

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS DE CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

ELIVELTON PEREIRA DA SILVA

RECOLIST: UM SISTEMA PARA AUXILIAR NA ESCOLHA DE FILMES E SÉRIES
PARA ASSISTIR

CAJAZEIRAS
2024

Elivelton Pereira da Silva

RECOLIST: UM SISTEMA PARA AUXILIAR NA ESCOLHA DE FILMES E SÉRIES
PARA ASSISTIR

Trabalho de Conclusão de Curso submetido ao Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Campus Cajazeiras, como requisito parcial para a obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Fabio Gomes de Andrade

Cajazeiras
2024

IFPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

S586r Silva, Elivelton Pereira da.
Recolist : um sistema para auxiliar na escolha de filmes e séries para assistir / Elivelton Pereira da Silva. – 2024.
36f. : il.
Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2024.
Orientador(a): Prof. Dr. Fabio Gomes de Andrade.
1. Desenvolvimento de sistemas. 2. Sistema *web Recolist*. 3. *Streaming*. 4. Seleção de filmes. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

IFPB/CZ

CDU: 004.4(043.2)



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

ELIVELTON PEREIRA DA SILVA

RECOLIST: UM SISTEMA PARA AUXILIAR NA ESCOLHA DE FILMES E SÉRIES PARA ASSISTIR

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Dr. Fabio Gomes de Andrade

Aprovada em: **16 de Outubro de 2024.**

Prof. Dr. Fabio Gomes de Andrade - Orientador

Prof. Me. Diogo Dantas Moreira - Avaliador

IFPB - Campus Cajazeiras

Prof. Tec.go. Antônio Ricart Jacinto de Oliveira Medeiros

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Fabio Gomes de Andrade**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 17/10/2024 19:39:44.
- **Diogo Dantas Moreira**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 19/10/2024 08:53:54.
- **Antonio Ricart Jacinto de Oliveira Medeiros**, PROF ENS BAS TEC TECNOLOGICO-SUBSTITUTO, em 21/10/2024 11:44:46.

Este documento foi emitido pelo SUAP em 16/10/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 620979
Verificador: 7b20a641be
Código de Autenticação:



Rua José Antônio da Silva, 300, Jardim Oásis, CAJAZEIRAS / PB, CEP 58.900-000
<http://ifpb.edu.br> - (83) 3532-4100

AGRADECIMENTOS

Agradeço inicialmente à minha mãe, que sempre me apoiou em tudo e acreditou em mim quando iniciei esta graduação.

Agradeço também aos amigos e familiares que me incentivaram e me acompanharam durante todo o tempo do curso.

Por fim, minha gratidão a todos os professores que me ajudaram a ter o conhecimento que tenho hoje, especialmente ao prezado professor e orientador, Dr. Fabio Gomes de Andrade, pela motivação, dedicação, oportunidade e paciência.

RESUMO

Os serviços de streaming se popularizaram muito e se tornaram um dos principais meios de entretenimento para as pessoas. Com isso, diversas plataformas de streaming foram surgindo ao longo do tempo, trazendo uma vasta quantidade de títulos e, conseqüentemente, causando a fragmentação de conteúdo e dificultando a escolha do usuário quanto ao que assistir em relação a filmes e séries. Isso faz com que ele perca tempo visualizando todo o catálogo, o que ocasionalmente, provoca a desistência em assistir algo. Visando simplificar essa decisão, este trabalho propõe um sistema web que oferece uma nova forma de visualizar obras interessantes para o usuário com base em recomendações personalizadas. Além disso, a solução proposta auxilia o usuário a decidir quais serviços de assinatura ele deve priorizar, indicando quais plataformas são mais relevantes com base nas preferências do usuário.

Palavras-chave: Fragmentação. Recomendação. Streaming. Usuário.

ABSTRACT

Streaming services have become very popular and have become one of the main forms of entertainment for people. With this, several streaming platforms have emerged over time, bringing a vast amount of titles and, consequently, causing content fragmentation and making it difficult for users to choose what to watch in relation to movies and series. This often results in users spending time browsing the entire catalog, which occasionally leads them to give up on watching something altogether. To simplify this decision, this work proposes a web-based system that provides users with a new way to view recommended titles based on personalized suggestions. Additionally, the proposed solution helps users decide which subscription services to prioritize by indicating which platforms are most relevant according to their preferences.

Palavras-chave: Fragmentation. Recommendation. Streaming. User.

LISTA DE FIGURAS

Figura 1 - Exemplo de requisição de acesso à endpoint de API.....	15
Figura 2 - Exemplo de resposta de endpoint de API.....	16
Figura 3 - Exemplo de estrutura de um documento do MongoDB.....	17
Figura 4 - Modelagem de dados.....	20
Figura 5 - Arquitetura do sistema.....	22
Figura 6 - Exemplo de rota /meu-perfil e função.....	24
Figura 7 - Exemplo de rota /logout e função.....	25
Figura 8 - Página inicial.....	26
Figura 9 - Página de login	27
Figura 10 - Página de cadastro	28
Figura 11 - Página de visualização de filme ou série	29
Figura 12 - Avaliações e relacionados de filme ou série.....	29
Figura 13 - Página de resultados de pesquisa	30
Figura 14 - Perfil do usuário.....	31

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
TCC	Trabalho de conclusão de curso
TMDb	The Movie Database
DVD	Digital Versatile Disc
JSON	JavaScript Object Notation
XML	Extensible Markup Language

SUMÁRIO

1. INTRODUÇÃO	9
1.1 Motivação	9
1.2 Objetivos	10
1.2.1 Objetivo Geral	10
1.2.2 Objetivos Específicos	10
1.3 Trabalhos relacionados	11
1.4 Metodologia	12
1.5 Organização do documento	12
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 Serviços de streaming de vídeo	14
2.2 TMDb API	14
2.3 MongoDB	16
3. A SOLUÇÃO PROPOSTA	18
3.2 Requisitos funcionais	18
3.3 Os dados	19
3.3.1 O processo de coleta dos dados	19
3.3.2 Modelagem dos dados	20
3.4 Arquitetura do sistema	21
3.5 Implementação	23
3.5.1 Módulo de solicitação de informações de filmes e séries	23
3.5.2 Módulo de gerenciamento de usuários	23
3.5.3 Módulo servidor	24
3.5.4 Módulo de visualização	25
3.6 Páginas da aplicação	26
3.6.1 Página inicial	26
3.6.2 Página de Login e cadastro	27
3.6.3 Página de visualização de filme ou série	28
3.6.4 Página de pesquisa	30
3.6.5 Página de perfil do usuário logado	30
5. CONCLUSÃO	31
REFERÊNCIAS	33

1. INTRODUÇÃO

A procura por formas de entretenimento está presente nas gerações atuais e passadas. Hoje, com a facilidade de acesso à internet, é possível ter acesso a diversas formas de entretenimento, seja jogando, lendo ou ouvindo música. Outra forma muito comum de entretenimento nos dias de hoje consiste em utilizar serviços de *streaming* para assistir a filmes e séries, o que tem se tornado cada vez mais frequente na vida de várias pessoas.

Os serviços de *streaming* de vídeo vieram como uma nova alternativa de entretenimento audiovisual, antes oferecida apenas pela televisão e pelo cinema. Atualmente, é possível ter acesso a uma grande quantidade de filmes e séries e, até mesmo, de plataformas diferentes. Essas plataformas produzem conteúdos exclusivos e adquirem os direitos autorais de obras já produzidas, que, em sua maioria, não são de total exclusividade.

Com isso, esses serviços de *streaming* vêm se tornando cada vez mais populares no mundo todo, com uma vasta quantidade de plataformas de *streaming* disponíveis atualmente. As plataformas mais populares de filmes e séries no território brasileiro em 2024 são a *Netflix*, a *Globoplay*, a *Disney+*, o *Star+*, o *Amazon Prime Video*, a *HBO Max*, a *Apple TV+* e a *Paramount Plus*. Cada uma delas tem sua variação de preço e a quantidade de filmes e séries disponibilizados, além de contar com obras exclusivas.

