

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO



**EXPLORANDO O DESENVOLVIMENTO FRONT-END: UM ESTUDO DE CASO EM
CATÁLOGO DE FILMES**

TAYNÁ RODRIGUES DE SOUSA CARVALHO

GOIÂNIA
2023

TAYNÁ RODRIGUES DE SOUSA CARVALHO

**EXPLORANDO O DESENVOLVIMENTO FRONT-END: UM ESTUDO DE CASO EM
CATÁLOGO DE FILMES**

Trabalho de Conclusão de Curso apresentado à
Escola Politécnica e de Artes, da Pontifícia
Universidade Católica de Goiás, como parte dos
requisitos para a obtenção do título de Bacharel
em Ciência da Computação.

Orientador(a):

Prof. Dr. José Luiz de Freitas Junior

Banca examinadora:

Prof. Me. Eugênio Júlio Messala Cândido
Carvalho

Prof. Me. Marcio Giovane Cunha Fernandes

GOIÂNIA

2023

TAYNÁ RODRIGUES DE SOUSA CARVALHO

**EXPLORANDO O DESENVOLVIMENTO FRONT-END: UM ESTUDO DE CASO EM
CATÁLOGO DE FILMES**

Trabalho de Conclusão de Curso aprovado em sua forma final pela Escola
Politécnica e de Artes, da Pontifícia Universidade Católica de Goiás, para obtenção
do título de Bacharel em Ciência da Computação, em ____/____/_____.

Orientador: Prof. Dr. José Luiz de Freitas Junior

Prof. Me. Eugênio Júlio Messala Cândido Carvalho

Prof. Me. Marcio Giovane Cunha Fernandes

GOIÂNIA

2023

A Deus pela minha vida e oportunidades.

Aos meus pais, pelo apoio incondicional.

Ao Eliel pelas noites em claro.

AGRADECIMENTOS

Ao Professor José Luiz de Freitas Junior, orientador acadêmico, pelo apoio e paciência.

À Coordenação da Escola de Ciências Exatas e da Computação, da Pontifícia Universidade Católica de Goiás por ajudar, de forma inestimável, na execução deste projeto.

À Norven pela sugestão do tema e investimento financeiro e de tempo em minha carreira.

A todos que direta ou indiretamente colaboraram para materialização deste trabalho.

LISTA DE ILUSTRAÇÕES

- Figura 1 - Diagrama de caso de uso
- Figura 2 - CSU 001: Cadastrar
- Figura 3 - CSU 002: Fazer Login
- Figura 4 - CSU 003: Explorar Conteúdo
- Figura 5 - CSU 004: Ver Galeria de Conteúdo
- Figura 6 - CSU 005: Buscar
- Figura 7 - CSU 006: Filtrar
- Figura 8 - CSU 007: Ordenar
- Figura 9 - CSU 008: Ver Detalhes
- Figura 10 - CSU 009: Marcar como favorito
- Figura 11 - CSU 010: Marcar como assistido
- Figura 12 - Modelo entidade-relacionamento ilustrativo
- Figura 13 - Modelo entidade-relacionamento
- Figura 14 - Logotipo *movie*
- Figura 15 - Paleta de cores
- Figura 16 - Grid de 8 pontos
- Figura 17 - Telas Figma detalhes do chamado
- Figura 18 - Componentes Figma
- Figura 19 - Requisições no Axios
- Figura 20 - Estrutura da Store
- Figura 21 - Chamadas da Store
- Figura 22 - *Controllers* Favorite
- Figura 23 - *Routes* Favorite
- Figura 24 - *Login* e cadastro
- Figura 25 - Tela Inicial
- Figura 26 - Seção Hero

LISTA DE TABELAS

Tabela 1 - DD 001: Dados do usuário

Tabela 2 - DD 002: Dados dos títulos favoritos

Tabela 3 - DD 003: Dados dos títulos assistidos

LISTA DE QUADROS

Quadro 1 - RF 001: Fazer Login

Quadro 2 - RF 002: Cadastrar

Quadro 3 - RF 003: Explorar Conteúdo

Quadro 4 - RF 004: Ver Listagem de Conteúdo

Quadro 5 - RF 005: Ver Detalhes

Quadro 6 - RF 006: Buscar

Quadro 7 - RF 007: Filtrar

Quadro 8 - RF 008: Ordenar

Quadro 9 - RF 009: Marcar como favorito

Quadro 10 - RF 010: Marcar como assistido

Quadro 11 - RN 001: Requisições à API otimizada

Quadro 12 - RN 002: Conexão com banco de dados eficiente

Quadro 13 - RN 003: Criptografia de informações sensíveis

Quadro 14 - RN 004: Fácil expansão de funcionalidades futuras

Quadro 15 - RN 005: Integridade e consistência dos dados em casos de recarregamento da página

Quadro 16 - RN 006: Sincronização eficiente entre a persistência local e o banco de dados

Quadro 17 - RN 007: Interface do usuário deve ser intuitiva

Quadro 18 - RN 008: Comunicação de ações disponíveis para usuários não autenticados

Quadro 19 - RN 009: Compatibilidade com os principais navegadores web e dispositivos móveis

LISTA DE SIGLAS

TMDb	<i>The Movie Database</i> (Base de Dados o Filme)
IMDb	<i>Internet Movie Database</i> (Base de Dados Filme da Internet)
UX	<i>User experience</i> (Experiência do Usuário)
UI	<i>User Interface</i> (Interface do Usuário)
HTML	<i>HyperText Markup Language</i> (Linguagem de marcação de Hipertexto)
CSS	<i>Cascading Style Sheets</i> (Folha de Estilo em cascata)
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicação)
JS	Java Script
SQL	<i>Structured Query Language</i> (Linguagem de Pesquisa Estruturada)
DDL	<i>Data Definition Language</i> (Linguagem de Definição de Dados)
DML	<i>Data Manipulation Language</i> (Linguagem de Manipulação de Dados)
npm	<i>Node Package Manager</i> (Gerenciador de Pacotes Node)
TCC	Trabalho de conclusão de curso

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	APRESENTAÇÃO	14
1.2	OBJETIVO	15
1.3	JUSTIFICATIVA.....	15
1.4	SÍNTESE.....	15
2	REFERENCIAL TEÓRICO.....	16
2.1	ESTUDO DE FERRAMENTAS SIMILARES	16
2.1.1	Netflix.....	16
2.1.2	Prime Video	17
2.1.3	HBO Max	17
2.1.4	The Movie Database (TMDb)	18
2.1.5	Letterboxd.....	19
2.1.6	IMDb (Internet Movie Database)	19
2.2	CONCEITOS DE DESIGN.....	20
2.2.1	Teoria das cores no desenvolvimento web.....	20
2.2.2	Experiência do Usuário (UX) e Interface do Usuário (UI)	20
2.3	FERRAMENTAS E TECNOLOGIAS.....	21
2.3.1	Grid de 8 pontos	21
2.3.2	Figma.....	21
2.3.3	HTML5, CSS3 e JavaScript	22
2.3.4	Vue3, Vuex e Vue Router	23
2.3.5	Options API.....	24
2.3.6	Node JS	25
2.3.7	API do The Movie DB.....	25
2.3.8	My SQL.....	26
2.3.9	AXIOS.....	26

