). Além de conter o *slogan* da aplicação, nela é possível acessar as páginas de filmes, séries, de login (para entrar na conta) e de cadastro.

Figura 26 - Página inicial



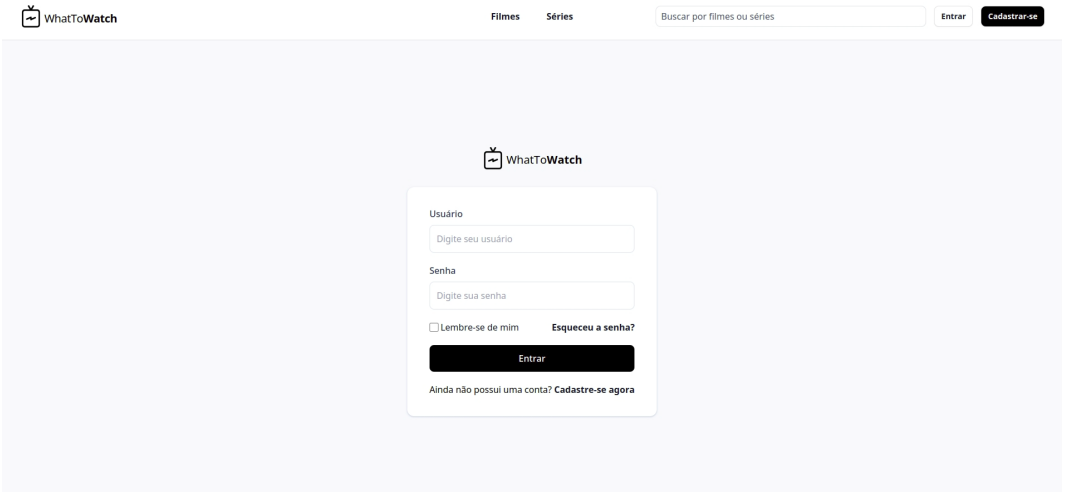
Fonte: Elaborado pela autora

5.1.2 Cadastro e *login*

Na página de cadastro (Figura 28), o usuário deverá informar as informações mínimas e obrigatórias para conseguir criar uma conta e interagir com funcionalidades específicas do sistema.

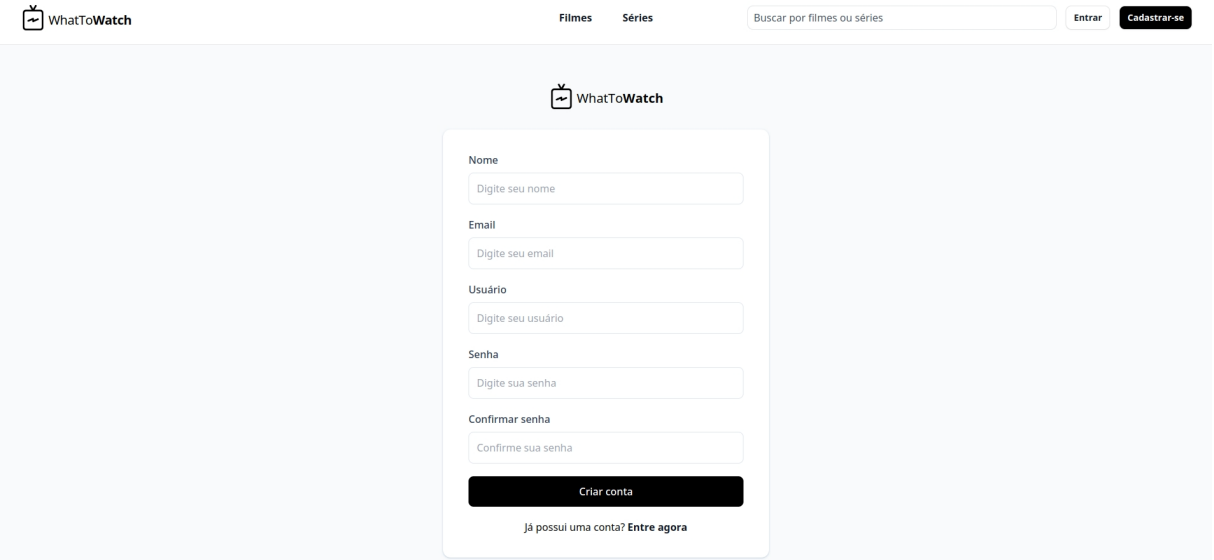
Na página de *login* (Figura 27), só é necessário informar o usuário e a senha para entrar em sua conta.

Figura 27 - Página de *login*/entrar na conta



Fonte: Elaborado pela autora

Figura 28 - Página de cadastro

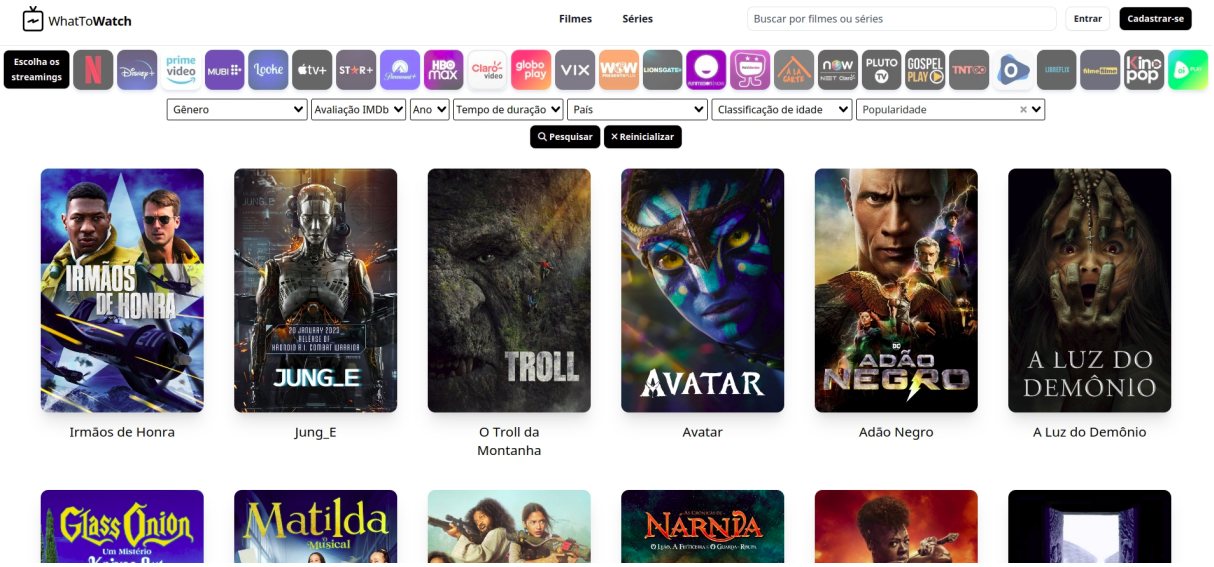


Fonte: Elaborado pela autora

5.1.3 Filmes

Quando o usuário acessa a página de filmes (Figura 29), é possível realizar três funções: a pesquisa e filtragem de títulos, dado características interessantes, e paginar sem nenhuma pretensão os títulos disponíveis. Na seção 5.1.5 está detalhado como é realizada a pesquisa e filtragem dos títulos.

Figura 29 - Página de filmes



Fonte: Elaborado pela autora

5.1.3.1 Visualização de um filme

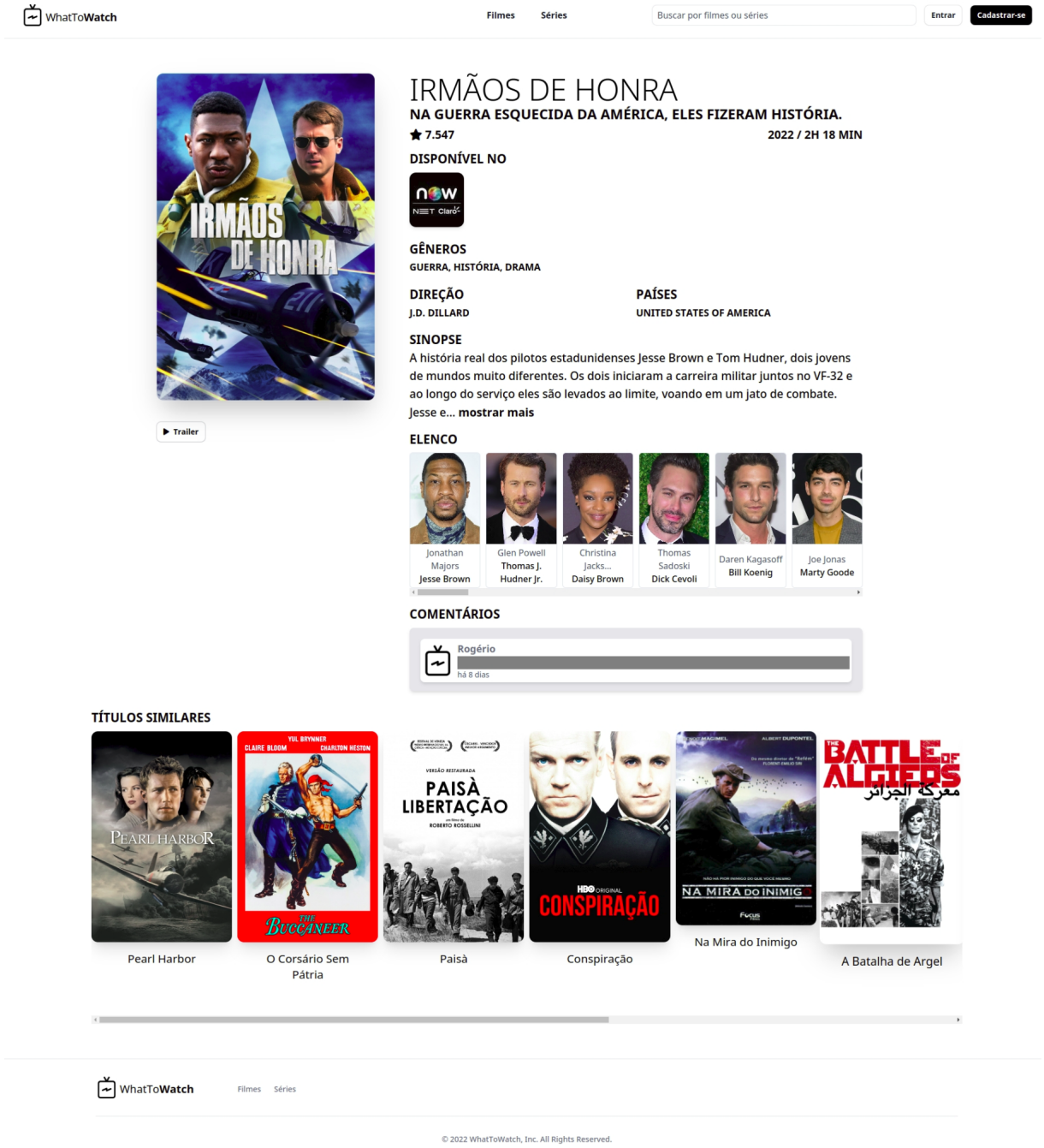
Quando um filme é selecionado, o usuário encontrará informações detalhadas sobre o mesmo, assim como, recomendações baseada no seu conteúdo (Figura 30). Conforme citado na seção 2.9.1.1 e 5.2.