1.1 Motivação

A grande quantidade de vídeos e séries disponíveis atualmente pode dificultar a vida do usuário na hora de decidir o que ele deseja assistir em um determinado momento. Segundo uma pesquisa realizada pela Nielsen¹ Em julho de 2023, um usuário gasta em média dez minutos e meio procurando algo que seja interessante para assistir. Esse número aumentou em relação ao último levantamento feito em 2019, que, na época, era de sete minutos. A mesma pesquisa também informa que 20% das pessoas não sabem o que assistir antes de procurarem, não conseguem encontrar algo para assistir e, ao invés disso, decidem realizar outra tarefa. Assim, o desenvolvimento de uma solução que auxilie as

¹ <https://www.nielsen.com/pt/insights/2023/media-metadata-success-in-streaming/>.

pessoas na realização dessa tarefa é de grande importância, tornando essa escolha mais fácil e proporcionando uma melhor experiência para os usuários.

A vasta quantidade de plataformas de *streaming* competindo por audiência e público para suas obras trouxe consigo a fragmentação dos serviços de streaming. Um usuário, ao pesquisar um filme ou série que deseja assistir, pode se deparar com um impasse de que a obra pesquisada só esteja disponível em um determinado serviço de streaming, gerando a dúvida sobre qual das plataformas disponíveis atende melhor às suas expectativas.

Com o intuito de superar essas limitações, este trabalho de conclusão de curso propõe uma aplicação que auxilia os usuários na hora de escolher um filme ou série para assistir, por meio de recomendações geradas com base nos gostos e preferências do usuário. Além disso, o trabalho desenvolvido também ajuda os usuários na escolha de serviços de streaming de vídeo por assinatura que tenham mais obras do seu interesse.

1.2 Objetivos

Esta seção descreve os objetivos deste trabalho de conclusão de curso.

1.2.1 Objetivo Geral

O objetivo geral deste TCC consiste no desenvolvimento de um sistema web que auxilie as pessoas na escolha de filmes e séries.

1.2.2 Objetivos Específicos

Este TCC tem também os seguintes objetivos específicos:

- fornecer um sistema de recomendação de filmes e séries personalizado, com base nas preferências de cada usuário;
- auxiliar o usuário na escolha de serviços de *streaming* com base na quantidade de itens de seu interesse;
- proporcionar uma melhor experiência ao usar este tipo de entretenimento;
- agilizar o processo de decisão do usuário, facilitando a escolha do que assistir.

1.3 Trabalhos relacionados

Atualmente, existem diversas aplicações que auxiliam na hora da decisão sobre o que assistir em relação a filmes e séries. Uma delas é a *WhatToWatch* (WhatToWatch, 2023), que tem como objetivo reduzir a experiência da fragmentação, disponibilizando um catálogo e recomendando filmes e séries para ajudar o usuário a encontrar o que assistir. Entretanto, não há um bom *feedback* para o usuário sobre quais serviços de streaming ele deve priorizar com base nas listas criadas.

Outra aplicação semelhante é a *Movielub* (RODRIGUES, 2021), que é bem semelhante ao *WhatToWatch* (SILVA, 2023), e auxilia o usuário a encontrar filmes do seu interesse, utilizando um sistema de recomendação de filmes baseado no gosto pessoal do usuário. Esses trabalhos utilizam a API do *The movie database* (TMDb, 2023)², que também foi utilizada como fonte de dados deste projeto. Contudo, igual ao seu semelhante, ele não tem um feedback sobre quais serviços de streaming devem ser priorizados e, além disso, não há informações de séries, somente filmes.

Outro sistema relacionado é o *JustWatch*³ (JustWatch, 2023), uma plataforma que mantém o usuário atualizado sobre as novidades em diversas plataformas de *streaming*. O sistema auxilia o usuário a ver o que é relevante para ele, contando com um sistema de acompanhamento de filmes e séries e notificando quando um título interessado está disponível. Entretanto, esse sistema não conta com uma seção de comentários e avaliações dos usuários que assistiram a obra.

O *Filmow*⁴ (Filmow, 2023), foi criado para pessoas que têm interesse em filmes e tem como sua principal ideia permitir que os usuários mostrem aos seus amigos os filmes que você já assistiram e as suas respectivas avaliações. Nessa plataforma o usuário pode ser informado sobre os filmes que serão lançados, os que estão no cinema e os que estão disponíveis em DVD para serem assistidos de casa. Porém, esse sistema não conta com a filtragem de títulos por serviço de streaming.

² <https://www.themoviedb.org/?language=pt-BR>

³ <https://www.justwatch.com/br>.

⁴ <https://filmow.com/>.

1.4 Metodologia

Para se atingir o objetivo proposto no projeto, foi necessário o esclarecimento das etapas de execução e planejamento. É essencial que seja demonstrado não só a proposta do projeto, mas também os recursos utilizados para atingir o objetivo. Assim, para a implementação deste trabalho de conclusão de curso, foi estabelecido o desenvolvimento das seguintes atividades:

- **Análise do estado da arte (A₁):** durante o período dessa atividade, foi realizada a leitura de materiais e projetos que abordam sistemas que auxiliam na decisão do que assistir no momento. A leitura, além do uso de plataformas que já estão desenvolvidas, ajudou a contextualizar as ideias que serão abordadas no trabalho a ser desenvolvido;
- **Elaboração do documento (A₂):** essa atividade refere-se à elaboração do documento deste trabalho de conclusão de curso;
- **Levantamento de requisitos (A₃):** nessa atividade foram identificados os requisitos necessários para atingir tanto o objetivo principal quanto os objetivos específicos do trabalho, definindo as funcionalidades que o sistema deveria disponibilizar para o usuário;
- **Definir a arquitetura do projeto (A₄):** nessa atividade foi definida a arquitetura do sistema, especificando a comunicação entre os seus módulos e o comportamento dos mesmos;
- **Projetar a base de dados (A₅):** nessa atividade foi definida a estrutura da base de dados, através de um esquema lógico que demonstra a estrutura do banco de dados implementado no sistema;
- **Implementação (A₆):** nessa atividade foi realizado o desenvolvimento do sistema. A atividade envolveu a codificação do sistema, implementação das funcionalidades definidas e a integração com as tecnologias necessárias.

1.5 Organização do documento

O restante deste documento está organizado da seguinte forma: o Capítulo 2 descreve a fundamentação teórica, expondo conceitos importantes e tecnologias essenciais para a resolução da problemática apresentada neste trabalho; o capítulo 3 descreve a solução proposta, apresentando os principais componentes

desenvolvidos durante o processo; o Capítulo 4 apresenta as conclusões e considerações finais deste trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados os conceitos, definições e tecnologias fundamentais para a realização deste trabalho. Inicialmente, serão abordados os serviços de streaming de vídeo, explicando a sua descrição técnica. Em seguida, apresenta a API do TMDb e o banco de dados MongoDB.

2.1 Serviços de streaming de vídeo

A evolução da internet e a expansão do seu número de usuários causaram um grande impacto na forma de se consumir obras audiovisuais. Isso levou ao surgimento dos serviços de streaming, e, com isso, surgiram meios alternativos que antes eram restritos à televisão e ao cinema.

A descrição técnica de *streaming*, muito presente nos recentes serviços de mídia, é o ato de transmitir e recuperar conteúdo digital armazenado e processado em um servidor remoto. Diferentemente do habitual *download*, o conteúdo é mantido em memória cache e não é armazenado permanentemente no disco rígido do dispositivo utilizado para acessar o serviço (SPILKER; COLBJØRNSSEN, 2020).

As indústrias cinematográficas, como a *Warner Bros*, estão tendo uma evolução conjunta aos serviços de streaming. No caso da *HBO Max*, as suas obras produzidas antes do surgimento desse tipo de serviço já são encontradas no seu catálogo, e, com a campanha “do cinema para sua casa”, não demora muito para disponibilizarem lançamentos recentes. Já a indústria televisiva, como a *Globoplay*, preza por obras exclusivas e conteúdos relacionados à emissora.