2.3.10	Netlify.....	27
2.3.11	PlanetScale	28
3	MODELAGEM DO SISTEMA.....	28
3.1	DESCRIÇÃO GERAL DO SISTEMA	28
3.2	REQUISITOS	29
3.3	DIGRAMA DE CASO DE USO	39
3.4	DIAGRAMAS DE SEQUÊNCIA	39
3.5	BANCO DE DADOS	45
3.5.1	MER.....	45
3.5.2	Descrição de dados do sistema	46
3.6	PROTOTIPAGEM.....	47
3.6.1	Identidade Visual.....	47
3.6.2	Sistema de Grid de 8pts.....	48
3.6.3	Projeto no Figma.....	49
4	DESENVOLVIMENTO DO SOFTWARE	51
4.1	BACK-END E ARMAZENAMENTO DE DADOS.....	51
4.2	IMPLEMENTAÇÃO FRONT-END	55
5	CONSIDERAÇÕES FINAIS	57
5.1	LIÇÕES APRENDIDAS	57
5.2	DIFICULDADES ENCONTRADAS	58
5.3	IMPLEMENTAÇÕES FUTURAS.....	58
5.4	CONCLUSÃO.....	59
6	REFERENCIAS	60

1 RESUMO

Este Trabalho de Conclusão de Curso aborda o desenvolvimento front-end de um catálogo de filmes web, explorando boas práticas e tecnologias atuais no campo da programação web, focando no desenvolvimento do front-end. O objetivo central foi criar uma plataforma digital intuitiva e atraente que oferecesse uma experiência de usuário superior. Para isso, foi realizada uma análise de plataformas de streaming existentes, estudando-se aspectos como design, funcionalidades, usabilidade e interatividade.

A metodologia empregada incluiu o uso de tecnologias contemporâneas no desenvolvimento web, como Vue3, Vuex, Vue Router, HTML5, CSS3 e JavaScript, além da integração com a API do The Movie Database para acesso a dados atualizados de filmes e a utilização do MySQL para armazenamento dos dados do usuário. O projeto enfatizou a importância de uma interface amigável, com foco na experiência do usuário e na facilidade de navegação.

As descobertas principais ressaltaram a relevância de uma arquitetura modular, design intuitivo, e a importância de considerar a usabilidade e a estética na criação de soluções web. Além disso, o estudo sublinhou a necessidade de uma integração eficiente entre diferentes tecnologias para garantir uma experiência de usuário coesa e envolvente.

Em conclusão, este TCC demonstra como a combinação de práticas de programação avançadas e um design centrado no usuário pode resultar em plataformas digitais eficazes e atraentes, particularmente no contexto de catálogos de filmes web. Este projeto não só fornece insights valiosos sobre o desenvolvimento web, mas também estabelece um modelo para futuras iniciativas que visam melhorar a experiência do usuário no ambiente digital.

Palavras-chave: *front-end*, componentização.

ABSTRACT

This Bachelor's Thesis refers to the front-end development of a web movie catalog, exploring the best practices and current technologies in the field of web programming. The main goal was to create an intuitive and engaging digital platform that provides a superior user experience. To achieve this, an in-depth analysis of existing streaming platforms was conducted, studying aspects such as design, features, usability, and interactivity.

The employed methodology included the use of contemporary web development technologies, such as Vue3, Vuex, Vue Router, HTML5, CSS3, and JavaScript, along with integration with The Movie Database API for access to up-to-date movie data. The project emphasized the importance of a user-friendly interface, focusing on user experience and ease of navigation.

The key findings highlighted the relevance of modular architecture, intuitive design, and the importance of considering usability and aesthetics in creating web solutions. Additionally, the study underscored the need for efficient integration among different technologies to ensure a cohesive and engaging user experience.

In a nutshell, this Bachelor's Thesis demonstrated how the combination of advanced programming practices and user-centered design can result in effective and attractive digital platforms, especially in the context of web movie catalogs. This project not only provided valuable insights into modern web development but also established a model for future initiatives aiming to enhance the user experience in the digital environment.

Keywords: front-end, componentization.

1 INTRODUÇÃO

A era digital, caracterizada por avanços tecnológicos sem precedentes, transformou radicalmente a maneira como interagimos com o mundo ao nosso redor. Esta revolução contínua no domínio da computação tem suas raízes na evolução histórica da tecnologia da informação, desde os primeiros computadores até as modernas interfaces de usuário. O desenvolvimento de interfaces de usuário, em particular, emergiu como uma disciplina crucial, moldando de maneira significativa a forma como consumimos e compartilhamos informações online (Tidwell, 2011; Tegmark, 2018).

Este trabalho dedica-se ao campo do desenvolvimento front-end, uma área que desempenha um papel vital na criação de experiências digitais envolventes e intuitivas. A escolha deste caso visa não apenas compreender os princípios técnicos subjacentes, mas também capturar a essência desta interseção fascinante, onde a programação encontra a arte (Robbins, 2007).

Ao explorar a construção de um catálogo de filmes, não se limitou apenas à análise de linhas de código e frameworks populares. Buscou-se desvendar as decisões de design, considerações de usabilidade e nuances que transformam uma aplicação digital em uma experiência envolvente. Este trabalho aspira a documentar não apenas um projeto de desenvolvimento *front-end*, mas também a revelar os desafios, aprendizados e insights que emergem ao criar uma interface destinada a cativar os amantes do cinema.

Este estudo de caso celebra não apenas o conhecimento técnico acumulado ao longo dos anos, mas também a paixão por moldar o futuro digital. Ao final desta jornada, almeja-se não apenas construir um catálogo de filmes funcional, mas também contribuir para a compreensão coletiva de como a Ciência da Computação pode enriquecer a experiência humana, um código por vez.

1.1 APRESENTAÇÃO

O ponto de partida desta pesquisa é a seguinte indagação fundamental: "Como realizar um desenvolvimento web, utilizando boas práticas de programação *front-end* e recursos atuais?" Este questionamento reflete a crescente importância do desenvolvimento *front-end*, uma disciplina que exige práticas seguras e é capaz de

proporcionar experiências únicas aos usuários. A resposta a essa questão será buscada através do desenvolvimento prático de um catálogo de filmes web.

A escolha estratégica desse projeto não apenas reflete uma demanda atual do mercado digital, mas também permite uma análise aprofundada do processo de desenvolvimento front-end.

1.2 *OBJETIVO*

O objetivo principal desta pesquisa é explorar as melhores práticas de programação e os recursos tecnológicos mais atuais no contexto do desenvolvimento web, aplicando-os em um projeto prático. Esta exploração envolve a investigação, análise e compreensão das melhores práticas de desenvolvimento web, a sua aplicação em um projeto real (o desenvolvimento de um catálogo de filmes web) e, finalmente, a avaliação dos resultados obtidos, considerando as práticas de programação e os recursos tecnológicos utilizados.