Das informações detalhadas, é possível saber a avaliação geral do título, em que ano foi lançado, sua duração, em quais serviços de *streaming* ele está disponível, quais são os gêneros da obra, a classificação de idade, quem é ou são as pessoas diretoras, a sinopse e o elenco do filme.

Como forma de interação com a página, o usuário poderá visualizar, caso disponível, o trailer do filme e os comentários de outras pessoas. Caso o usuário esteja conectado em sua conta, será possível marcar o título como visto e selecionar a avaliação desejada: de 1 a 10. Por padrão, foi definido que a avaliação será sempre 0.

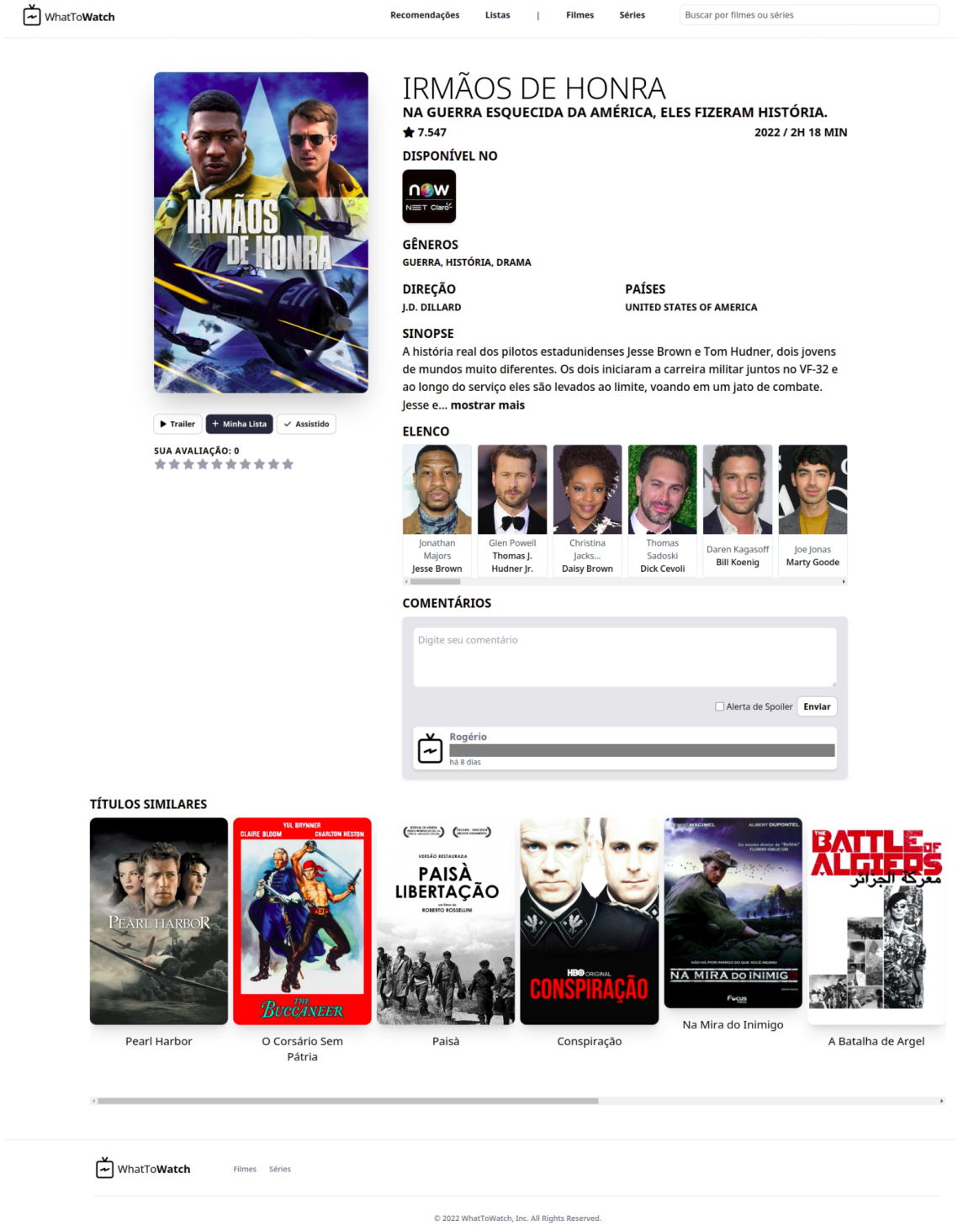
Outras funcionalidades disponíveis para o usuário conectado (Figura 31) é a possibilidade de realizar os comentários do filme selecionado e adicioná-lo na lista padrão de assistir mais tarde.

Figura 30 - Visualização do filme pelo usuário visitante



Fonte: Elaborado pela autora

Figura 31 - Visualização do filme pelo usuário conectado em sua conta

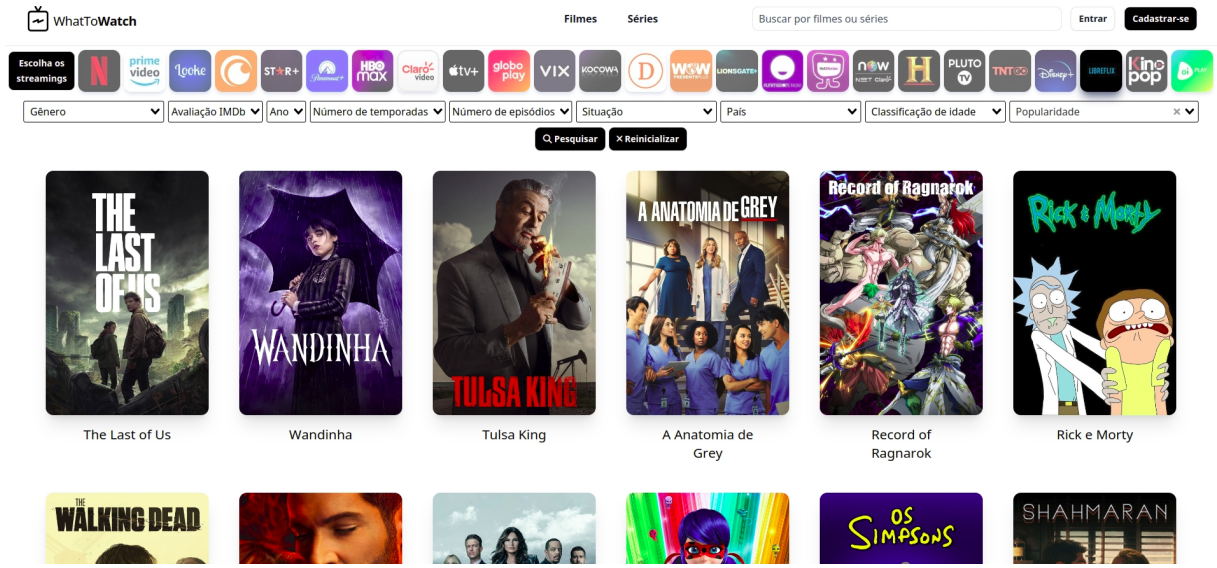


Fonte: Elaborado pela autora

5.1.4 Séries

A página de séries (Figura 32) é similar à de filmes, pois é possível realizar três funções: pesquisar e filtrar títulos dado às características de interesse, e paginar sem nenhuma pretensão quais são os títulos disponíveis. Na seção 5.1.5 está detalhado o modo de realização da pesquisa e filtragem dos títulos.