2.2 TMDb API

O *The Movie Database* (TMDb) é um banco de dados criado por Travis Bell em 2008 e guarda informações acerca de filmes e programas de TV. Desde sua criação, o TMDb é atualizado frequentemente por meio da contribuição da sua própria comunidade. Oficialmente, o banco de dados oferece suporte a 39 idiomas e é amplamente utilizado em 180 países. A API TMDb tem o objetivo de fornecer acesso às informações relacionadas a filmes e séries armazenadas no banco de dados do TMDb (TMDb, 2023).⁵

Com a data de acesso de 05 de outubro de 2024, o banco de dados fornecido contava com os seguintes status:

⁵<https://www.themoviedb.org>.

- 976.292 filmes (+113.036 em relação a novembro/2023) ;
- 182.509 séries (+23.365 em relação a novembro/2023);
- 304.456 temporadas de séries (+41.614 em relação a novembro/2023);
- 4.846.616 episódios de séries (+633.944 em relação a novembro/2023);
- 3.648.278 pessoas (atores e equipe de gravação) (+533.622 em relação a novembro/2023);
- 6.339.536 imagens (+1.206.624 em relação a novembro/2023);

Para permitir que os seus dados possam ser acessados e utilizados por aplicações de software, o TMDb disponibiliza uma API que disponibiliza *endpoints* para acesso a dados sobre filmes, séries, temporadas, episódios entre outras informações.

A Figura 1 representa uma requisição feita para acessar o *endpoint* que fornece informações de filmes populares. Esse código base é disponibilizado na documentação da API do TMDb. Para a utilização desta API, é necessário realizar um cadastro no site para obter uma chave única para conseguir acesso aos seus recursos. Essa chave é descrita no código como “process.env.CHAVE_API”, protegida pela biblioteca *dotenv*, que oculta as informações através do gerenciamento de variáveis de ambiente. Feita a requisição, a API retorna um documento *JSON* com uma lista de filmes populares, contendo informações de títulos, popularidade, gêneros, sinopse, entre outras, conforme exibido na figura 2.

Figura 1 - Exemplo de requisição de acesso à endpoint de API

```
1 require('dotenv').config()
2
3 const options = {
4   method: 'GET',
5   headers: {
6     accept: 'application/json',
7     Authorization: `Bearer ${process.env.CHAVE_API}`
8   }
9 };
10
11 fetch('https://api.themoviedb.org/3/movie/popular?language=pt-BR&page=1', options)
12   .then(response => response.json())
13   .then(response => console.log(response))
14   .catch(err => console.error(err));
```

Fonte: Autor

Figura 2 - Exemplo de resposta de endpoint de API

```
{
  "backdrop_path": "/lgkPzcOSnTvjeMnuFzozR05HHw1.jpg",
  "id": 519182,
  "title": "Meu Malvado Favorito 4",
  "original_title": "Despicable Me 4",
  "overview": "Nesta sequência, o vilão mais amado do planeta retorna e agora Gru, Lucy, Margo, Edith e Agnes dão as boas-vindas a um novo membro da família: Gru Jr., que pretende atormentar seu pai. Enquanto se adapta com o pequeno, Gru enfrenta um novo inimigo, Maxime Le Mal, forçando sua namorada Valentina e a família a fugir do perigo.",
  "poster_path": "/jWYTtmxSuWVXP22hxAeXdQZLZrh.jpg",
  "media_type": "movie",
  "adult": false,
  "original_language": "en",
  "genre_ids": [
    16,
    10751,
    35,
    28
  ],
  "popularity": 1187.775,
  "release_date": "2024-06-20",
  "video": false,
  "vote_average": 7.125,
  "vote_count": 1996
}
```

Fonte: Autor

2.3 MongoDB

O *MongoDB* (BANKER, 2016) é um banco de dados não relacional orientado a documentos. Ele é um banco de dados distribuído em seu núcleo, de modo que a alta disponibilidade, a escalabilidade horizontal e a distribuição geográfica são incorporadas e fáceis de usar. Ele oferece armazenamento de dados em documentos flexíveis do tipo *JSON* e *XML*, o que significa que os campos podem variar de documento para documento e a estrutura de dados pode ser alterada ao longo do tempo.

O modelo de documento mapeia os objetos em seu código de aplicativo, tornando os dados fáceis de trabalhar. Na Figura 3 é mostrado um exemplo da estrutura de um documento do MongoDB. Cada documento possui um identificador, definido como “*_id*”. Na figura mostrada, os campos “*firstname*” e “*lastname*” armazenam, respectivamente, um nome e sobrenome. Além de campos para armazenar informações, o MongoDB também é capaz de armazenar documentos de forma aninhada, como é mostrado no exemplo do campo “*address*”, que contém os campos “*street*”, “*city*”, “*state*” e “*zip*”, que representam informações de endereço, respectivamente, rua, cidade, estado e código postal.

Figura 3 - Exemplo de estrutura de um documento do MongoDB

```
{
  _id: "5cf0029caff5056591b0ce7d",
  firstname: 'Jane',
  lastname: 'Wu',
  address: {
    street: '1 Circle Rd',
    city: 'Los Angeles',
    state: 'CA',
    zip: '90404'
  }
}
```

Fonte: MongoDB⁶

⁶ <https://www.mongodb.com/pt-br/what-is-mongodb>.

3. A SOLUÇÃO PROPOSTA

Este capítulo descreve a implementação da solução proposta neste trabalho. Inicialmente, ele apresenta os possíveis usuários junto aos requisitos funcionais. Em seguida, são apresentados a modelagem de dados do sistema, a arquitetura com a descrição de seus módulos e, por fim, a implementação final realizada e os protótipos das telas.

3.1 Atores do sistema

Os possíveis usuários para este sistema são pessoas que buscam recomendações de filmes ou séries e que desejam saber qual plataforma contém as opções de títulos que eles tenham interesse de assistir, assim como aqueles que não sabem o que assistir e procuram sugestões. Além disso, o público também inclui pessoas que possuem vários serviços de streaming, bem como aqueles que estão considerando assinar um novo serviço.

3.2 Requisitos funcionais

Os requisitos funcionais de um sistema descrevem o que ele deve fazer. Eles dependem do tipo de software a ser desenvolvido, de quem são seus possíveis usuários e da abordagem geral adotada pela organização ao escrever os requisitos. (Sommerville, 2011). O sistema proposto por este TCC auxilia na decisão do usuário sobre o que assistir, permitindo que ele escolha um filme ou série com base em recomendações personalizadas. Além disso, ele fornece informações úteis que ajudam o usuário a priorizar os serviços de assinatura que mais se alinham com os seus interesses. Para atingir esses objetivos, foram definidos os seguintes requisitos funcionais:

- **Gerenciar cadastro (RF1):** o sistema deveria permitir ao usuário criar uma conta e personalizar o seu perfil, além de gerenciar a privacidade de seus itens, como listas ou avaliações;
- **Criar listas personalizadas (RF2):** o sistema deveria permitir ao usuário criar listas personalizadas diretamente de recomendações com base no seu interesse ou escolha do mesmo. Essas listas poderiam ser organizadas por categoria e nome;

- **Avaliar filme ou série (RF3):** o sistema deveria permitir ao usuário avaliar um filme ou série, deixando um comentário exibido em seu perfil e atribuindo uma avaliação que pode ir de uma a cinco estrelas;
- **Marcar como visto (RF4):** o sistema deveria permitir ao usuário marcar como vista uma obra já assistida por ele. Se fosse uma série, o usuário poderia marcar os episódios da temporada que ele já assistiu;
- **Exibir quantidade de itens por plataforma (RF5):** o sistema deveria mostrar a quantidade de obras presentes nas listas do usuário presentes em cada plataforma de streaming, para que ele tivesse uma ideia sobre qual serviço ele deveria priorizar;
- **Visualizar perfil de outros usuários (RF6):** o sistema deveria permitir aos usuários visualizar perfis de outros usuários, mostrando o que o usuário que está sendo visualizado permitiu.