1.3 *JUSTIFICATIVA*

A escolha do catálogo de filmes como objeto de estudo é fundamentada em sua relevância na era digital, onde a busca, recomendação e visualização de filmes e séries se tornaram atividades comuns e amplamente realizadas online. O fascínio contínuo do público pelo universo cinematográfico, intensificado pela ascensão das plataformas de streaming como Netflix, Amazon Prime e Disney+, evidencia a necessidade de ferramentas que auxiliem na escolha e organização de conteúdo.

Um catálogo de filmes web se apresenta como uma solução ideal, oferecendo uma interface amigável e recursos que facilitam a busca e categorização de títulos. Ele se apresenta como uma solução eficaz em que o usuário consegue ter uma visão mais ampla de títulos de diferentes plataformas, todos em apenas um lugar.

Além disso, este projeto possibilita uma exploração mais profunda do processo de desenvolvimento web, proporcionando insights valiosos sobre a aplicação das melhores práticas de programação e a utilização de recursos tecnológicos atuais em um contexto real.

1.4 *SÍNTESE*

Esta introdução delinea a visão geral dos objetivos do estudo, destacando os motivos que conduziram a escolha do catálogo de filmes como estudo de caso e a

metodologia utilizada. Ela estabelece a estrutura do trabalho, que abordará os principais aspectos do desenvolvimento web e suas aplicações práticas neste contexto específico. Através dessa abordagem abrangente, pretende-se contribuir significativamente para o entendimento coletivo de como a Ciência da Computação pode moldar e aprimorar a experiência humana na era digital.

2 REFERENCIAL TEÓRICO

2.1 ESTUDO DE FERRAMENTAS SIMILARES

Neste capítulo, será realizada uma análise aprofundada de diversas plataformas de streaming de vídeo que serviram como referência para o desenvolvimento do catálogo de filmes web proposto. Cada plataforma será minuciosamente explorada quanto à sua finalidade, funcionalidades principais, interfaces e experiência do usuário.

2.1.1 Netflix

A Netflix (2023), líder no segmento de streaming, destaca-se não apenas pela sua vasta biblioteca de filmes, séries, documentários e produções originais, mas também pelo seu compromisso em proporcionar uma experiência de usuário excepcional. A plataforma é meticulosamente projetada para ser intuitiva, garantindo que os usuários não apenas encontrem o que desejam assistir, mas também descubram novos conteúdos alinhados aos seus gostos e interesses.

Ao entrar na página inicial da Netflix, o design refinado e as categorias meticulosamente organizadas capturam a atenção, apresentando uma variedade de títulos que vão desde os mais aclamados até sugestões personalizadas. Esta personalização é aprimorada pelo avançado algoritmo da Netflix, que, com base no histórico e preferências do usuário, propõe conteúdos alinhados aos seus gostos (Netflix, 2023).

A plataforma ainda permite a criação de diversos perfis, uma característica valiosa para famílias ou compartilhamentos de conta, já que cada perfil desfruta de recomendações, listas e configurações únicas, assegurando uma vivência singular para cada usuário. A Netflix (2023) ainda enriquece a experiência do usuário com recursos como modo offline, avaliações e trailers integrados, tornando a seleção de conteúdos uma jornada envolvente e gratificante.

2.1.2 Prime Video

O Prime Video (2023), plataforma de streaming desenvolvida pela Amazon, destaca-se pela diversidade de seu catálogo, que abrange filmes, séries e produções originais, além de proporcionar uma experiência de usuário refinada. A plataforma é projetada com interfaces limpas, organizadas e intuitivas, permitindo uma navegação suave e agradável, refletindo o compromisso da Amazon em oferecer um serviço de alta qualidade e centrado no usuário.

Ao interagir com o Prime Video, os usuários encontram uma variedade de categorias e recomendações, facilitando a exploração do vasto catálogo da plataforma. O algoritmo de recomendação do Prime Video é inteligente e adaptativo, sugerindo títulos com base nas preferências e no comportamento de visualização do usuário, otimizando a descoberta de novos conteúdos (Prime Video, 2023).

Uma característica distintiva do Prime Video (2023) é a possibilidade de baixar títulos, permitindo aos usuários desfrutarem de seus conteúdos favoritos mesmo sem conexão à internet, ideal para viagens ou para acessar os conteúdos em locais com sinal de internet instável.

Além disso, a plataforma disponibiliza funcionalidades como controle parental, que garante um ambiente seguro para todos os públicos, e a personalização de perfis, que permite a criação de experiências individualizadas para cada usuário. A integração de avaliações, comentários e informações adicionais sobre os títulos enriquece ainda mais a experiência do usuário, auxiliando na seleção de conteúdos e proporcionando um ambiente interativo e informativo (Prime Video, 2023).

2.1.3 HBO Max

A HBO Max (2023), emergindo como uma das plataformas de streaming mais prestigiadas, é resultado da fusão do aclamado conteúdo da HBO com uma ampla gama de títulos de diversas marcas renomadas sob a égide da WarnerMedia. O que realmente diferencia a HBO Max no cenário de streaming é a sua dedicação inabalável à produção de conteúdo original de alta qualidade, juntamente com uma coleção diversificada que atende a uma ampla gama de públicos e gostos.

Ao mergulhar na interface da HBO Max, os usuários são recebidos por um design elegante e funcionalidades pensadas para maximizar a experiência de visualização. A plataforma utiliza algoritmos avançados para fornecer listas de visualização personalizadas, garantindo que os usuários sempre tenham algo novo e

relevante para assistir. As recomendações são meticulosamente curadas com base nas preferências e histórico de visualização do usuário, tornando a descoberta de novos títulos uma jornada emocionante e personalizada (HBO Max, 2023).

Além disso, a HBO Max (2023) entende a importância da flexibilidade, oferecendo a opção de baixar conteúdo, o que permite aos usuários desfrutarem de seus programas e filmes favoritos em movimento ou em áreas que não possuem conexão com a internet. O controle parental é outra funcionalidade essencial, proporcionando aos pais a tranquilidade de saber que seus filhos estão acessando conteúdo apropriado para sua idade. Combinando todas essas características, a HBO Max não apenas entrega conteúdo de qualidade, mas também uma experiência de usuário impecável e adaptada às necessidades de cada indivíduo.

2.1.4 The Movie Database (TMDb)

Emergindo como uma das bases de dados mais colaborativas e abrangentes, o The Movie Database (TMDb) é uma comunidade global dedicada a catalogar e discutir filmes e programas de televisão. O que realmente diferencia o TMDb no cenário de bases de dados é sua natureza comunitária, permitindo que entusiastas do cinema de todo o mundo contribuam, enriquecendo a plataforma com informações detalhadas e insights sobre uma vasta gama de títulos (TMDb, 2023).

Ao explorar o TMDb (2023), os usuários são recebidos por uma interface visualmente atraente e intuitiva, facilitando a descoberta e pesquisa de títulos. Cada entrada é meticulosamente detalhada, oferecendo desde sinopses até trailers, imagens e avaliações. A capacidade de contribuir e editar informações torna a experiência no TMDb única, permitindo que os membros se sintam parte integrante da comunidade.