Figura 32 - Página de séries



Fonte: Elaborado pela autora

5.1.4.1 Visualização de uma série

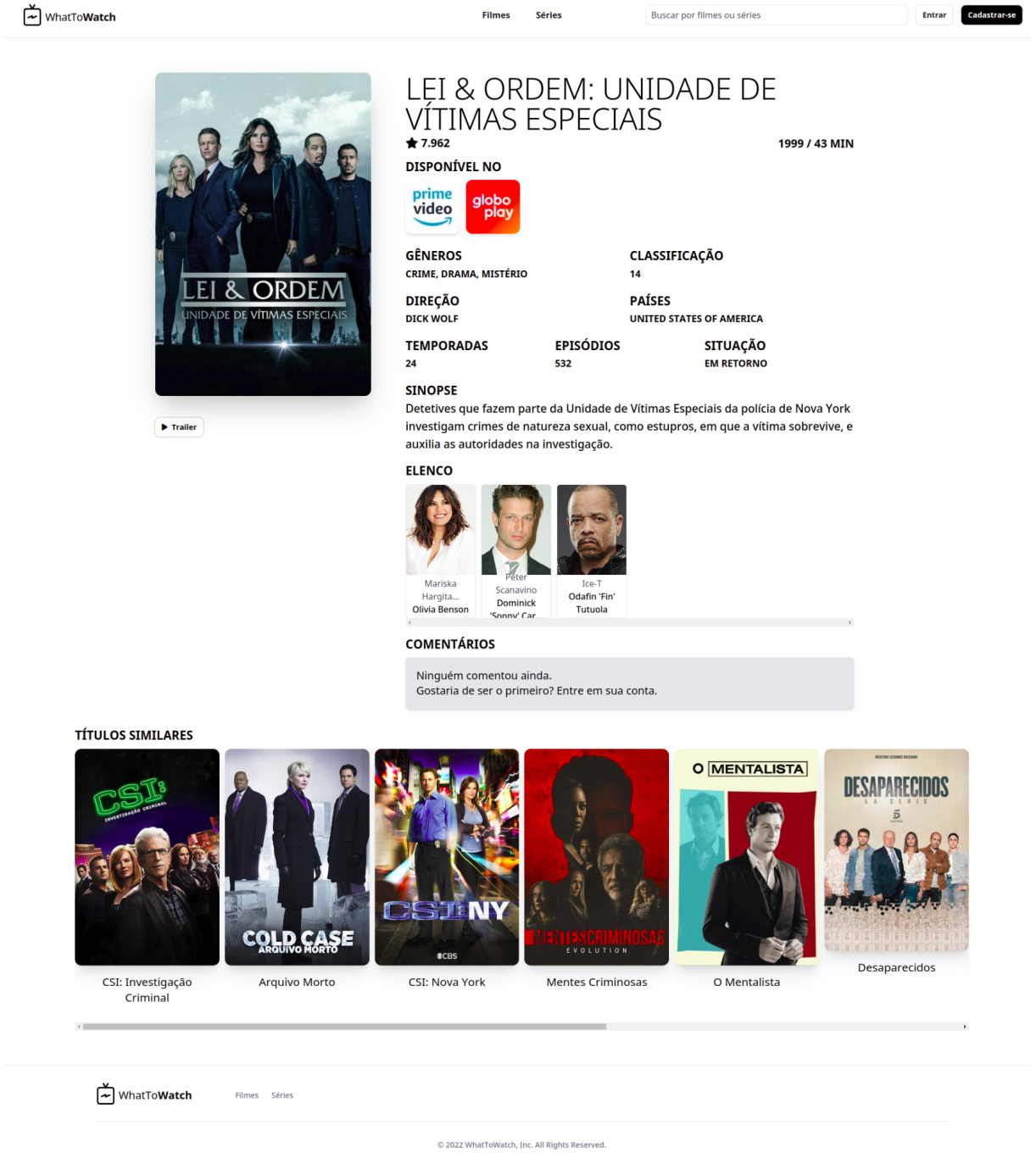
Similar a visualização de filmes, no caso das séries (Figura 33), o usuário encontrará informações detalhadas sobre as mesmas, assim como, recomendações baseadas em seus conteúdos. Conforme citado na seção 2.10.4 e 5.2.

Através das informações detalhadas, é possível saber a avaliação geral do título, em que ano foi lançado, sua duração, em quais serviços de *streaming* ela está disponível, quais são os gêneros da obra, a classificação de idade, quem é ou são as pessoas diretoras, a sinopse, o elenco da série e quais são as temporadas disponíveis.

Como forma de interação com a página, o usuário poderá visualizar, caso disponível, o trailer da série e os comentários de outras pessoas. Caso o usuário esteja conectado em sua conta, será possível a essa pessoa marcar o título como visto e selecionar a avaliação desejada: de 1 a 10. Por padrão, foi definido que a avaliação será sempre 0.

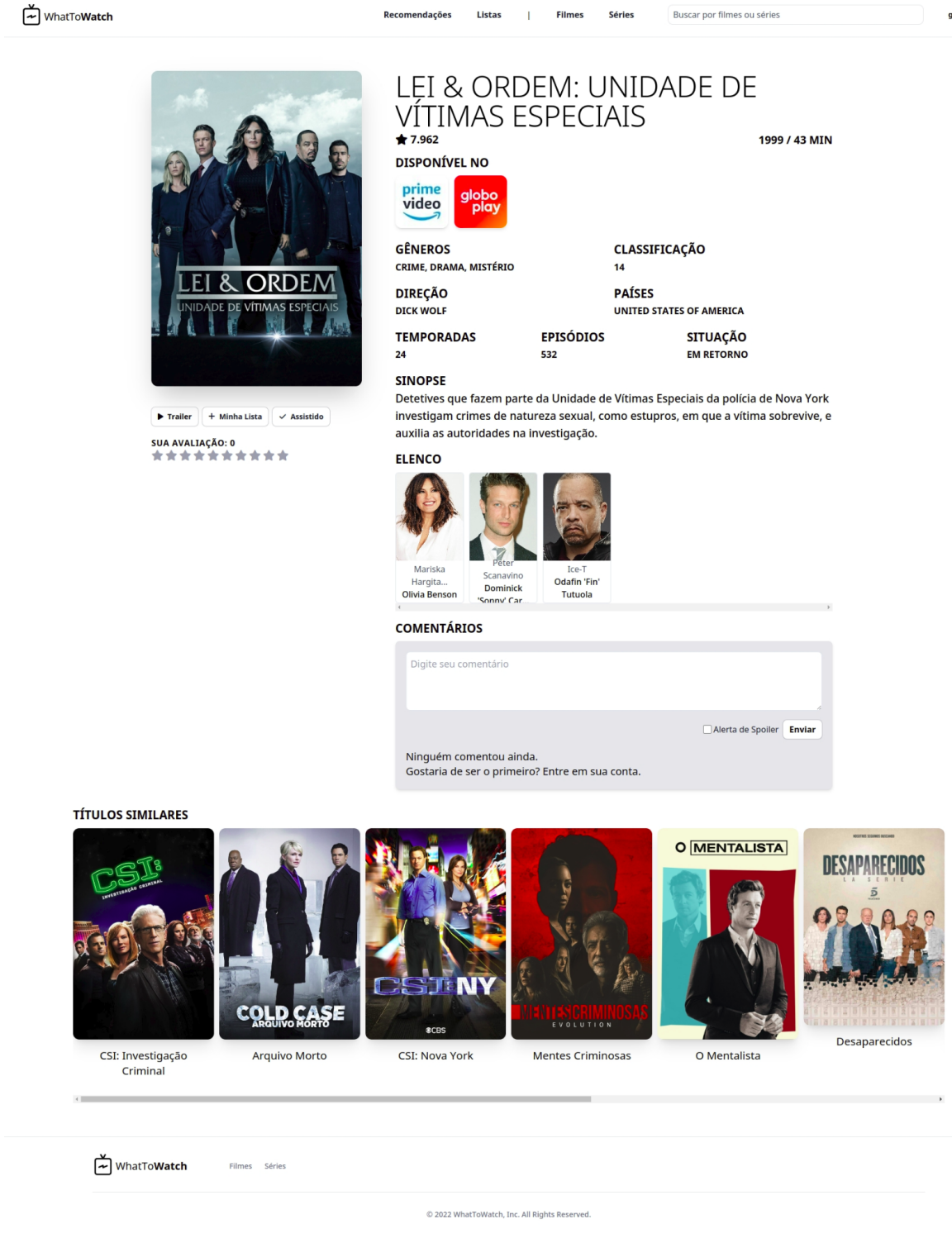
Outras funcionalidades disponíveis para o usuário conectado (Figura 34): a possibilidade de realizar os comentários da série selecionada e adicioná-la na lista padrão de assistir mais tarde.

Figura 33 - Visualização da série pelo usuário visitante



Fonte: Elaborado pela autora

Figura 34 - Visualização da série pelo usuário conectado em sua conta

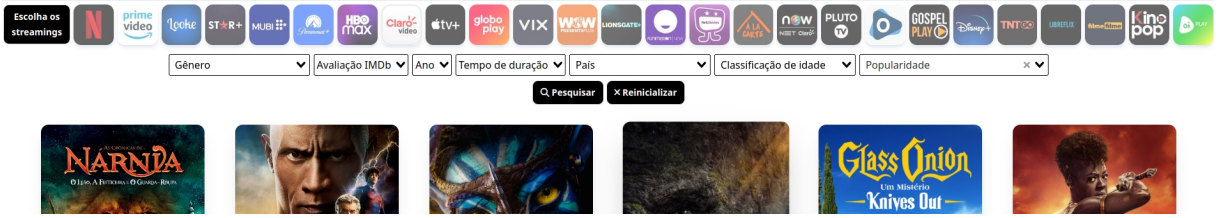


Fonte: Elaborado pela autora

5.1.5 Pesquisa

A pesquisa ajuda a encontrar um título de interesse e ela pode ser realizada em duas formas: pela página de títulos (Figura 35) e pela a barra de navegação (Figura 36).