3.3 Os dados

O sistema desenvolvido conta com um banco de dados. As próximas subseções descrevem, respectivamente, o processo de coleta dos dados e o esquema lógico do banco de dados do sistema.

3.3.1 O processo de coleta dos dados

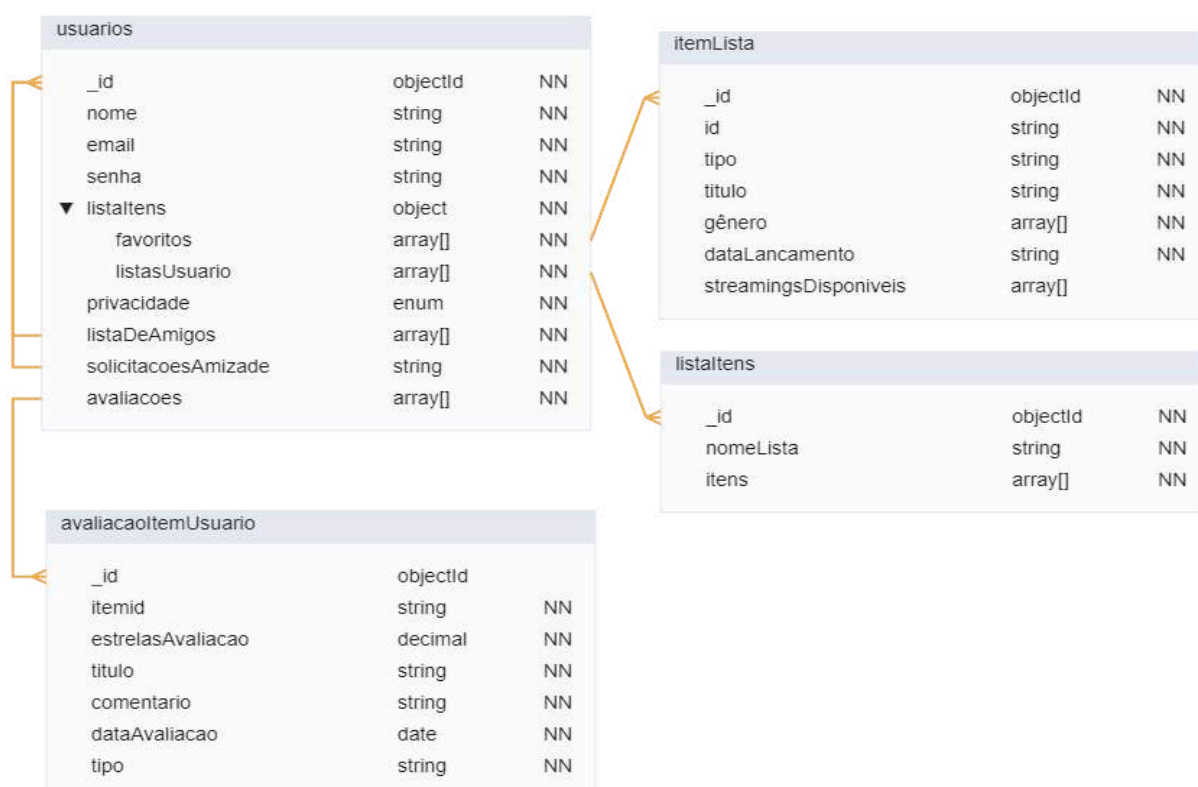
A *Application Programming Interface* (API) que fornece o acesso à base de dados utilizada neste projeto é a *The Movie Database* (TMDb). Através de requisições feitas para essa API são recuperadas informações sobre filmes, séries, temporadas, episódios, entre outras. Os dados são retornados como um documento no formato JSON. O sistema, ao recebê-los, realiza o tratamento e a manipulação desses dados e os armazena em um banco de dados local. Uma das principais escolhas para utilização desta API foi a sua boa documentação, com a disponibilização de um guia, uma referência para a API e um ambiente para testar as requisições, o que facilitou o caminho para a implementação.

Para o armazenamento persistente de informações decidiu-se utilizar o banco de dados não relacional *MongoDB*. Por ser um banco de dados orientado a documentos, ele armazena os dados em documentos semelhantes a *JSON* com esquemas. A sua vantagem é que ele permite um modelo de dados flexível. Assim, não é necessário definir a estrutura dos seus dados antes de inseri-los (MongoDB, 2023)⁷.

3.3.2 Modelagem dos dados

O esquema lógico usado para a implementação de banco de dados do sistema representa os dados como coleções, que são semelhantes às tabelas de um banco de dados relacional. A Figura 4 representa a coleção principal do sistema, que é a de usuários, descrevendo os seus atributos. Além disso, a figura mostra os dados aninhados das avaliações realizadas pelo usuário e também das listas dos usuários e dos filmes ou séries presentes nas listas.

Figura 4 - Modelagem de dados



Fonte: Autor

⁷ <https://www.mongodb.com/pt-br/what-is-mongodb>.

A coleção *usuarios* armazena as informações dos usuários da ferramenta, como o identificador, o nome, o email e senha, que será usada para fins de autenticação. Além disso, cada usuário tem uma referência para as suas listas de filmes ou séries junto da definição do tipo de privacidade. O tipo de privacidade pode ser: *público*, o que significa que qualquer usuário pode visualizar os dados públicos do usuário; *somente amigos*, o que significa que apenas os amigos do usuário podem visualizar esses dados; ou *privado*, o que significa que somente o próprio usuário pode visualizar seus dados. Cada usuário também conta com sua lista de amigos, solicitações de amizade para o usuário e avaliações sobre filmes ou séries realizadas pelo próprio usuário.

A coleção *listaltens* armazena as informações das listas criadas pelos usuários. Para cada lista são registrados o nome da lista, e uma lista de filmes ou séries que se denominam como itens.

A coleção *itemLista* armazena as informações dos filmes ou séries que compõem a lista de cada usuário. Para cada item são armazenados o identificador, que é importante para realizar as requisições para a API o TMDb, o título do do item, o tipo, utilizado para identificar se é um filme ou uma série, os gêneros, a data de lançamento e os serviços de *streaming* nos quais o conteúdo está disponível.

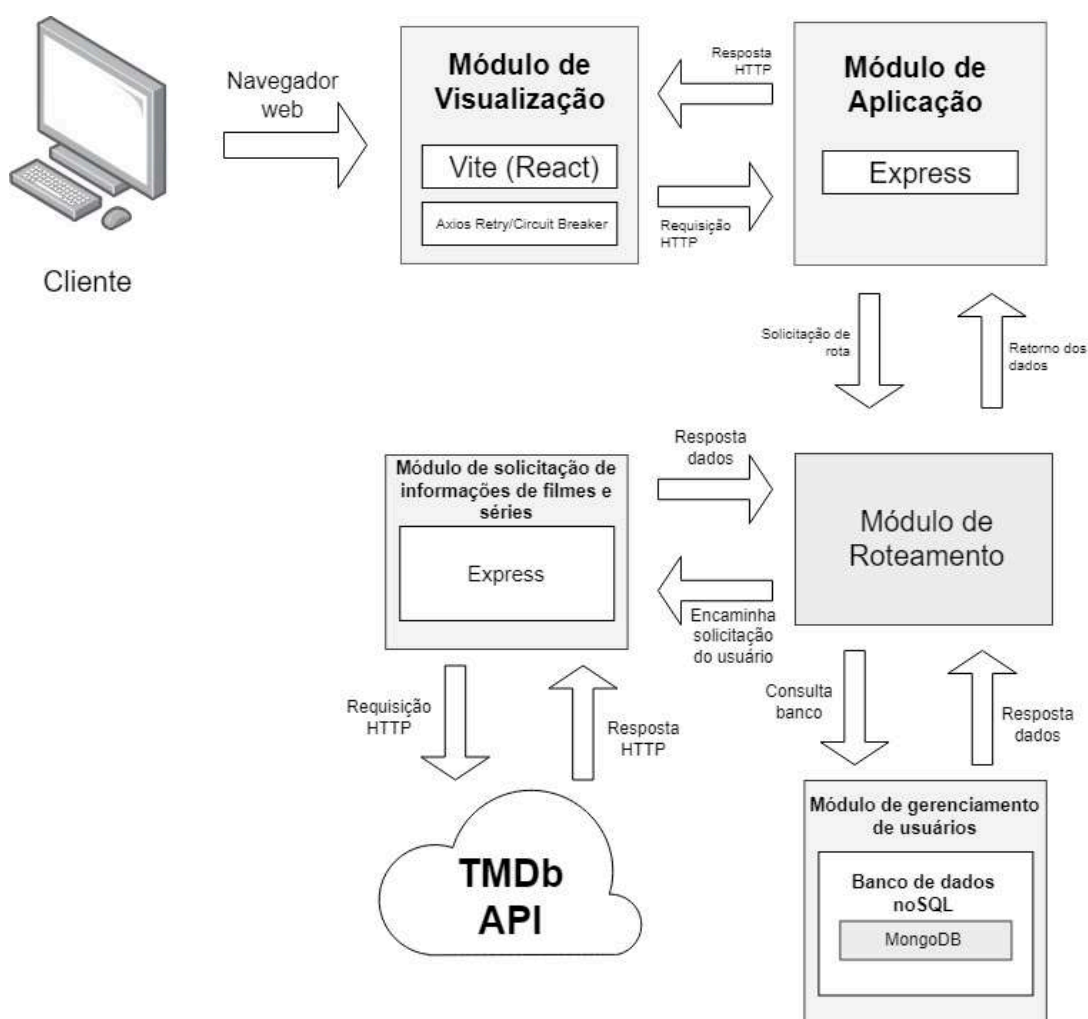
A coleção *avaliacaoItemUsuario* armazena as informações das avaliações realizadas pelos usuários. Para cada avaliação, são registrados o *itemId*, que identifica o ID do filme ou série avaliada, a quantidade de estrelas atribuídas (entre 1 e 5), o título da avaliação, a data da avaliação e o tipo, indicando se o item é um filme ou uma série.