Para aqueles com inclinações técnicas, a API robusta do TMDb oferece oportunidades ilimitadas de integração e desenvolvimento. Além disso, a plataforma promove a interatividade, permitindo que os usuários criem listas personalizadas e se conectem a redes sociais para compartilhar suas paixões cinematográficas. A plataforma também disponibiliza sua API para ser utilizada em diferentes aplicações. Em essência, o TMDb não é apenas um catálogo, é um espaço onde a paixão pelo cinema e pela televisão é celebrada, discutida e compartilhada (TMDb, 2023).

2.1.5 Letterboxd

O Letterboxd (2023), posicionando-se como uma intersecção única entre rede social e catálogo cinematográfico, é o refúgio dos aficionados por cinema. O que realmente distingue o Letterboxd no mundo dos catálogos de filmes é sua abordagem comunitária e social, permitindo que os amantes do cinema registrem, discutam e celebrem sua paixão por filmes de todas as eras e gêneros.

Ao entrar no universo do Letterboxd, os usuários são imediatamente imersos em uma experiência rica e colaborativa. A plataforma permite que os membros cataloguem os filmes que assistiram, anotem suas impressões e atribuam classificações, criando assim um diário cinematográfico pessoal (Letterboxd, 2023).

Além disso, os usuários podem criar e explorar listas temáticas, desde "Os melhores filmes *noir*" até "Recomendações para um dia chuvoso". A interatividade é amplificada pela capacidade de seguir outros entusiastas, descobrindo novos filmes através de suas avaliações, listas e discussões (Letterboxd, 2023).

O Letterboxd (2023) também se destaca por sua abordagem comunitária, com debates animados e análises aprofundadas sobre títulos específicos, permitindo uma troca rica de perspectivas e insights. Em resumo, o Letterboxd é mais do que um simples catálogo, é uma comunidade vibrante onde a arte do cinema é apreciada, analisada e celebrada em todas as suas nuances.

2.1.6 IMDb (Internet Movie Database)

O IMDb (2023), reconhecido mundialmente como a autoridade definitiva em informações cinematográficas, serve como uma enciclopédia abrangente para filmes, séries e personalidades da indústria. O que realmente destaca o IMDb no universo dos catálogos de filmes é sua vastidão e profundidade de conteúdo, combinadas com uma rica tapeçaria de avaliações e discussões da comunidade global de cinéfilos.

Ao navegar pelo IMDb (2023), os usuários são apresentados a um tesouro de informações. Cada título listado na plataforma é acompanhado por uma série de detalhes meticulosos, desde sinopses, elenco e equipe técnica, até curiosidades, trilhas sonoras e histórias de bastidores.

Uma das características mais valiosas é a seção de avaliações e críticas, onde os membros podem expressar suas opiniões e debater nuances de produções

específicas. Além disso, o IMDb oferece listas curadas, como "Top 250" ou "Os mais populares", guiando os usuários através do vasto mundo do cinema (IMDb, 2023).

A seção "IMDbPro" é um recurso adicional para profissionais da indústria, fornecendo insights sobre tendências de bilheteria, contatos e eventos relevantes. A plataforma também permite que os usuários criem listas personalizadas, rastreiem suas visualizações e se conectem com outros entusiastas. Em essência, o IMDb não é apenas um catálogo, é um universo onde a paixão pelo cinema é documentada, discutida e celebrada em detalhes meticulosos (IMDb, 2023).

2.2 *CONCEITOS DE DESIGN*

2.2.1 **Teoria das cores no desenvolvimento web**

A teoria das cores é uma área de estudo que se dedica a entender como percebemos as cores, as emoções e significados que elas podem transmitir. No contexto do desenvolvimento web, a cor desempenha um papel crucial na comunicação visual, influenciando não apenas a estética, mas também a funcionalidade e a experiência do usuário (Beaird, 2020).

Cores têm o poder de evocar sentimentos, criar foco e guiar a jornada do usuário em um site. Por exemplo, o vermelho é frequentemente associado à paixão, urgência ou perigo, enquanto o azul pode transmitir confiança e profissionalismo (Bleicher, 2023).

A escolha adequada das cores pode melhorar a legibilidade, atração e até mesmo a taxa de conversão de um site, mantendo os usuários mais focados no que realmente o site quer destacar, promovendo conforto na navegação, de maneira que seja muito mais agradável ao usuário a interação com o site, garantindo que ele permaneça ali por mais tempo (Beaird, 2020).

2.2.2 **Experiência do Usuário (UX) e Interface do Usuário (UI)**

A Experiência do Usuário (UX) refere-se à percepção e resposta emocional de um usuário ao usar um produto ou serviço, neste caso, um website ou aplicação web. Envolve todos os aspectos da interação do usuário com o produto, desde a facilidade de uso até a satisfação geral. Por outro lado, a Interface do Usuário (UI) é o ponto de interação entre o usuário e o sistema digital, abrangendo os elementos visuais, como botões, ícones e menus (Sharma e Tiwari, 2021).

A relação entre UX e UI é intrínseca. Uma boa UI pode facilitar uma excelente UX, mas uma UI visualmente agradável sem consideração pela experiência do usuário pode resultar em frustração e abandono. No desenvolvimento web, é essencial que designers e desenvolvedores colaborem estreitamente para garantir que a estética (UI) e a funcionalidade (UX) trabalhem em harmonia para atender às necessidades e expectativas do usuário (Sharma e Tiwari, 2021).

2.3 FERRAMENTAS E TECNOLOGIAS

No cenário atual de desenvolvimento web, a escolha de ferramentas e tecnologias adequadas é fundamental para garantir a eficiência, escalabilidade e adaptabilidade de um projeto. Estas escolhas não apenas influenciam a qualidade do produto, mas também o fluxo de trabalho, a colaboração entre equipes e a manutenção a longo prazo. Neste tópico, serão abordadas as ferramentas e tecnologias selecionadas para o desenvolvimento do catálogo de filmes web, destacando suas características, benefícios e relevância no contexto do projeto.

2.3.1 Grid de 8 pontos

O grid de 8 pontos é uma convenção que utiliza múltiplos de 8 para escolher o espaçamento, tamanhos de elementos e fontes, mantendo um visual agradável e padronizado para o sistema desenvolvido, proporcionando uma melhor experiência do usuário (Kenzie, 2023).

A princípio, é inserido um grid no protótipo, que pode ser uma grade, linhas ou colunas, seguindo o padrão de múltiplos de 8. Depois há duas vertentes, o hard grid, em que seu alinhamento segue fielmente o grid colocado, mesmo que os espaçamentos fiquem com números muito aleatórios, e o soft grid, que segue apenas as colunas ultrapassando um pouco, quando necessário, tornando o processo de design mais rápido e o de desenvolvimento menos complexo (Kenzie, 2023).

2.3.2 Figma

O Figma (2023) emergiu como uma ferramenta de design e prototipação líder no mercado, especialmente no contexto do desenvolvimento web. Sua interface intuitiva, capacidade de colaboração em tempo real e vasta gama de funcionalidades o tornam uma escolha ideal para designers e desenvolvedores. Além de suas

funcionalidades básicas de design, o Figma destaca-se pela capacidade de gerar CSS otimizado, facilitando a transição da fase de design para a implementação.