Figura 35 - Funcionalidade de pesquisa e seus filtros na página de filmes



Fonte: Elaborado pela autora

Figura 36 - Funcionalidade de pesquisa na barra de navegação

Fonte: Elaborado pela autora

A pesquisa na página de títulos, na verdade é uma aplicação de filtros. Os quais são: filtrar por serviço de *streaming* (Quadro 1 e 2), gêneros (Quadro 3 e 4), avaliação IMDb, ano de lançamento, tempo de duração, país de lançamento, classificação de idade (Quadro 5), com a possibilidade em realizar ordenação por títulos populares, recém lançados, melhores avaliados e por ordem alfabética.

Quadro 1 - Serviços de *streaming* disponíveis para filtragem de filmes

Netflix	Disney Plus	Amazon Prime Video	Star Plus	Claro video
Looke	Paramount Plus	HBO Max	Apple TV Plus	Globoplay
Funimation Now	MUBI	NetMovies	Belas Artes à La Carte	VIX
NOW	Oldflix	GOSPEL PLAY	TNTGo	Libreflix
WOW Presents Plus	Filme Filme	KinoPop	Oi Play	Pluto TV
Lionsgate Plus				

Fonte: Elaborado pela autora

Quadro 2 - Serviços de *streaming* disponíveis para filtragem de séries

Netflix	Disney Plus	Amazon Prime Video	Star Plus	Claro video
Looke	Paramount Plus	HBO Max	Apple TV Plus	Globoplay
Funimation Now	Kocowa	NetMovies	Crunchyroll	VIX
NOW	DOCSVILLE	History Play	TNTGo	Libreflix
WOW Presents Plus	Filme Filme	KinoPop	Oi Play	Pluto TV
Lionsgate Plus				

Fonte: Elaborado pela autora

Quadro 3 - Gêneros disponíveis para filtragem de filmes

Ficção científica	Comédia	Guerra	Cinema TV	Aventura	Romance
História	Documentário	Família	Terror	Ação	Drama
Faroeste	Thriller	Animação	Crime	Mistério	Música
Fantasia					

Fonte: Elaborado pela autora

Quadro 4 - Gêneros disponíveis para filtragem de séries

Comédia	Sci-Fi & Fantasy	Romance	História	Talk	Soap
Família	Kids	Documentário	Reality	Faroeste	Drama
Animação	War & Politics	Crime	News	Mistério	Action & Adventure

Fonte: Elaborado pela autora

Quadro 5 - Classificação de idade disponível para filtragem de filmes e séries

Livre	10 anos	12 anos	14 anos	16 anos	18 anos

Fonte: Elaborado pela autora

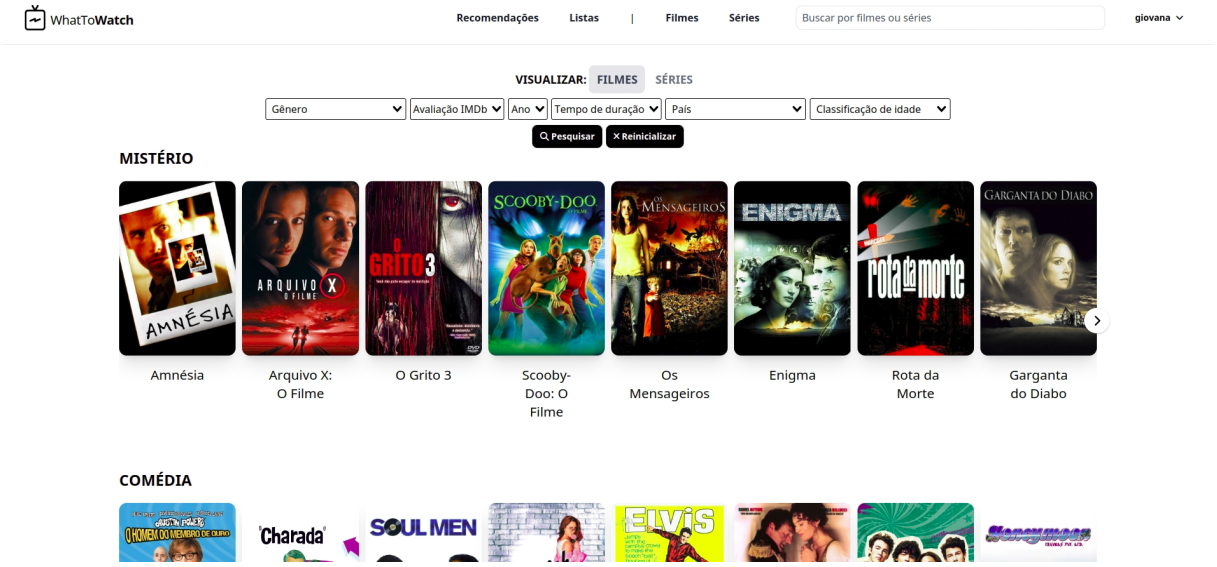
Já a pesquisa pela a barra de navegação é a maneira mais direta de se encontrar um título de interesse, porque nela se realiza a pesquisa por meio do nome do título. Esse nome pode ser informado tanto em português quanto em inglês.

5.1.6 Recomendações personalizadas

Existe uma página para visualizar as recomendações personalizadas (Figura 37) que consiste na execução do algoritmo de recomendação colaborativa - visto na seção 2.9.2.2 e 2.10.4.

Essa página consiste basicamente em listar as recomendações geradas por gênero com a possibilidade da aplicação de alguns filtros mencionados na seção 5.1.5.

Figura 37 - Recomendações personalizadas para o usuário conectado



Fonte: Elaborado pela autora

5.1.7 Listas

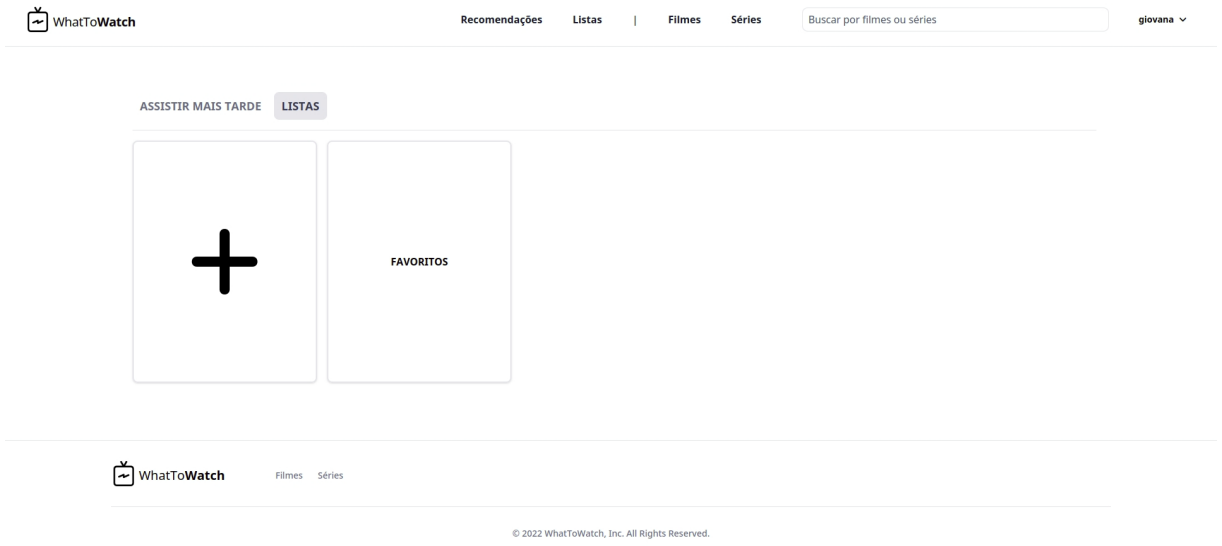
A funcionalidade de listas está disponível somente para o usuário conectado, por meio de uma página própria (Figura 38). Nesta página, o usuário poderá visualizar o conteúdo da lista padrão “assistir mais tarde”, separada em filmes e séries.

Outra possibilidade é o usuário criar listas personalizadas. Para isso, é necessário clicar no símbolo (+) de adicionar. Depois é necessário informar um título e, caso necessário, uma descrição. Para adicionar títulos a lista criada, também é preciso informar na caixa de pesquisa o nome do título. Se for encontrado algo, é necessário selecionar a linha apresentada e o título aparecerá na lista. Para excluir um item, é preciso levar a seta do mouse em cima do título e clicar para removê-lo.

Isso quando a lista está no modo edição, se for o modo visualização, clicar no título só abrirá sua página de detalhes.

Outro ponto interessante, é a possibilidade de ser controlado o modo de exibição da lista personalizada em: pública ou privada. Com esses modos de exibição a lista pode ser visualizada na página de perfil do usuário - seção 5.1.8.