3.4 Arquitetura do sistema

A arquitetura de implementação do sistema é apresentada na Figura 5. O *módulo de visualização* é responsável pela comunicação do cliente com o sistema. Essa interação é feita utilizando-se requisições HTTP e para a sua implementação foi usada a biblioteca *Axios*, com a implementação de técnicas de *retry* e *circuit breaker*. O *retry* realiza novas tentativas automaticamente caso ocorram falhas durante uma requisição. Já o *circuit breaker* bloqueia quaisquer requisições que falhem consecutivamente.

Por outro lado, o *módulo de aplicação* se comunica com o *módulo de roteamento* para solicitar o método necessário para a requisição do cliente, esse módulo age como um intermediário entre o *módulo de solicitação de informações de filmes e séries* e o *módulo de gerenciamento de usuários*.

Figura 5 - Arquitetura do sistema



Fonte: Autor

Na Figura 5 também é possível visualizar a interação entre os módulos que compõem a arquitetura. O *módulo de visualização* é responsável pela interação com o cliente, recebendo as suas requisições e exibindo os seus resultados na tela. A tecnologia utilizada para implementar este módulo foi a biblioteca React.js com o Vite, que é baseada na linguagem de programação java script.

Já o *módulo de aplicação* recebe as solicitações do *módulo de visualização* e consulta o *módulo de roteamento* para encontrar a rota responsável por lidar com essa solicitação. Após encontrar a rota adequada, será executado um método que processa a solicitação do cliente. Dependendo do tipo de requisição, se for para obter dados do usuário ou informações sobre filmes e séries, por fim, a solicitação será encaminhada ao módulo de gerenciamento de usuários ou o de solicitação de informações de filmes e séries.

O *módulo de gerenciamento de usuários* recebe a consulta através do *módulo de rotas* e faz uma operação de create, read, update ou delete (*CRUD*), e responde conforme sua solicitação.

O *módulo de solicitação de informações de filmes e séries* envia uma requisição à *API* do *TMDb* com base no que o usuário deseja visualizar e retorna a resposta com essas informações.

3.5 Implementação

Esta seção descreve a implementação do sistema de acordo com as especificações apresentadas na arquitetura (Figura 5).

3.5.1 Módulo de solicitação de informações de filmes e séries

A função do módulo de solicitação de informações de filmes e séries consiste em obter dados relevantes sobre esses itens, como sinopse, gêneros, data de lançamento, entre outros. Para desenvolver a funcionalidade deste módulo, é necessário enviar requisições para a *API* do The Movie Database (*TMDb*), que é responsável por retornar essas informações. Esse processo ocorre da seguinte maneira: o módulo de solicitação de informações de filmes e séries realiza uma requisição para a *API*. O método a ser invocado e os valores dos parâmetros dependem do tipo de solicitação enviada. A requisição inclui um identificador que define se o item pesquisado é um filme ou série. Com base nesses parâmetros, a *API* retorna os dados solicitados.

3.5.2 Módulo de gerenciamento de usuários

Através do *mongoDB*, o módulo de gerenciamento de usuários armazena e fornece informações sobre os usuários do sistema. Neste módulo é realizado o cadastro, atualização e exclusão de dados dos usuários. O módulo também

assegura uma melhor proteção para as informações sensíveis do usuário. Por exemplo, a senha informada pelo usuário ao cadastrar ou atualizar o sistema é criptografada utilizando um algoritmo de *hash*. Para o gerenciamento de sua privacidade, o módulo permite que os usuários controlem quem pode visualizar as suas informações. Para isso eles podem optar por três tipos de privacidade: *privada*, onde apenas o usuário tem acesso às suas informações; *somente amigos*, onde apenas os amigos do usuário podem visualizar suas informações e *público*, onde todos podem visualizar as informações do usuário.

3.5.3 Módulo servidor

O módulo servidor é responsável por enviar e gerenciar as requisições HTTP enviadas pelo usuário. Este módulo é o intermediário entre servidor e cliente, permitindo que o usuário acesse as informações do servidor de forma eficiente. Ele foi implementado usando a biblioteca *Express*. A solicitação do usuário é encaminhada através do *módulo de rotas*, que gerencia um conjunto de rotas que indicam o caminho para o qual a requisição deve ser encaminhada. Cada rota definida indica uma solicitação *HTTP* diferente. Por exemplo, na Figura 6, a rota *"/meu-perfil"* é utilizada para listar os dados do usuário logado atualmente, solicitando essas informações ao banco de dados como demonstrado na função de *meuPerfil*. Enquanto na figura 7, é demonstrada a rota *"/logout"* é utilizada para o usuário desconectar-se do sistema, encerrando sua sessão.

Figura 6 - Exemplo de rota */meu-perfil* e função

```
usuarioRouter.get("/meu-perfil", autUsuario,
UsuarioController.meuPerfil);
const meuPerfil = async (req, res) => { //UsuarioController.js
  try {
    const usuarioId = req.usuario.usuarioId;
    const usuario = await Usuario.findById(usuarioId, "-senha
-__v");
    res.send(usuario);
  } catch (erro) {
    res.status(500).json({ msg: "Erro ao obter perfil", erro:
erro.message });
  }
};
```

Fonte: Autor

Figura 7 - Exemplo de rota /logout e função

```
usuarioRouter.post("/logout", autUsuario,  
UsuarioController.logoutUsuario);  
const logoutUsuario = async (req, res) => { //UsuarioController.js  
  res.clearCookie("token");  
  res.clearCookie("usuario");  
  res.clearCookie("autenticado");  
  res.status(200).send("Usuário desconectado com sucesso");  
};
```

Fonte: Autor

Esse caminho invoca um método que realiza operações necessárias para atender a requisição, podendo ser direcionada para o *módulo de solicitação de filmes e séries* ou o *módulo de gerenciamento de usuários*. Este processo de roteamento garante que cada requisição seja tratada e encaminhada de maneira adequada.