No Figma (2023), é possível inserir o estilo do documento como cores, tipografia e efeito, que garantem que cada componente esteja dentro da identidade visual da empresa para a qual o site está sendo desenvolvido. Outra funcionalidade muito importante é que ao alterar um estilo, todos os componentes do design que utilizam aquele estilo serão alterados juntamente.

Uma boa prática no Figma (2023) é optar pela utilização de frames, em detrimento de formas, pois eles facilitam no dimensionamento dos itens, sem gerar distorções e no alinhamento dos itens que o compõem. O tamanho dos frames pode ser ajustado para abraçar o conteúdo ou para se ajustar dentro de outros frames, facilitando sua utilização e gerando um código CSS mais responsivo.

Os frames também possuem a função de auto layout que garante o alinhamento e padronização dos espaçamentos dos itens presentes no frame, facilitando a utilização do mesmo componente em diferentes telas e agilidade na alteração de espaçamentos, caso necessário. Outra facilidade é que ele possui um comportamento semelhante ao *flexbox* do CSS, gerando um código CSS muito mais usual, pois a utilização do *flexbox* no estilo de um site é essencial para promover uma boa responsividade (Figma, 2023).

Outra funcionalidade importante é a criação de componentes por meio de elementos utilizados no Figma (2023). Quando classificado como componente, aquele conjunto de elementos, juntamente com seus estados e interações fica armazenado para que seja replicado no design. Sempre que um componente pai é alterado, todos os seus componentes filhos sofrem a alteração também, facilitando e agilizando alterações de design.

2.3.3 HTML5, CSS3 e JavaScript

No cenário contemporâneo da web, HTML5, CSS3 e JavaScript emergem como pilares essenciais para a construção bem-sucedida de projetos online inovadores e funcionais. Cada uma dessas tecnologias desempenha um papel crucial, contribuindo para a criação de experiências web ricas e envolventes (Collins, 2017).

O HTML5, com sua gama de novas *tags* semânticas e recursos avançados, proporciona a capacidade de desenvolver estruturas web mais claras e acessíveis. Além de ser uma linguagem de marcação, o HTML serve como um guia interpretativo

para os navegadores, facilitando a compreensão do significado de cada parte do código. Sua presença estratégica em algoritmos de busca otimiza a visibilidade do conteúdo, melhorando a eficiência dos motores de busca (MDN, 2023a).

Por sua vez, o CSS3 assume a responsabilidade pela estilização da página, oferecendo um conjunto expandido de propriedades e valores que permitem uma estética visualmente atraente. Ao empregar o CSS3, os desenvolvedores têm a capacidade de personalizar o design e o posicionamento dos elementos definidos no HTML, garantindo uma apresentação visual coesa e agradável (MDN, 2023b).

O JavaScript, como uma linguagem de programação dinâmica, eleva a interatividade a um novo patamar. Sua função é ir além do estático, permitindo a criação de funcionalidades avançadas que transformam sites em aplicativos interativos e dinâmicos. Além disso, o JavaScript desempenha um papel crucial na integração entre o *front-end* e o *back-end*, colaborando com frameworks e bibliotecas para proporcionar uma experiência de usuário fluida e consistente (MDN, 2023c).

Em resumo, a tríade HTML5, CSS3 e JavaScript não apenas estabelece a base tecnológica para o desenvolvimento web, mas também colabora harmoniosamente para criar uma sinergia entre estrutura, estilo e interatividade. Essas tecnologias são essenciais para a evolução contínua da web, capacitando desenvolvedores a conceberem experiências online cada vez mais sofisticadas e envolventes.

2.3.4 Vue3, Vuex e Vue Router

O Vue.js, um framework JavaScript progressivo, tem se destacado na construção de interfaces de usuário e aplicações de página única, e a versão 3, conhecida como Vue3, elevou ainda mais o padrão ao introduzir melhorias notáveis em termos de desempenho e escalabilidade. No ecossistema Vue, dois elementos-chave desempenham papéis fundamentais no desenvolvimento de aplicações web avançadas: Vuex e Vue Router (Vue.js, 2023a; Vuex, 2023; Vue Router, 2023).

O Vuex (2023), uma biblioteca de gerenciamento de estado dedicada ao Vue.js, oferece uma abordagem estruturada para a administração de dados. A centralização dos dados na "store" simplifica a garantia da integridade deles, facilitando o acesso uniforme a todos os componentes da aplicação. A "store" consiste em "*states*" (dados armazenados), "*actions*" (funções que permitem a alteração ou solicitação de dados da "store" por outros componentes) e "*mutations*" (funções intermediárias entre "*actions*" e "*states*").

Por sua vez, o Vue Router, como solução oficial para roteamento em aplicações Vue, desempenha um papel crucial na configuração e organização das rotas entre os diversos componentes. Ele fornece uma maneira eficiente de navegar entre as diferentes páginas de uma aplicação, garantindo uma transição suave e coerente (Vue Router, 2023; Vue.js, 2023c).

A utilidade do Vue transcende o simples desenvolvimento de páginas web, adentrando o domínio da componentização no projeto. Através do Vue, cada página é estruturada de maneira modular, com seções distintas para o HTML ("`<template></template>`"), JavaScript ("`<script></script>`"), e CSS ("`<style></style>`"). Essa abordagem facilita a manutenção, escalabilidade e colaboração entre desenvolvedores, promovendo uma arquitetura limpa e modular.

2.3.5 Options API

Options API é uma API desenvolvida pelo Vue 3. Este API é essencial para a definição de componentes em aplicações Vue, proporcionando uma estrutura clara e declarativa para organizar o código. Com o Options API, os desenvolvedores podem estruturar componentes de maneira mais intuitiva, melhorando a legibilidade e a manutenção do código (Vue.js, 2023b; Vue.js, 2023c).

Uma das características notáveis do Options API é a clareza na definição de dados, métodos e *hooks* do ciclo de vida do componente. Os dados são organizados em uma função `data()`, tornando mais explícito como o estado do componente é inicializado. Isso facilita a compreensão do fluxo de dados e o seu armazenamento (Vue.js, 2023b).

Além disso, o Options API facilita a composição de comportamentos reutilizáveis por meio da função `setup()`. Nelas os desenvolvedores podem extrair lógica de componentes para funções reutilizáveis e ainda manter uma relação clara entre o estado do componente e suas funcionalidades (Vue.js, 2023b; Vue.js, 2023c).

Outra ferramenta notável é a organização dos *hooks* do ciclo de vida em funções separadas, tornando o código mais modular e fácil de entender. Os *hooks* do ciclo de vida são agrupados logicamente, proporcionando uma visão mais clara do fluxo de vida de um componente Vue (Vue.js, 2023b).

Em resumo, o Options API do Vue 3 representa um avanço significativo na construção de componentes Vue mais legíveis, modulares e reutilizáveis. Essas

melhorias não apenas simplificam o processo de desenvolvimento, mas também contribuem para a manutenção eficiente de aplicações Vue complexas.