Figura 38 - Visualização das listas



Fonte: Elaborado pela autora

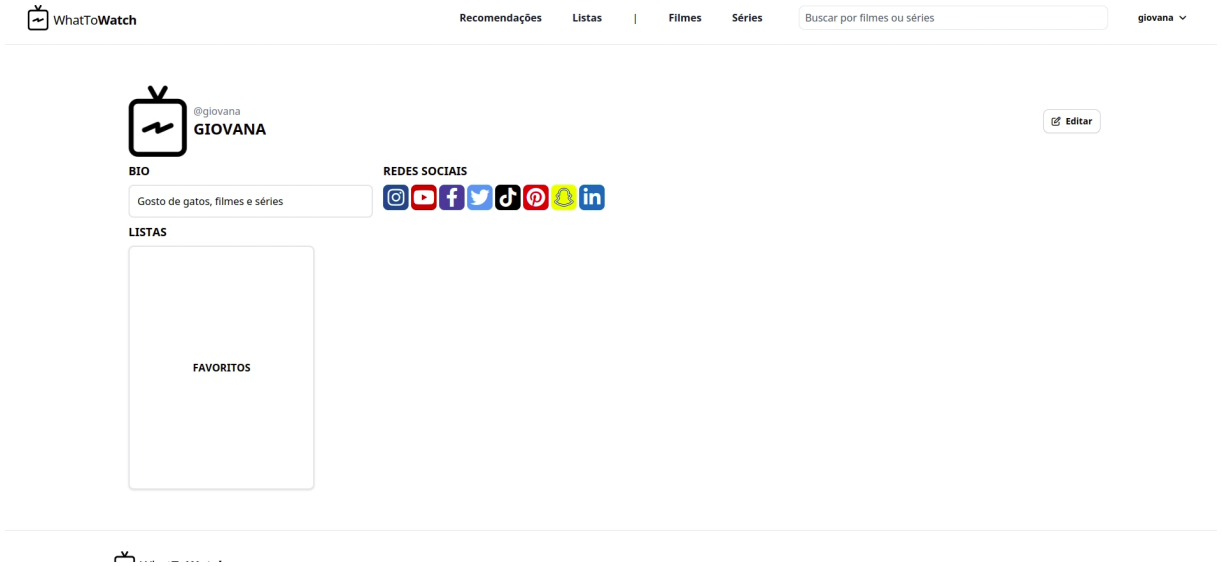
5.1.8 Perfil

A página de perfil também só está disponível para os usuários conectados. Nela existem dois modos de interação: visualização e edição.

No modo visualização (Figura 39), qualquer outro usuário poderá ver o nome da pessoa, nome de usuário, as redes sociais disponíveis, uma descrição e também, se disponível, suas listas.

No modo de edição, o usuário poderá informar suas redes sociais e uma descrição sobre si mesmo.

Figura 39 - Modo visualização do perfil do usuário



Fonte: Elaborado pela autora

5.2 Sistema de recomendação

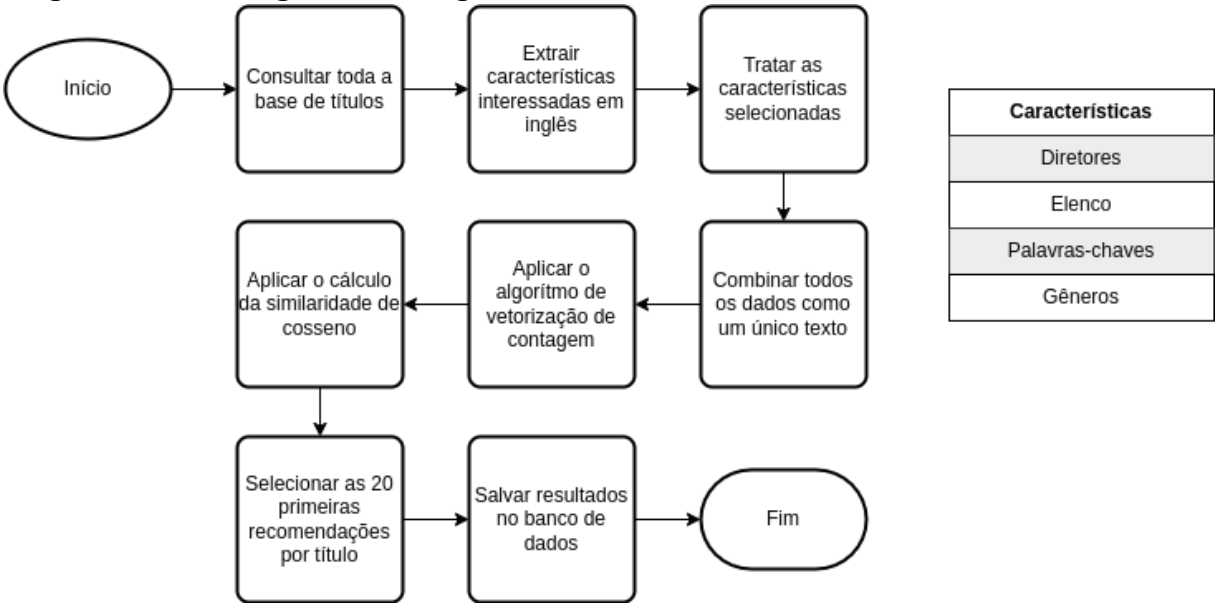
O sistema desenvolvido possui um sistema de recomendação que faz o uso de recomendações não personalizadas e gera dois tipos diferentes de recomendação: baseada no conteúdo e a colaborativa.

Para a recomendação não personalizada, existem quatro tipos: títulos populares, recém lançados, os melhores avaliados e alfabético. Os títulos populares não precisavam ser gerados porque a informação está disponível pelo serviço de terceiros (seção 3.2). Logo, foi necessário ordenar os títulos que possuem a maior classificação de popularidade. Os títulos recém lançados consistem em ordenar os mesmos pela data de lançamento de maneira decrescente. Para os títulos melhores avaliados, também consiste na ordenação dos mesmos de maneira decrescente (maior nota para menor) e, por fim, a última recomendação consiste em trazer os títulos ordenados de maneira alfabética. As recomendações não personalizadas estão disponíveis na página de visualização geral de filmes e séries, respectivamente, as seções 5.1.3 e 5.1.4.

Para as recomendações baseada no conteúdo (Figura 40) foi utilizado a primeira abordagem (item 2.9.1.1). Com ela, o usuário visitante (que não possui conta) já consegue ter acesso a algumas recomendações, mesmo que não considere seu histórico de interações. O algoritmo aplicado foi o de vetorização de contagem (seção 2.10.1) e teve como critério de recomendação ao utilizar os

atributos tanto para filmes e séries: as palavras-chave, as cinco primeiras pessoas do elenco, a(s) pessoa(s) diretora(s) e os gêneros. Essas recomendações estão disponíveis na página do título selecionado (seções 5.1.3.1 e 5.1.4.1).

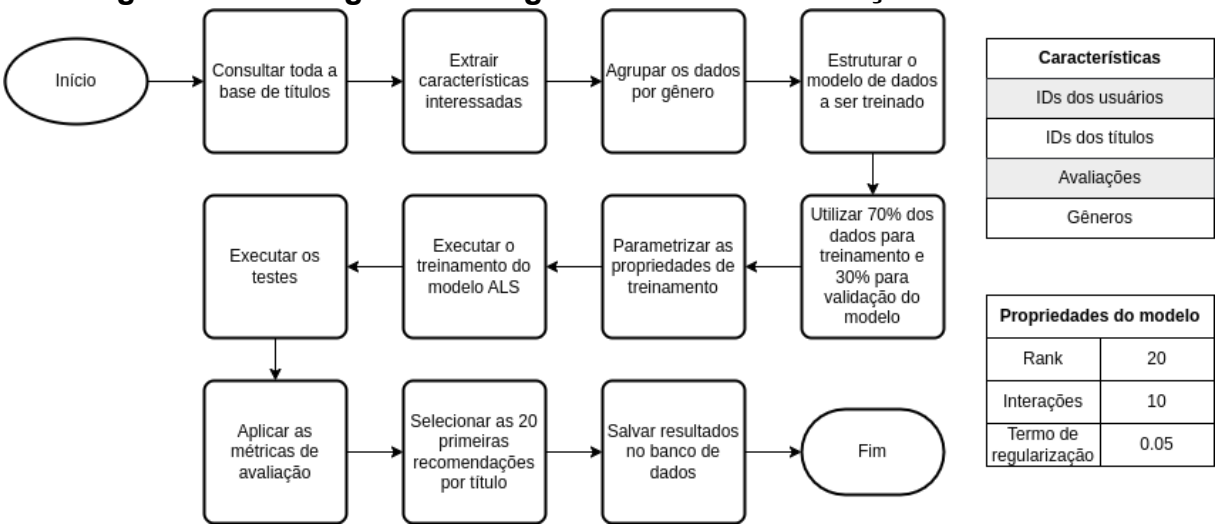
Figura 40 - Fluxograma do algoritmo de recomendação baseada no conteúdo



Fonte: Elaborado pela autora

Para a recomendação colaborativa (Figura 41) foi utilizada a baseada em modelo (seção 2.9.2.2). Com o uso do modelo, é possível identificar tendências de comportamento dos usuários para prever as avaliações não informadas ou não existentes. O único comportamento utilizado na recomendação é o simples ato de marcar o título como visto ou não visto e sua avaliação. O algoritmo aplicado foi o dos mínimos quadrados alternados (seção 2.10.4) o qual só faz o uso dos dados: identificação do usuário, identificação do título e a avaliação. Outras informações também utilizadas são as propriedades de configuração do modelo: *rank* ou ranqueamento, número de interações e o termo de regularização. A propriedade ranqueamento, representa o número de características que ajudam a explicar as tendências de comportamentos dos usuários dados suas escolhas de quais títulos avaliam e como foi essa nota. O número de interações representa quantas vezes o algoritmo será executado. Por fim, o termo de regularização é usado para evitar a situação de sobre ajustamento do modelo (seção 2.10.3). Essas recomendações estão disponíveis na página de recomendações do usuário somente quando o mesmo está conectado em sua conta (seção 5.1.6).