3.5.4 Módulo de visualização

O módulo de visualização recebe requisições do cliente por meio de um navegador *web*. Esse módulo é responsável por exibir as informações fornecidas do módulo servidor ao usuário. Ele foi desenvolvido utilizando-se a *Vite*, que é uma ferramenta de construção de projetos que, para esse sistema, foi integrada com a biblioteca *React*. Essa combinação foi escolhida para facilitar a construção da aplicação, facilitar a atualização em tempo real da página web enquanto está sendo realizado o desenvolvimento, eliminando a necessidade de se recarregar toda a página para visualizar alterações realizadas.

Com isso, foram desenvolvidas interfaces com o objetivo de proporcionar uma boa experiência ao usuário. Essas interfaces incluem uma tela de início da aplicação, que é a primeira tela que o usuário visualiza ao entrar na aplicação; a tela de login, que o usuário usa para acessar a sua conta; a tela de visualização de filmes ou séries, que apresentam informações desses itens; a tela de perfil do usuário, onde ele visualiza e configura os seus dados; e a tela de busca, onde é

retornado resultados para o que o usuário informa na barra de pesquisa. Todas as telas são descritas e apresentadas com mais detalhes na próxima seção.

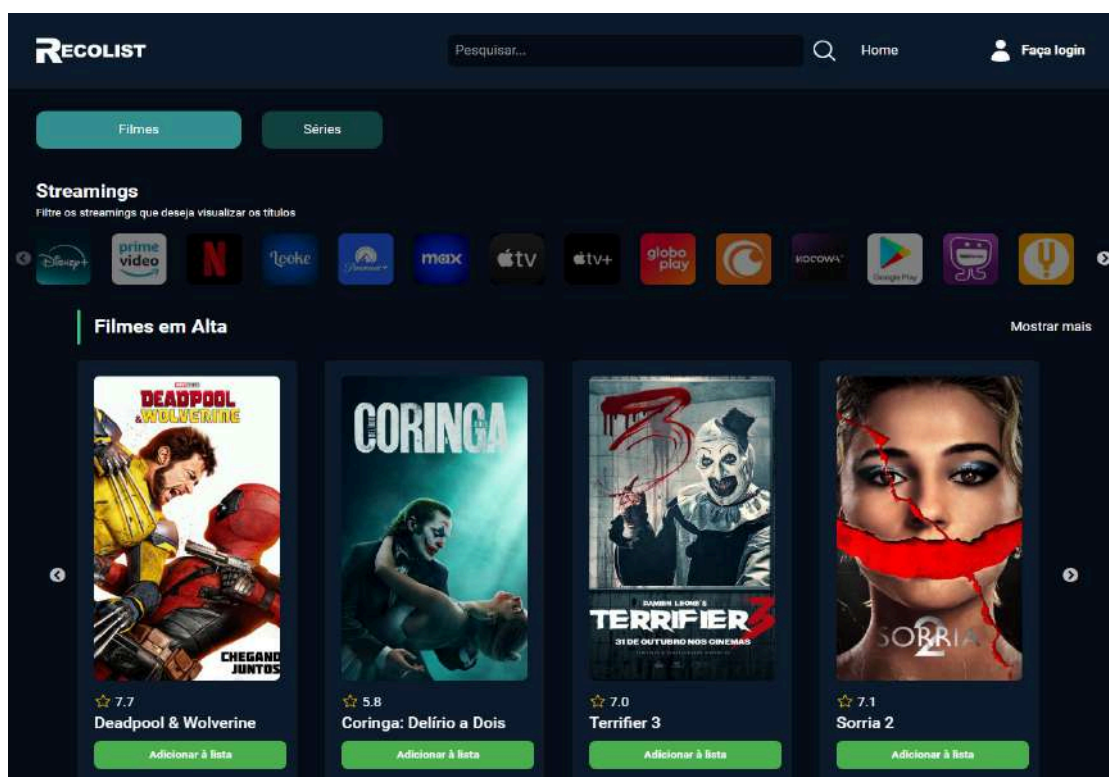
3.6 Páginas da aplicação

Esta seção apresenta as páginas desenvolvidas para visualização do sistema, demonstrando a interface do usuário e suas funcionalidades. Serão descritas as telas, mostrando como o usuário pode interagir com os recursos oferecidos pela aplicação.

3.6.1 Página inicial

A página inicial, que é mostrada na Figura 8, é a primeira página que o usuário visualiza ao entrar no sistema. Nela, é possível acessar a maioria das funcionalidades disponibilizadas para os usuários como filtrar por streaming, adicionar um item à lista do usuário e filtrar por filme ou série.