2.3.6 Node JS

Node.js é um ambiente de execução JavaScript no lado do servidor, projetado para criar aplicativos web escaláveis. Sua arquitetura assíncrona e orientada a eventos permite que operações intensivas de entrada/saída, como chamadas de rede e acesso a banco de dados, ocorram simultaneamente, melhorando a eficiência. Com o padrão CommonJS para modularização, os desenvolvedores podem organizar seus códigos em módulos reutilizáveis, promovendo a manutenibilidade (Hughes-Croucher e Wilson, 2012; Node.js, 2023).

O Node.js inclui o npm, um robusto gerenciador de pacotes que simplifica o processo de instalação e gerenciamento de dependências. Esse ecossistema dinâmico é composto por uma ampla gama de módulos e frameworks, como o popular Express.js para construção de aplicativos web e o Socket.io para facilitar o desenvolvimento de aplicações em tempo real. Além disso, a capacidade do Node.js de lidar eficientemente com muitas conexões simultâneas contribuem para sua escalabilidade, tornando-o uma escolha atraente para projetos de larga escala (Node.js, 2023).

2.3.7 API do The Movie DB

A API do The Movie Database (TMDb) é uma poderosa ferramenta que oferece acesso a uma vasta quantidade de informações relacionadas a filmes e programas de televisão. Desenvolvedores podem integrar essa API em suas aplicações para obter dados detalhados sobre filmes, programas de TV, elenco, equipes de produção, avaliações e muito mais (TMDb-API, 2023).

Com *endpoints* bem documentados, a API do TMDb permite que os desenvolvedores personalizem suas consultas para atender às necessidades específicas de suas aplicações. Isso inclui a capacidade de buscar informações sobre filmes em cartaz, próximos lançamentos, detalhes específicos de um filme ou série, e até mesmo obter imagens e trailers (TMDb-API, 2023).

Essa API é valiosa para a criação de aplicativos relacionados ao mundo do entretenimento, como plataformas de recomendação, catálogos de filmes, ou até mesmo para fornecer informações detalhadas sobre produções cinematográficas em

um site ou aplicativo. Além disso, a API do TMDb é gratuita para uso básico, com opções de autenticação para acesso mais avançado, tornando-a acessível e atraente para uma ampla gama de desenvolvedores (TMDb-API, 2023).

2.3.8 My SQL

SQL, ou *Structured Query Language*, é uma linguagem de programação projetada para gerenciar e manipular bancos de dados relacionais. Amplamente utilizada em sistemas de gerenciamento de banco de dados (SGBDs) como MySQL, PostgreSQL e Microsoft SQL Server, o SQL oferece uma maneira padronizada de realizar operações como consulta, inserção, atualização e exclusão de dados. Sua estrutura declarativa permite aos desenvolvedores especificar o que desejam alcançar, em vez de precisar definir o processo exato para alcançar um resultado (Date, 1989; MySQL, 2023).

A linguagem SQL é dividida em categorias principais, incluindo o DDL (*Data Definition Language*), usado para definir e modificar a estrutura do banco de dados, e o DML (*Data Manipulation Language*), que lida com a manipulação dos dados contidos no banco. A capacidade de realizar consultas complexas, realizar junções de tabelas e garantir a integridade referencial são aspectos-chave que tornam o SQL essencial para a administração eficaz de dados em ambientes relacionais (MySQL, 2023).

Além disso, o SQL é altamente escalável e compatível com uma ampla gama de aplicações, desde sistemas de gerenciamento de conteúdo até aplicações empresariais complexas. Sua natureza padronizada e ampla adoção na indústria o tornam uma habilidade valiosa para profissionais de desenvolvimento e administração de bancos de dados (MySQL, 2023).

2.3.9 AXIOS

O Axios é uma biblioteca JavaScript amplamente utilizada para realizar requisições HTTP, tanto no navegador quanto no ambiente Node.js. Comumente empregado em projetos Vue.js e React, o Axios simplifica o processo de comunicação entre o *frontend* e o *backend*. Sua popularidade advém da sua sintaxe simples e fácil integração, tornando-o uma escolha comum para desenvolvedores que buscam uma solução robusta e eficiente para gerenciamento de requisições HTTP (Axios, 2023).

Uma das características marcantes do Axios é a sua promessa baseada em JavaScript, que permite lidar com requisições de forma assíncrona de maneira mais

elegante e legível. Além disso, o Axios oferece um conjunto abrangente de funcionalidades, incluindo suporte para interceptores de requisição e resposta, facilitando a manipulação global de requisições, como adicionar cabeçalhos ou tratar erros de forma centralizada (Axios, 2023).

Outro ponto positivo do Axios é sua compatibilidade com as principais tecnologias, tornando-o versátil para diferentes tipos de projetos. Ele suporta tanto a API de *Promises* quanto a API de *async/await*, proporcionando flexibilidade na escolha do estilo de programação. Além disso, a capacidade de realizar automaticamente a conversão de dados para JSON simplifica o processo de intercâmbio de informações entre o *frontend* e o *backend* (Axios, 2023).

2.3.10 Netlify

O Netlify é uma plataforma de desenvolvimento moderna que oferece uma abordagem abrangente e simplificada para a construção, implantação e gerenciamento de aplicativos da web. Baseada em uma arquitetura de nuvem, a plataforma Netlify oferece recursos como hospedagem de sites, integração contínua, entrega contínua (CI/CD) e funcionalidades de escala automática.

A proposta fundamental do Netlify é proporcionar aos desenvolvedores uma experiência de desenvolvimento sem atritos, onde a complexidade da infraestrutura é ocultada, permitindo que equipes concentrem seus esforços na criação de funcionalidades inovadoras. A integração contínua é facilitada por meio de gatilhos automatizados que disparam pipelines de construção e implantação sempre que há alterações no repositório de código. Além disso, o Netlify suporta ambientes de pré-visualização, possibilitando que equipes revejam as alterações antes da implantação em produção.

A escalabilidade é tratada de forma transparente, com recursos como a distribuição global de conteúdo por meio de uma rede de entrega de conteúdo (CDN), garantindo um desempenho otimizado em diferentes regiões do mundo. Com suporte a linguagens de programação populares e uma variedade de ferramentas integradas, o Netlify se destaca como uma escolha valiosa para desenvolvedores que buscam eficiência e simplicidade em seus fluxos de trabalho.

2.3.11 PlanetScale

O PlanetScale é uma plataforma de banco de dados distribuído que se destaca pela sua abordagem inovadora na gestão de dados em escala global. Projetado para atender às demandas de aplicativos modernos, o PlanetScale baseia-se no MySQL e oferece uma solução altamente disponível e escalável que elimina os desafios associados à replicação de dados em diferentes localidades geográficas.

A arquitetura do PlanetScale permite que as organizações gerenciem dados de maneira consistente, independentemente da localização dos usuários finais. Isso é alcançado por meio de uma combinação de replicação automática, particionamento inteligente e gerenciamento eficiente de esquemas. Além disso, o PlanetScale oferece uma interface familiar para desenvolvedores MySQL, facilitando a transição e a adoção por equipes já proficientes nesse sistema de gerenciamento de banco de dados.