Figura 41 - Fluxograma do algoritmo de recomendação colaborativa



Fonte: Elaborado pela autora

A execução das recomendações baseada no conteúdo e a colaborativa ocorrem em lote (seção 2.7.1 e 5.4), após os usuários utilizarem o sistema por um tempo. Em ambiente de produção, é necessário que as recomendações sejam executadas pelo menos uma vez ao dia, mas se houver momentos em que os recursos computacionais estiverem ociosos, é possível executar as recomendações mais de uma vez.

Tecnicamente, as recomendações são executadas em um serviço a parte desenvolvido em Python, com bibliotecas e frameworks relacionadas ao ramo de ML.

Apesar de ter sido implementado duas técnicas de recomendação, elas são executadas linearmente (uma após a outra) e por categoria de título. No caso, primeiro são executadas as recomendações baseadas no conteúdo e colaborativa dos filmes e depois das séries. Outro critério adotado é na utilização dos dados em inglês para gerar as recomendações. Então para o usuário, ele verá a maioria das informações em português, mas para gerar as recomendações ficou diferente para deixar as análises dos dados homogênea (seção 5.4).

O processo para gerar as recomendações baseada em conteúdo, consiste em:

1. Consultar toda a base de dados do título;
2. Aplicar um tratamento nos dados a serem utilizados (organização e limpeza), conforme os critérios de seleção dos atributos citados anteriormente;

3. Aplicar o algoritmo vetorização de contagem, ao parametrizar a desconsideração de artigos e conjunções;
4. Calcular o grau de similaridade entre os itens da matriz esparsa gerada com a similaridade de cosseno (seção 2.11.1);
5. Obter os 20 primeiros resultados do título de iteração;
6. Por fim, salvar todas as recomendações em sua respectiva coleção no banco de dados não relacional.

A respeito do processo de geração de recomendações colaborativas, ele se dá por:

1. Consultar toda a base de dados do título;
2. Aplicar um tratamento nos dados a serem utilizados (organização e limpeza), conforme os critérios de seleção dos atributos citados anteriormente;
    a. Os títulos são agrupados por gêneros para facilitar a visualização dos resultados na página de recomendações personalizadas do usuário;
3. Aplicar o algoritmo dos mínimos quadrados alternados ao usar uma implementação pronta da ferramenta Apache Spark (seção 3.3.12);
    a. Os dados são separados em 70% para gerar as previsões e 30% para realizar os testes do modelo;
    b. É possível parametrizar a quantidade de fatores latentes a serem utilizados, o número de iterações que o modelo deve executar, o termo de regularização para evitar o sobreajustamento (seção 2.10.3) e se será considerado as preferências implícitas dos títulos e usuários;
4. Calcular a raiz do erro quadrático médio e o coeficiente de determinação das previsões;
5. Obter os 20 primeiros resultados do título de iteração;
6. Por fim, salvar todas as recomendações em sua respectiva coleção no banco de dados relacional.

5.3 Testes realizados

Nesta seção, serão apresentados os testes e resultados obtidos dos sistemas de recomendação implementados.

5.3.1 Recomendação baseada em conteúdo

Dado as argumentações apresentadas sobre as escolhas do sistema na seção 5.2, como consequência surgiu um problema: não é possível verificar o desempenho do algoritmo. Ao revisar a literatura sobre o assunto (seção 2.9.1), percebeu-se que a utilização de alguma métrica só seria usual se gerar-se um histórico de interação do usuário com o sistema (seção 2.9.1.2). Essa segunda abordagem é similar a recomendação colaborativa, contudo, nesse caso, é considerado as características dos títulos. A decisão em utilizar a primeira abordagem foi mantida para não obrigar a criação da conta. Essa decisão visa despertar o interesse dos possíveis usuários que usariam o sistema de maneira recorrente.

Entretanto, é possível fazer uma simples análise para ter noção dos resultados das recomendações. Por exemplo, das 11.700 recomendações realizadas para os filmes, 10,07% delas possuem o grau de similaridade de pelo menos 50%. Agora, das 4.319 recomendações realizadas para as séries, 58,74% delas possuem o grau de similaridade de pelo menos 50%.

5.3.2 Recomendação colaborativa

Para viabilizar a geração das recomendações colaborativas, foi desenvolvido uma automação em Java para gerar o histórico de interações do usuário com os títulos. Essa automação consiste em:

1. Obter todos os usuários cadastrados (no caso, 10);
2. Obter 1000 títulos (o algoritmo executa duas vezes, para gerar o histórico de filmes e séries);
    a. 500 títulos serão selecionados aleatoriamente;
    b. Para a outra metade, será gerado tendências de consumo dos filmes e séries;

   i. Essa tendência consiste em selecionar aleatoriamente três gêneros para depois consultar os títulos filtrados na base de dados. O retorno consiste numa seleção também aleatória caso retorne mais de 500 itens;
3. Para ⅓ dos títulos selecionados, os mesmos são marcados como vistos, mas sem uma avaliação escolhida. No caso, o valor padrão para ela é 0.
   a. Essa configuração ajuda simular o comportamento de um usuário real ao gerar a famosa matriz esparsa das avaliações;
4. Para os ⅔ restantes, aleatoriamente o algoritmo escolherá se avalia um título com a nota ou não;
   a. Logo, cinco possíveis cenários predominantes ocorrerão para cada usuário:
      i. os títulos poderão não ter uma avaliação definida;
      ii. os títulos terão uma avaliação definida, mas em sua maioria com notas menores que 5;
      iii. os títulos terão uma avaliação definida, mas em sua maioria com notas maiores que 5;
      iv. as avaliações dos títulos será equilibrada;
      v. a distribuição de títulos avaliados e não avaliados será equilibrada;
   b. Quando o título é avaliado, aleatoriamente sua nota pode ocorrer em três formas:
      i. avaliação entre 1 e 10;
      ii. avaliação entre 1 e 5;
      iii. avaliação entre 5 e 10
5. Por fim, os resultados são salvos em uma coleção no banco de dados não relacional.

Conforme descrito na seção 5.2, o algoritmo da recomendação colaborativa faz o uso de um modelo onde ao final de sua execução são aplicadas as métricas de avaliação: raiz do erro quadrático médio (RMSE) e o coeficiente de determinação (R^2). Como o modelo foi aplicado aos dados agrupados por gênero, foram geradas várias avaliações demonstradas nas Tabelas 1 e 2.

Tabela 1 - Avaliação dos filmes agrupados por gêneros

Gêneros	**RMSE**	**R^2**
Aventura	3.87	-0.44
Animação	3.50	-0.33
Comédia	3.33	-0.21
Mistério	3.85	-0.38
Ficção científica	3.89	-0.62
Música	4.58	-0.76
Thriller	3.94	-0.51
Drama	3.51	-0.33
Ação	3.50	-0.31
Crime	2.33	-0.16
Terror	2.85	-0.57
Fantasia	3.23	-0.20
Romance	3.50	-0.41
Família	3.71	-0.43
Faroeste	4.59	-0.75
História	1.57	-0.28
Documentário	3.46	-0.36
Guerra	3.85	-0.46
Cinema TV	3.88	-0.34

Fonte: Elaborado pela autora

Tabela 2 - Avaliação das séries agrupadas por gêneros

Gêneros	**RMSE**	**R^2**
Kids	2.66	-0.20
Crime	3.22	-0.28
Animação	3.64	-0.32
Documentário	2.91	-0.30
Reality	3.06	-0.22
Action & Adventure	3.21	-0.24
Sci-Fi & Fantasy	3.26	-0.23
Comédia	3.27	-0.30
Drama	3.24	-0.29
Família	3.33	-0.37
Mistério	3.48	-0.36
Faroeste	2.09	-0.19
Soap	3.09	-0.41
War & Politics	5.31	-0.99
Talk	2.80	-0.71

Fonte: Elaborado pela autora

Dado as definições das métricas na seção 2.12, percebe-se que existe espaço para otimizar a parametrização do modelo para melhorar o desempenho do mesmo. Contudo, como os resultados não tendem muito para o intervalo infinito, eles podem ser aceitos como um sistema de recomendação minimamente funcional. Outra coisa que pode ajudar a melhorar o desempenho do algoritmo está na desconsideração do agrupamento dos gêneros dos títulos.

5.4 Limitações do trabalho

Durante o desenvolvimento da aplicação foram encontrados alguns problemas que fizeram com que limitasse o escopo de funcionamento da mesma. Sendo elas:

1. Necessário trabalhar com recomendação em lote

Quando foi realizado o protótipo a fim de viabilizar o desenvolvimento do trabalho final com as tecnologias usadas, foi percebido a lentidão no desempenho do sistema. Antes somente havia implementado a recomendação em tempo real baseada no conteúdo dos títulos e para executar o algoritmo e retornar seus resultados na página de detalhes do mesmo, estava demorando em média 30 segundos para carregar a página completamente. Na época, havia somente 100 filmes e séries na base de dados.