Figura 8 - Página inicial

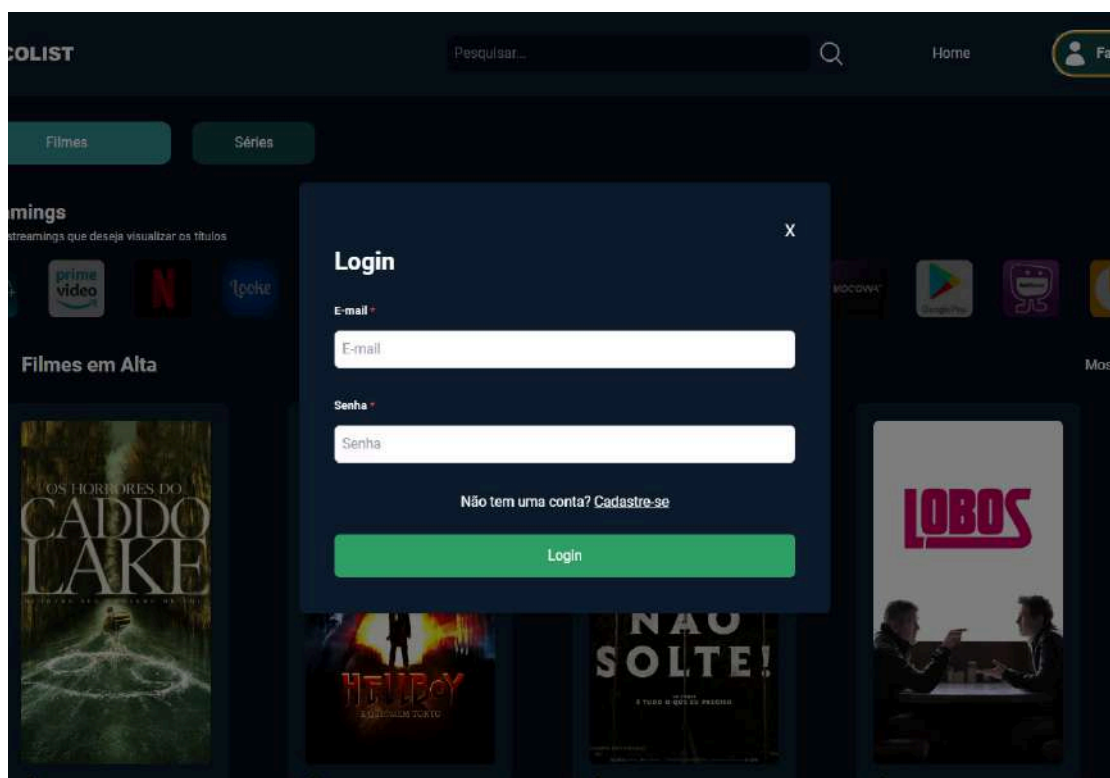


Fonte: Autor

3.6.2 Página de *Login* e cadastro

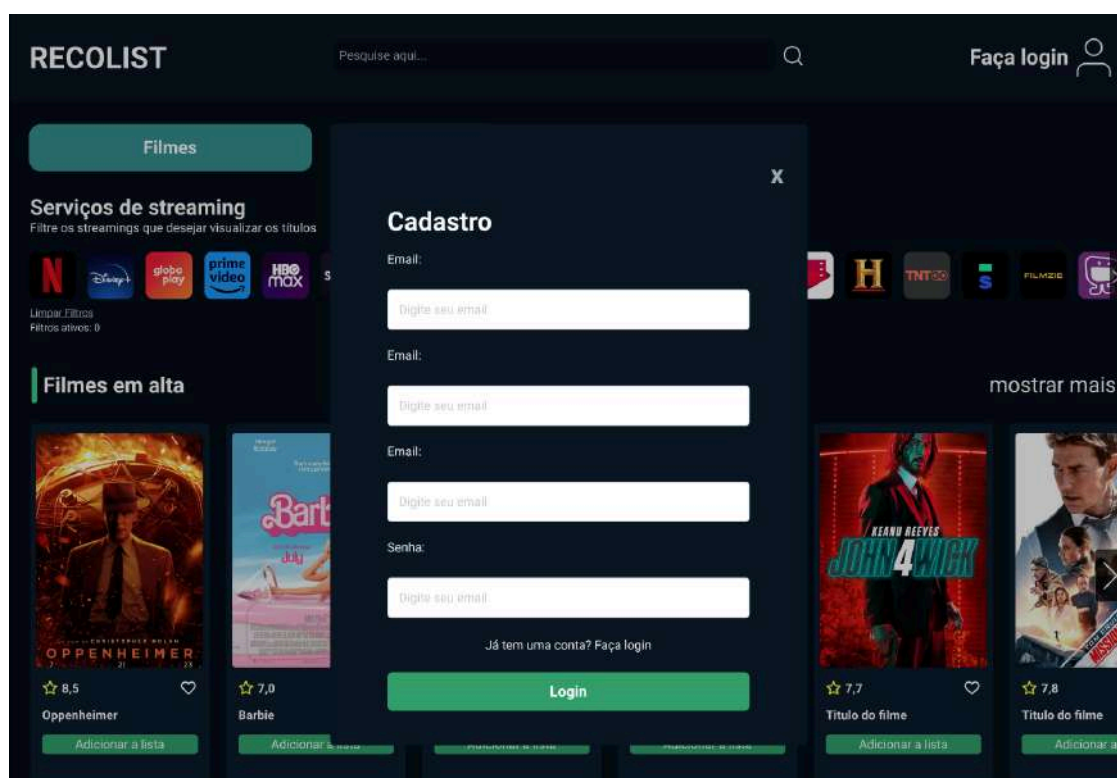
Em todas as páginas do sistema, o usuário pode visualizar a opção para fazer login ou realizar um cadastro. Quando ele escolhe uma dessas opções, é aberta uma página de *login* (Figura 9) e o usuário deverá informar suas credenciais de acesso. Caso o usuário ainda não tenha cadastro, ele pode fazer a sua inscrição acessando a página de cadastro, que é exibida na Figura 10. Nesse caso, o usuário deve preencher o formulário de informações necessárias para criar uma conta. Após realizar um desses procedimentos, o usuário deverá clicar no botão *login* ou *cadastro*, a depender do formulário e poderá iniciar uma sessão.

Figura 9 - Página de login



Fonte: Autor

Figura 10 - Página de cadastro



Fonte: Autor

3.6.3 Página de visualização de filme ou série

Quando um filme ou série é selecionado pelo usuário, a página de detalhes do título é exibida, apresentando informações detalhadas sobre o mesmo. Para o desenvolvimento da página de detalhes (Figura 11), foi utilizada a biblioteca *color-thief* do *React*, com ela, é possível identificar a cor predominante na página e aplicá-la ao fundo. Nela, são informados a classificação indicativa de idade, os gêneros presentes, a duração, os *streamings* onde contém o título, a avaliação geral, a sinopse, os diretores, o elenco e as tags. Além disso, o usuário pode marcar o filme como visto e adicionar o filme a uma lista personalizada ou a lista de favoritos. Abaixo dessas informações (Figura 12), há a possibilidade do usuário realizar uma avaliação e visualizar as que foram feitas por outros usuários, por fim, são listadas recomendações baseadas no título.

Figura 11 - Página de visualização de filme ou série

The screenshot shows a movie page for "Forrest Gump: O Contador de Histórias (1994)". On the left is a poster of Forrest Gump sitting on a bench. To the right, the title is displayed in white on a dark blue background. Below the title, there is a rating of 14, genres (Comédia - Drama - Romance), and a duration of 2h e 22 minutos. A red 'N' icon is visible. A green button says "Adicionar à lista". Below that is a star rating of 8.5. The synopsis is in Portuguese, mentioning Forrest Gump's life and the Vietnam War. The director is listed as Robert Zemeckis. At the bottom, there is a section for the cast ("Elenco") with three actor portraits and a "Tags" section with labels like "vietnam war", "vietnam veteran", "mentally disabled", "friendship", "usa president", "washington dc, usa", and "post-traumatic stress disorder (ptsd)".

Fonte: Autor

Figura 12 - Avaliações e relacionados de filme ou série

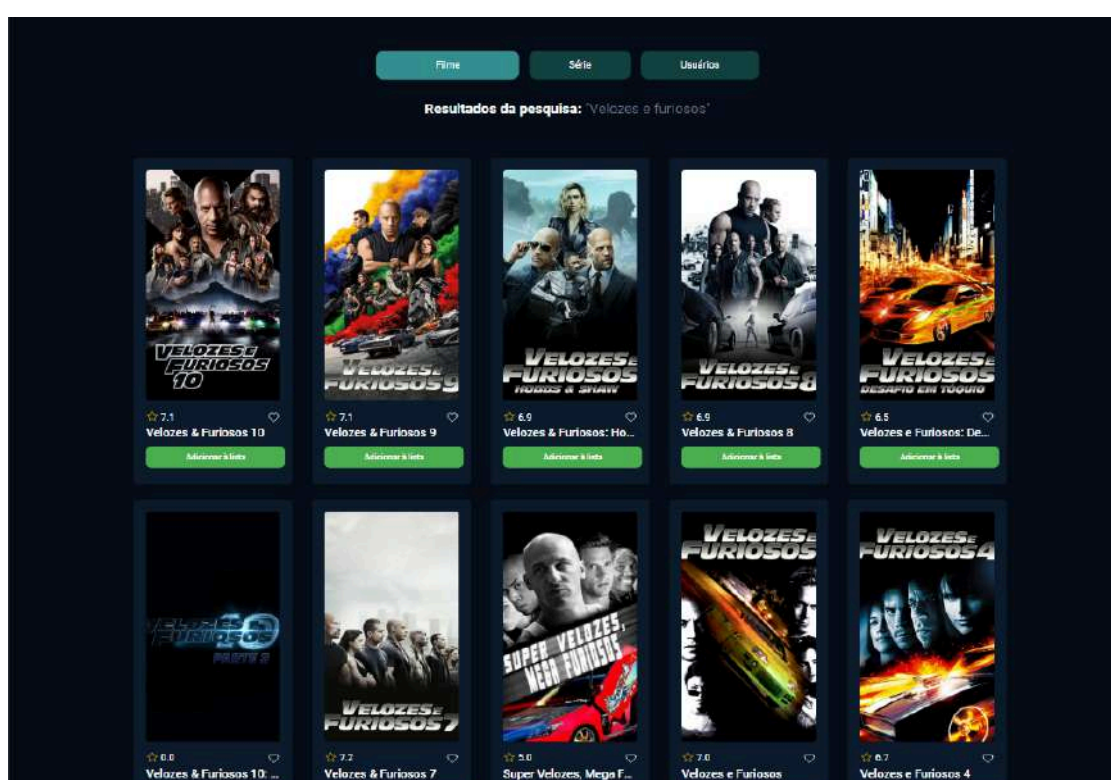
The screenshot shows a section of a movie page with reviews and recommendations. At the top, it says "Confira o que os usuários estão achando desse filme". There are buttons for "Avaliar filme", "Avaliações públicas", and "Avaliações de amigos". Under "Sua Avaliação", there is a review by "Vooit" dated 16/12/2024 with a 5-star rating. The review text is partially visible. Below that, there are "Avaliações Públicas" with a review by "eiz" dated 10/10/2024 with a 5-star rating. At the bottom, there is a "Recomendados" section with five movie posters: "The Godfather", "The Godfather Part II", "The Godfather Part III", "Pulp Fiction", and "Coração Valente".