Ao adotar o PlanetScale, as organizações podem beneficiar-se da confiabilidade, escalabilidade e flexibilidade necessárias para enfrentar os desafios de operar aplicativos em escala global. A plataforma contribui significativamente para a consistência e a disponibilidade de dados, permitindo que as empresas atendam às expectativas dos usuários em um mundo cada vez mais interconectado.

3 MODELAGEM DO SISTEMA

3.1 DESCRIÇÃO GERAL DO SISTEMA

O sistema é construído em torno da API TMDb para obter detalhes sobre filmes e séries, utilizando o Axios no Vue.js para requisições e manipulação reativa dos dados. O *backend*, alimentado por Node.js (2023) e MySQL (2023), incorpora tabelas para usuários, favoritos e assistidos, facilitando operações como DELETE e SELECT. A persistência local é assegurada pela store do Vue.js (2023a), exigindo login para associar dados e alertando sobre armazenamento não autenticado.

A navegação entre seções como home, filmes e séries é gerenciada suavemente pelo Vue Router (2023) e pelo componente header. As páginas contêm uma estrutura consistente oferecendo opções de ordenação e filtragem. Ao clicar em

qualquer título, o usuário é direcionado para a tela de detalhes, onde pode ver mais detalhes do filme.

A aplicação também oferece a possibilidade de o usuário marcar os filmes e séries como favoritos e assistidos, esses dados ficam armazenados associados ao login do usuário. Componentes Vue.js (2023c) são implementados para modularizar o sistema, promovendo reutilização eficiente de código e facilitando a manutenção contínua do sistema.

3.2 REQUISITOS

Na seção a seguir são demonstrados os requisitos funcionais e não funcionais, que descrevem a necessidade do usuário e do sistema.

Quadro 1 - RF 001: Fazer Login

Identificador	Nome
RF 001	Fazer autenticação do usuário
Prioridade	Autor
Essencial	Tayná Rodrigues de Sousa Carvalho
Descrição	
Este requisito funcional viabiliza o acesso dos usuários às suas contas no sistema. Para efetuar o login, é mandatório que o usuário forneça um endereço de e-mail e uma senha, ambos válidos. Uma vez verificadas e autenticadas as credenciais, o usuário obtém acesso integral à plataforma e a todas as funcionalidades pertinentes. Em situações em que as credenciais inseridas sejam incorretas, o sistema prontamente apresentará uma mensagem clara de erro, permitindo ao usuário a oportunidade de tentar o processo de login novamente.	

Fonte: Elaborado pelo Autor

Quadro 2 - RF 002: Cadastrar

Identificador

Nome

RF 002

Cadastrar cliente

Prioridade

Autor

Essencial

Tayná Rodrigues de Sousa Carvalho

Descrição

Este requisito funcional descreve o processo pelo qual um novo usuário se registra no sistema. O usuário deve fornecer informações pessoais, como nome, e-mail e criar uma senha. O sistema deve verificar a unicidade do e-mail e confirmar a senha para completar o cadastro. Uma vez cadastrado, o usuário pode acessar o sistema usando as credenciais definidas.

Fonte: Elaborado pelo Autor

Quadro 3 - RF 003: Explorar Conteúdo

Identificador

Nome

RF 003

Explorar filmes e séries por categorias

Prioridade

Autor

Essencial

Tayná Rodrigues de Sousa Carvalho

Descrição

Este requisito engloba a funcionalidade avançada de navegação e descoberta de filmes e séries na plataforma. O usuário pode explorar diversas categorias de conteúdo, incluindo filmes e séries em geral, seus favoritos, aqueles já assistidos, e os que estão em alta popularidade. Cada categoria é apresentada em listagens distintas e organizadas, facilitando a busca e o acesso. Ao selecionar um título de interesse, o usuário é conduzido a uma página detalhada, que oferece informações abrangentes sobre o filme ou série.

Fonte: Elaborado pelo Autor

Quadro 4 - RF 004: Ver Listagem de Conteúdo

Identificador	Nome
RF 004	Listagem de filmes e séries
Prioridade	Autor
Essencial	Tayná Rodrigues de Sousa Carvalho
Descrição	
<p>Este requisito funcional detalha como os usuários acessam e interagem com o conteúdo disponível na plataforma. Independentemente da página em que se encontram, os usuários terão à disposição uma ampla variedade de filmes e séries, organizados em listas na interface. Essa estrutura facilita a navegação e a descoberta de novos conteúdos. Além disso, a plataforma oferece recursos de filtragem e ordenação, permitindo aos usuários personalizar a visualização dos conteúdos conforme suas preferências.</p>	

Fonte: Elaborado pelo Autor

Quadro 5 - RF 005: Ver Detalhes

Identificador	Nome
RF 005	Ver Detalhes do filme ou da serie
Prioridade	Autor
Essencial	Tayná Rodrigues de Sousa Carvalho
Descrição	
<p>Este requisito funcional permite que os usuários vejam informações detalhadas sobre um filme ou série selecionado, incluindo sinopse, elenco, trailer, ano de lançamento, duração, avaliação, nacionalidade, gêneros e outros. Ao acessar os</p>	

detalhes de um filme ou série, o usuário pode também marcar o conteúdo como favorito ou assistido.

Fonte: Elaborado pelo Autor

Quadro 6 - RF 006: Buscar

Identificador**Nome**

RF 006

Buscar por palavra chave

Prioridade**Autor**

Essencial

Tayná Rodrigues de Sousa Carvalho

Descrição

Este requisito funcional permite que os usuários pesquisem filmes e séries no sistema usando palavras-chave. O sistema deve responder com uma listagem de conteúdo que correspondem aos termos de pesquisa em títulos, descrições ou elenco. A busca pode ser refinada com a ajuda de filtros e ordenação para melhor atender às necessidades do usuário.

Fonte: Elaborado pelo Autor

Quadro 7 - RF 007: Filtrar

Identificador**Nome**

RF 007

Filtrar filmes e series

Prioridade**Autor**

Essencial

Tayná Rodrigues de Sousa Carvalho

Descrição

Este requisito funcional descreve a funcionalidade que permite aos usuários aplicarem filtros específicos, como gênero, para restringir a listagem de conteúdo

disponível. O sistema deve atualizar a listagem de conteúdo em tempo real conforme os filtros são aplicados ou removidos.

Fonte: Elaborado pelo Autor

Quadro 8 - RF 008: Ordenar

Identificador**Nome**

RF 008

Ordenar filmes e series

Prioridade**Autor**

Essencial

Tayná Rodrigues de Sousa Carvalho

Descrição

Este requisito funcional permite que os usuários alterem a ordem de exibição do conteúdo listado, seja por popularidade, avaliação dos usuários, data de lançamento ou alfabeticamente. A interface deve refletir a nova ordem imediatamente após a seleção do critério de ordenação pelo usuário.