Ao buscar alternativas, foi constatado que a recomendação em tempo real é muito custosa financeiramente e também em relação aos recursos computacionais empregados.

Por exemplo, no estudo de Hazem, Awad e Yousef (2022), eles fizeram um sistema de recomendação em tempo real de maneira distribuída. No quesito de infraestrutura, eles subiram uma arquitetura de sistema que combina vários computadores para trabalharem em conjunto. Cada computador dessa arquitetura pode ser representado por um nó (ponto de conexão) e, no caso, eles utilizaram 3. Cada nó possuía 96 núcleos de processadores (ou 3 chips de processadores) e em cada núcleo foi executada a recomendação sob 2.3 GHz (representa a velocidade de um processador) com 30 GB de memória RAM.

Já em outro estudo, o de Pereira *et al.* (2014), o processo foi um pouco diferente. Eles fizeram a recomendação colaborativa em tempo real, mas com a arquitetura do projeto na nuvem - usaram no caso, o serviço conhecido como AWS.

Ao usar a mesma analogia dos nós, no quesito de infraestrutura eles usaram 55. No que tange a banco de dados, eles usaram dois servidores, o de persistência consumia 25 GB de memória RAM, já o que salva as recomendações consumia 20 GB.

Dado os recursos citados e outros necessários para gerenciar tudo isso, eles tiveram como gasto de aproximadamente $15.000 dólares ou R$ 78.150 reais mensais na cotação de 10 jan. 2023.

A título de comparação, a máquina pessoal utilizada para executar o sistema de recomendação citado na seção 5.2, possui como especificação técnica conforme a Tabela 3.

Tabela 3 - Especificação técnica da máquina pessoal

Modelo da máquina	Acer Nitro AN515-51
Memória RAM	32 GB
Processador	Intel® Core™ i5-7300HQ com 4 núcleos
Frequência baseada em processador	2.50 GHz

Fonte: Elaborado pela autora

Com base nos argumentos apresentados, foi decidido trabalhar com a recomendação em lote para retirar essa carga extra de lentidão do sistema e evitar ter custo financeiro.

2. Uso do banco de dados não relacional

Outro problema que pode estar relacionado com o item anterior é com a base de dados. De início, foi usado o PostgreSQL que é um banco de dados relacional e como o sistema foi divido em várias entidades, quando se abria a página de detalhes do título selecionado, o banco demorava uns 15 segundos para puxar todos os relacionamentos a fim de obter os dados necessários.

Por essa razão, trocou-se o banco para dados não relacionais a fim de diminuir essa quebra de entidades para trazer todos os dados necessários em poucas consultas. Essa troca melhorou o desempenho também na execução dos algoritmos de recomendação.

3. Qualidade dos dados obtidos pelo serviço de terceiro

Mais um problema encontrado foi com a qualidade dos dados obtidos pelo serviço de terceiros citado na seção 3.2.

A base de dados desse serviço é populada pela comunidade e a tradução das informações utilizadas muitas vezes estão incompletas ou nem existem em português. Como foi adotado um critério de utilização desses dados (na seção 3.2 e 5.2), a quantidade de informações disponíveis para uso reduziu-se gradativamente. Contudo, para não minar tanto a usabilidade do sistema, a maioria das informações estão disponíveis em português e outras em inglês.

No momento, a aplicação possui 11.700 filmes e 4.319 séries.

4. Limitação na execução dos algoritmos de recomendação

Antes de aplicar o critério de uso dos dados - mencionado no item anterior -, havia, aproximadamente, 40 mil filmes na base de dados. Quando tentou-se executar o algoritmo de recomendação baseada no conteúdo dos itens, ocorreu um estouro de memória RAM utilizada. O que se aprendeu dessa situação é que se a base de dados aumentar, o sistema de recomendação da aplicação atual não funcionará. Será necessário realizar uma atualização na infraestrutura da máquina usada ou migrar as sub aplicações para a nuvem (máquinas potentes disponibilizadas por empresas que podem ser acessadas remotamente). Espera-se que otimizar o algoritmo ou até mesmo testar outros podem ajudar nesse sentido também.

6. CONCLUSÃO

Conforme definido no objetivo geral (seção 1.1.2), este projeto teve a proposta de desenvolver uma aplicação *web* com algumas características de uma rede social, com um sistema de recomendação para auxiliar cinéfilos a encontrarem filmes ou séries de interesse com facilidade.

Para desenvolver o projeto, foram estudados os conceitos e tecnologias necessárias para desenvolver tanto o sistema de recomendação, quanto o *front-end* do sistema, apresentados na seção 3.3.

Sobre os sistemas de recomendações desenvolvidos, apesar de encontrar materiais relativamente abundantes na internet, foi o assunto mais denso a ser entendido e posto em prática. Principalmente, ao se preocupar em produzir um sistema minimamente funcional.

Como encerramento, conclui-se que o projeto atingiu os principais objetivos propostos (interação com os títulos e as recomendações) e infere-se que ele consiga auxiliar os cinéfilos a encontrarem filmes ou séries de interesse.

Como trabalhos futuros, existe uma vasta margem de possibilidades para desenvolver melhorias e novas funcionalidades, que podem melhorar a experiência dos usuários, tais como:

- Adicionar mais funcionalidades de redes sociais ao sistema:
    - Permitir que os usuários criem grupos de discussão, controlem quem pode participar e a visibilidade desse grupo;
    - Disponibilizar aos usuários um mural em sua página de perfil para poderem receber mensagens de outras pessoas;
    - Permitir que os usuários consigam marcar pessoas (ator, atriz e diretor, por exemplo) como favoritas para receber notificações de lançamentos;
    - Permitir que usuários enviem solicitações de amizades a outras pessoas;
    - Disponibilizar o grau de compatibilidade que um usuário pode ter com outros;
    - Disponibilizar um canal de denúncias em lugares que possam ser comentados conteúdos impróprios.
- Tornar o sistema mais inclusivo ao aplicar técnicas de acessibilidade;
- Disponibilizar o sistema na versão de aplicativo para aparelhos móveis;

- Encontrar outro serviço de terceiro ou desenvolver um *web scraping* para aumentar o catálogo de filmes e séries com dados com maior qualidade (em português);
    - Usar uma ferramenta de tradução paga pode ajudar nesse sentido;
- Melhorar o desempenho do sistema;
- Adicionar outras formas de login, ao usar integrações com outras plataformas como Facebook, Twitter e Google;
- Melhorar o motor de busca;
- Disponibilizar uma página com as estatísticas de conteúdos consumidos para o usuário;
- Melhorar a página de recomendações personalizadas;
- Adicionar a técnica de recomendação híbrida;
- Melhorar os sistemas de recomendações existentes ao testar outros algoritmos/modelos, caso necessário;
- Executar as recomendações em tempo real para que os usuários visualizem rapidamente as sugestões de conteúdo.

REFERÊNCIAS

ALAKE, Richmond. **Understanding Cosine Similarity And Its Application**. 2020. Disponível em: https://towardsdatascience.com/understanding-cosine-similarity-and-its-application-fd42f585296a. Acesso em: 15 jan. 2023.

AWS. **O que é Java?** 2023. Disponível em: https://aws.amazon.com/pt/what-is/java/?nc1=h_ls. Acesso em: 09 jan. 2023a.

AWS. **O que é uma API?** 2023. Disponível em: https://aws.amazon.com/pt/what-is/api/?nc1=h_ls. Acesso em: 12 jan. 2023b.

BROWNLEE, Jason. **Difference Between Classification and Regression in Machine Learning**. 2017. Disponível em: https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/. Acesso em: 15 jan. 2023.

C3.AI. Root Mean Square Error (RMSE). 2023. Disponível em: https://c3.ai/glossary/data-science/root-mean-square-error-rmse/. Acesso em: 15 jan. 2023.

CAMARGO, Robson. Quais os benefícios de criar um diagrama de contexto? 2018. Disponível em: https://robsoncamargo.com.br/blog/Quais-os-beneficios-de-criar-um-diagrama-de-contexto. Acesso em: 18 ago. 2022.

CASAROTTO, Camila. TF-IDF: a abordagem de otimização on page que seu blog precisa. a abordagem de otimização on page que seu blog precisa. 2019. Disponível em: https://rockcontent.com/br/blog/tf-idf/. Acesso em: 15 jan. 2023.

CHEE, Brian J. S.; FRANKLIN, Curtis. Cloud Computing: technologies and strategies of the ubiquitous data center. Boca Raton, Fl: Crc Press, 2019. 288 p.

CHEN, Jing; FANG, Jianbin; LIU, Weifeng; TANG, Tao; YANG, Canqun. CIMF: a fine-grained and portable alternating least squares algorithm for parallel matrix factorization. **Future Generation Computer Systems**, Amsterdam, The Netherlands, v. 108, p. 1192-1205, jul. 2020. Disponível em: https://www.sciencedirect.com/science/article/pii/S0167739X17318198. Acesso em: 07 fev. 2023.

CLOUD, Google. What is Apache Spark? 2023. Disponível em: https://cloud.google.com/learn/what-is-apache-spark. Acesso em: 17 jan. 2023.

COMMONS, Wikimedia. **File:HTML source code example.svg**. 2023. Disponível em: https://commons.wikimedia.org/wiki/File:HTML_source_code_example.svg. Acesso em: 07 fev. 2023.