Fonte: Autor

3.6.4 Página de pesquisa

Quando o usuário realiza uma pesquisa por itens do seu interesse, utilizando palavras-chave, é exibida uma página (Figura 13) com as seguintes opções de filtro: “filme” “série” ou “usuário”. Como resultado, são retornados os itens que satisfazem os critérios de seleção estabelecidos na consulta, juntamente com os dados sobre a pesquisa que foi realizada.

Figura 13 - Página de resultados de pesquisa

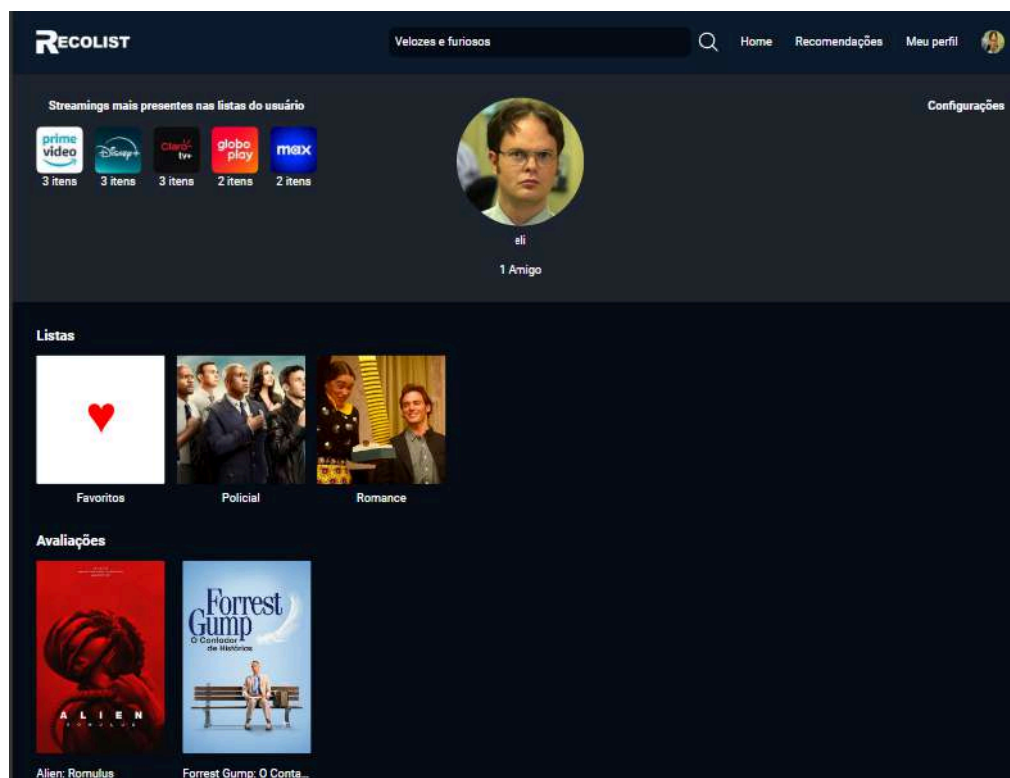


Fonte: Autor

3.6.5 Página de perfil do usuário logado

Os usuários podem visualizar o seu próprio perfil bem como o perfil de outros usuários. Conforme exibido na Figura 14, nesses perfis eles podem visualizar informações como os serviços de streaming mais presentes nas listas, foto de perfil, quantidade de pessoas que o usuário segue e o seu número de seguidores, além de listas criadas e avaliações realizadas. Se o usuário estiver visualizando seu próprio perfil, ele poderá gerenciar suas listas e seu cadastro, editando privacidade, nome de usuário, e-mail e senha.

Figura 14 - Perfil do usuário logado



Fonte: Autor

5. CONCLUSÃO

Atualmente, a internet é amplamente utilizada para oferecer uma variedade de filmes e séries através de plataformas de *streaming*, levando ao surgimento de diversas dessas plataformas. Como resultado, a quantidade de serviços de *streaming* disponíveis e a exclusividade de suas produções tem causado a fragmentação desses serviços. Isso torna difícil para muitos usuários encontrar o que assistir relacionado a filmes ou séries nos serviços que ele já assina.

Para ajudar a superar essas limitações, este TCC propõe um sistema web que auxilia os usuários na hora de escolher um filme ou série por meio de recomendações com base em suas preferências. Além disso, ele ajuda o usuário a identificar os serviços de streaming que ele deve priorizar a assinatura, informando quais títulos de interesse do usuário estão disponíveis em cada plataforma disponível.

A implementação do sistema atendeu todos os requisitos e funcionalidades propostos, garantindo sua utilidade ao auxiliar o usuário na tomada de decisão em

relação à escolha de qual filme ou série assistir, oferecendo uma organização dos títulos disponíveis e plataformas de streaming que devem ser priorizadas.

O trabalho pode ser estendido por futuras implementações. Um possível trabalho futuro consiste em incluir novos serviços de streaming que contenham outros gêneros como animes, novelas e documentários, ampliando o acesso do usuário a títulos de interesse em diferentes serviços. Além disso, a implementação de uma funcionalidade para exibição de trailers, permitindo que o usuário visualize prévias de filmes ou séries antes de decidir o que assistir. Outra possibilidade seria o desenvolvimento de um aplicativo móvel para proporcionar uma experiência aprimorada para os usuários que utilizarem o sistema em smartphones

REFERÊNCIAS

BANKER, K. et al. **MongoDB in Action**: Covers MongoDB version 3.0. 2. ed. Lewis St.: Manning, 15 abr. 2016. v. 3. 480 p. (MongoDB in Action, v. 3). ISSN 1063-8016.

ISBN 1617291609. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=&id=kzkzEAAAQBAJ&oi=fnd&pg=PT21&dq=mongodb&ots=8V0TtXa339&sig=9WhVBmDnybwosJNyHvxTfXVNrAE&redir_esc=y#v=onepage&q=mongodb&f=false. Acesso em: 23 nov. 2023.

NIELSEN. **Metadados de mídia**: A peça essencial para o sucesso no streaming.

set. 2023. Disponível em: <https://www.nielsen.com/pt/insights/2023/media-metadata-success-in-streaming/>. Acesso em: 02 out. 2023.

RODRIGUES, C. G. A. L. et al. **MovieIub**: a rede social de filmes. 07 dez. 2021. 50 p. Monografia (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) — FACULDADE DE TECNOLOGIA DE AMERICANA “MINISTRO RALPH BIASI”. Disponível em: <https://ric.cps.sp.gov.br/handle/123456789/12839>. Acesso em: 06 nov. 2023.


SILVA, G. C. R. da. **WhatToWatch**: sistema web de recomendação de filmes e séries. 02 fev. 2023. 80 p. Monografia (Bacharelado em Sistemas de Informação) —

Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, Campus Bauru. Disponível em: <https://repositorio.unesp.br/items/a43f5912-5dd9-4a8e-8b5f-7b47794f5b15>. Acesso em: 04 set. 2023.

SOMMERVILLE, I. **Software Engineering**. 9. ed. Pearson University, 2011/11/15. v. 1.

529 p. (9, v. 1). ISBN 8579361087. Disponível em: https://books.google.com.br/books/about/Engenharia_de_software.html?hl=pt-BR&id=H4u5ygAACAAJ&redir_esc=y. Acesso em: 11 nov. 2023.

SPIPKER, Hendrik Storstein; COLBJØRNSSEN, Terje. **The dimensions of streaming:** toward a typology of an evolving concept. Media, Culture & Society. United Kingdom, 27 fev. 2020 1210-1225 p. Disponível em: <https://journals.sagepub.com/doi/full/10.1177/0163443720904587>. Acesso em: 09 set. 2023.

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Cajazeiras - Código INEP: 25008978
	Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
	CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

Documento Digitalizado Ostensivo (Público)

TCC - Elivelton Pereira da Silva

Assunto:	TCC - Elivelton Pereira da Silva
Assinado por:	Elivelton Silva
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Elivelton Pereira da Silva, DISCENTE (202122010002) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS, em 22/10/2024 18:08:30.

Este documento foi armazenado no SUAP em 22/10/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1287896

Código de Autenticação: 2b0d7f74a7