CURY, Thiago. **TIPOS DE CHAVES NO BANCO DE DADOS RELACIONAL**. 2015. Disponível em:

http://thiagocury.eti.br/disciplinas/banco-de-dados-relacional/conceito-de-chaves-no-banco-de-dados-relacional.php. Acesso em: 07 fev. 2023.

ECLIPSE. **Using JavaScript Syntax Coloring**. Disponível em: https://www.eclipse.org/pdt/help/html/using_javascript_syntax_coloring.htm. Acesso em: 07 fev. 2023.

ELETRONICS, Starting. **CSS Introduction**: part 12 of the arduino ethernet shield web server tutorial. Part 12 of the Arduino Ethernet Shield Web Server Tutorial. 2013. Disponível em: https://startingelectronics.org/tutorials/arduino/ethernet-shield-web-server-tutorial/CSS-introduction/. Acesso em: 07 fev. 2023.

GAROUSI, Vahid; MESBAH, Ali; BETIN-CAN, Aysu; MIRSHOKRAIE, Shabnam. A systematic mapping study of web application testing. Information And Software Technology. Newton, Ma, United States, p. 1374-1396. ago. 2013. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0950584913000396. Acesso em: 12 jan. 2023.

GÉRON, Aurélien. **Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow**: concepts, tools, and techniques to build intelligent systems. 3. ed. Sebastopol, Ca: O'Reilly Media, 2022. 856 p.

HAZEM, Heidy; AWAD, Ahmed; YOUSEF, Ahmed Hassan. A distributed real-time recommender system for big data streams. **Ain Shams Engineering Journal.** Amsterdam, Netherlands, p. 102026-102052. 10 abr. 2022. Disponível em: https://www.sciencedirect.com/science/article/pii/S2090447922003379. Acesso em: 10 jan. 2023.

HORA, Andre. **Fig 1 - uploaded by Andre Hora**. 2023. Disponível em: https://www.researchgate.net/figure/Example-of-code-sample-SpringBoot-framework_fig1_336633890. Acesso em: 07 fev. 2023.

HUYEN, Chip. **Designing Machine Learning Systems**: an iterative process for production-ready. Sebastopol, Ca: O'Reilly Media, 2022. 386 p.

IBM. **What is Java Spring Boot?** 2023. Disponível em: https://www.ibm.com/topics/java-spring-boot. Acesso em: 09 jan. 2023a.

IBM. **What is MongoDB?** 2023. Disponível em: https://www.ibm.com/topics/mongodb. Acesso em: 09 jan. 2023b.

IBM. **What is natural language processing (NLP)?** 2023. Disponível em: https://www.ibm.com/topics/natural-language-processing. Acesso em: 16 jan. 2023c.

INTRODUCTION: What is Vue?. Disponível em: https://vuejs.org/guide/introduction.html#what-is-vue=. Acesso em: 02 jun. 2022.

ISINKAYE, F.O.; FOLAJIMI, Y.O.; OJOKOH, B.A.. Recommendation systems: principles, methods and evaluation. **Egyptian Informatics Journal.** Amsterdam, The

Netherlands, p. 261-273. 20 ago. 2015. Disponível em: https://reader.elsevier.com/reader/sd/pii/S1110866515000341?token=005A7457861A8B9178CBB9E6E5F2499DC99056A0DEAFCD8556A36A7FBDE35EF5FFFF27D711B1DA9580C61E01C1FBDDB8&originRegion=us-east-1&originCreation=20230117073544. Acesso em: 17 jan. 2023.

JETBRAINS. **Python code insight**. 2023. Disponível em: https://www.jetbrains.com/help/pycharm/python-code-insight.html. Acesso em: 07 fev. 2023a.

JETBRAINS. **Tailwind CSS**. 2023. Disponível em: https://www.jetbrains.com/help/phpstorm/tailwind-css.html. Acesso em: 07 fev. 2023b.

JETBRAINS. **Vue.js**. 2022. Disponível em: https://www.jetbrains.com/help/idea/vue-js.html. Acesso em: 07 fev. 2023.

JOHNSON, Steven. **O poder inovador da diversão**: como o prazer e o entretenimento mudaram o mundo. Brasil: Zahar, 2017. 384 p. Tradução de: Claudio Carina.

KANTOR, Ilya. **An Introduction to JavaScript**. Disponível em: https://javascript.info/intro. Acesso em: 02 jun. 2022.

KERVIZIC, Julien. **Become a Pro at Pandas, Python's Data Manipulation Library**. Disponível em: https://www.kdnuggets.com/2019/06/pro-pandas-python-library.html. Acesso em: 07 fev. 2023.

LIAO, Kevin. **Prototyping a Recommender System Step by Step Part 2**: alternating least square (als) matrix factorization in collaborative filtering. Alternating Least Square (ALS) Matrix Factorization in Collaborative Filtering. 2018. Disponível em: https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1. Acesso em: 16 jan. 2023.

MCKINNEY, Wes. **Python for Data Analysis**: data wrangling with pandas, numpy, and jupyter. 3. ed. Sebastopol, Ca: O'Reilly Media, 2022. 579 p.

MDN CONTRIBUTORS. **HTML: Linguagem de Marcação de Hipertexto**. 2022. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/HTML. Acesso em: 02 jun. 2022a.

MDN CONTRIBUTORS. **CSS**. 2022. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/CSS. Acesso em: 02 jun. 2022b.

MENDES, Antonio. **Arquitetura de Software**: desenvolvimento orientado para arquitetura. 1. ed. Editora Campus. Rio de Janeiro - RJ, 2002. 240 p.

MONGODB. **Get Started with MongoDB**. Disponível em: https://www.mongodb.com/basics/get-started. Acesso em: 07 fev. 2023a.

MONGODB. **What Is a Non-Relational Database?** 2023. Disponível em: https://www.mongodb.com/databases/non-relational. Acesso em: 12 jan. 2023b.

NVIDIA. **Scikit-learn**: what is scikit-learn?. What Is Scikit-learn?. 2023. Disponível em: https://www.nvidia.com/en-us/glossary/data-science/scikit-learn/. Acesso em: 09 jan. 2023.

O QUE você quer fazer com diagramas de fluxo de dados? Disponível em: https://www.lucidchart.com/pages/pt/o-que-e-um-diagrama-de-fluxo-de-dados. Acesso em: 18 ago. 2022.

PEREIRA, Rafael; LOPES, Hélio; BREITMAN, Karin; MUNDIM, Vicente; MUNDIM, Vicente. Cloud based real-time collaborative filtering for item–item recommendations. **Computers In Industry.** Amsterdam, Netherlands, p. 279-290. fev. 2014. Disponível em: http://research.globo.com/articles/tkde.pdf. Acesso em: 10 jan. 2023.

PYTHON. **What is Python? Executive Summary**. 2023. Disponível em: https://www.python.org/doc/essays/blurb/. Acesso em: 09 jan. 2023.

RAHMAN, Md. Masudur. **Fig 4 - uploaded by Md. Masudur Rahman**. 2023. Disponível em: https://www.researchgate.net/figure/Source-Code-Example-Customerjava-Partial_fig2_317401664. Acesso em: 07 fev. 2023.

ROSEN, Devan. **The Social Media Debate**. New York, Ny: Routledge, 2022. 234 p.

ROSSI, Gustavo; PASTOR, Oscar; SCHWABE, Daniel; OLSINA, Luis. **Web Engineering**: modelling and implementing web applications. London, Uk: Springer, 2010. 462 p.

S, Gigimol; JOHN, Sincy. A Survey on Different Types of Recommendation Systems. **International Research Journal Of Advanced Engineering And Science.** Jammu And Kashmir, p. 111-113. jan. 2016. Disponível em: http://irjaes.com/wp-content/uploads/2020/10/IRJAES-V1N4P187Y16.pdf. Acesso em: 15 jan. 2023.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Database System Concepts**. 7. ed. United States: McGraw-Hill Education, 2019. 1376 p.

SPILKER, Hendrik Storstein; COLBJØRNSEN, Terje. The dimensions of streaming: toward a typology of an evolving concept. **Media, Culture & Society.** United Kingdom, p. 1210-1225. 27 fev. 2020. Disponível em: https://journals.sagepub.com/doi/full/10.1177/0163443720904587. Acesso em: 12 jan. 2023.

SOPHIE. **A gentle introduction to Alternating Least Squares**. 2018. Disponível em: https://sophwats.github.io/2018-04-05-gentle-als.html. Acesso em: 16 jan. 2023.

SOPPIN, Shashidhar. **Machine Learning**: building a predictive model with scikit-learn. Building a Predictive Model with Scikit-learn. 2018. Disponível em: https://www.opensourceforu.com/2018/09/machine-learning-building-a-predictive-model-with-scikit-learn/. Acesso em: 07 fev. 2023.

SUFIYAN, Taha. **What is Node.js**: a comprehensive guide. A Comprehensive Guide. 2023. Disponível em: https://www.simplilearn.com/tutorials/nodejs-tutorial/what-is-nodejs. Acesso em: 07 fev. 2023.

TMDB. The Movie Database. 2023. Disponível em: https://www.themoviedb.org/. Acesso em: 24 jan. 2023.

TURNEY, Shaun. **Coefficient of Determination (R^2) | Calculation & Interpretation**. 2022. Disponível em: https://www.scribbr.com/statistics/coefficient-of-determination/#:~:text=What%20is%20the%20definition%20of,predicted%20by%20the%20statistical%20model. Acesso em: 15 jan. 2023.