Fonte: Elaborado pelo Autor

Quadro 9 - RF 009: Marcar como favorito

Identificador**Nome**

RF 009

Marcar como favorito

Prioridade**Autor**

Essencial

Tayná Rodrigues de Sousa Carvalho

Descrição

Este requisito funcional permite aos usuários marcarem filmes e séries como favoritos. Quando um usuário marca um item como favorito, o sistema deve garantir que ele seja facilmente acessível através de uma seção de favoritos. Isso facilita o

acesso rápido a programas e filmes que o usuário deseja assistir novamente ou manter em destaque.

Fonte: Elaborado pelo Autor

Quadro 10 - RF 010: Marcar como assistido

Identificador

Nome

RF 010

Marcar filme ou serie como assistido

Prioridade

Autor

Essencial

Tayná Rodrigues de Sousa Carvalho

Descrição

Este requisito funcional possibilita que os usuários indiquem que já assistiram a um determinado conteúdo. Ao marcar um item como assistido, o sistema deve registrar esta informação no perfil do usuário, permitindo ao usuário manter um histórico dos conteúdos vistos.

Fonte: Elaborado pelo Autor

Quadro 11 - RN 001: Requisições à API otimizada

Identificador

Nome

RNF 001

Requisições à API de 3 segundos

Prioridade

Autor

Essencial

Tayná Rodrigues de Sousa Carvalho

Descrição

Nesse requisito funcional, o tempo de resposta para as requisições à API TMDb deve ser de no máximo 3 segundos.

Fonte: Elaborado pelo Autor

Quadro 12 - RN 002: Conexão com banco de dados eficiente

Identificador

Nome

RNF 002

Conexão com banco na primeira ou segunda tentativa

Prioridade

Autor

Essencial

Tayná Rodrigues de Sousa Carvalho

Descrição

Nesse requisito funcional, a conexão com o banco de dados MySQL deve ser eficiente, minimizando o tempo de execução das operações CRUD, de forma que não sejam necessárias mais de 2 tentativas.

Fonte: Elaborado pelo Autor

Quadro 13 - RN 003: Criptografia de informações sensíveis

Identificador

Nome

RNF 003

Criptografia de informações sensíveis

Prioridade

Autor

Essencial

Tayná Rodrigues de Sousa Carvalho

Descrição

Esse requisito garante que as informações sensíveis dos usuários armazenadas no banco de dados sejam devidamente criptografadas para garantir a confidencialidade

Fonte: Elaborado pelo Autor

Quadro 14 - RN 004: Fácil expansão de funcionalidades futuras

Identificador

Nome

RNF 004	Fácil expansão de funcionalidades futuras por meio da modularização
----------------	---

Prioridade

Autor

Essencial	Tayná Rodrigues de Sousa Carvalho
-----------	-----------------------------------

Descrição

Esse requisito permite fácil expansão de funcionalidades futuras, como a inclusão de novos componentes, por meio da arquitetura modular e escalável.

Fonte: Elaborado pelo Autor

Quadro 15 - RN 005: Integridade e consistência dos dados em casos de recarregamento da página

Identificador

Nome

RNF 005	Integridade e consistência dos dados em casos de recarregamento da página
----------------	---

Prioridade

Autor

Essencial	Tayná Rodrigues de Sousa Carvalho
-----------	-----------------------------------

Descrição

Esse requisito garante a integridade e consistência dos dados entre sessões, mesmo em casos de recarregamento da página, por meio da persistência local de dados usando vuex.

Fonte: Elaborado pelo Autor

Quadro 16 - RN 006: Sincronização eficiente entre a persistência local e o banco de dados

Identificador	Nome
RNF 006	Sincronização eficiente entre a persistência local e o banco de dados
Prioridade	Autor
Essencial	Tayná Rodrigues de Sousa Carvalho
Descrição	
Neste requisito, a sincronização eficiente entre a persistência local e o banco de dados deve ser assegurada para evitar inconsistências nos dados do usuário.	
Fonte: Elaborado pelo Autor	

Quadro 17 - RN 007: Interface do usuário deve ser intuitiva

Identificador	Nome
RNF 007	Interface do usuário deve ser intuitiva
Prioridade	Autor
Essencial	Tayná Rodrigues de Sousa Carvalho
Descrição	
Esse requisito infere que a interface do usuário deve ser intuitiva, garantindo uma experiência de usuário agradável.	
Fonte: Elaborado pelo Autor	

Quadro 18 - RN 008: Comunicação de ações disponíveis para usuários não autenticados

Identificador

Nome

RNF 008

Comunicação de ações disponíveis para usuários não autenticados

Prioridade

Autor

Essencial

Tayná Rodrigues de Sousa Carvalho

Descrição

Esse requisito permite que as ações disponíveis para usuários não autenticados sejam claramente comunicadas, incentivando a autenticação para uma experiência mais completa.

Fonte: Elaborado pelo Autor

Quadro 19 - RN 009: Compatibilidade com os principais navegadores web e dispositivos móveis

Identificador

Nome

RNF 009

Compatibilidade com os principais navegadores web e dispositivos móveis

Prioridade

Autor

Essencial

Tayná Rodrigues de Sousa Carvalho

Descrição

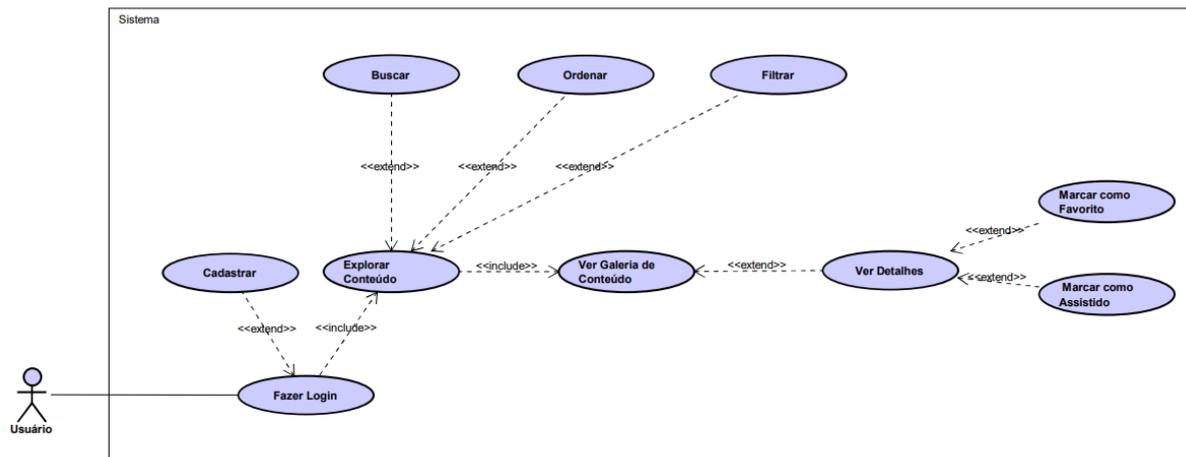
Esse requisito infere que o sistema deve ser compatível com os principais navegadores web e dispositivos móveis para atender a uma ampla variedade de usuários.

Fonte: Elaborado pelo Autor

3.3 DIGRAMA DE CASO DE USO

Na Figura 1, é demonstrado por meio do diagrama a interação entre os casos de uso, e do usuário com o sistema.

Figura 1 - Diagrama de caso de uso

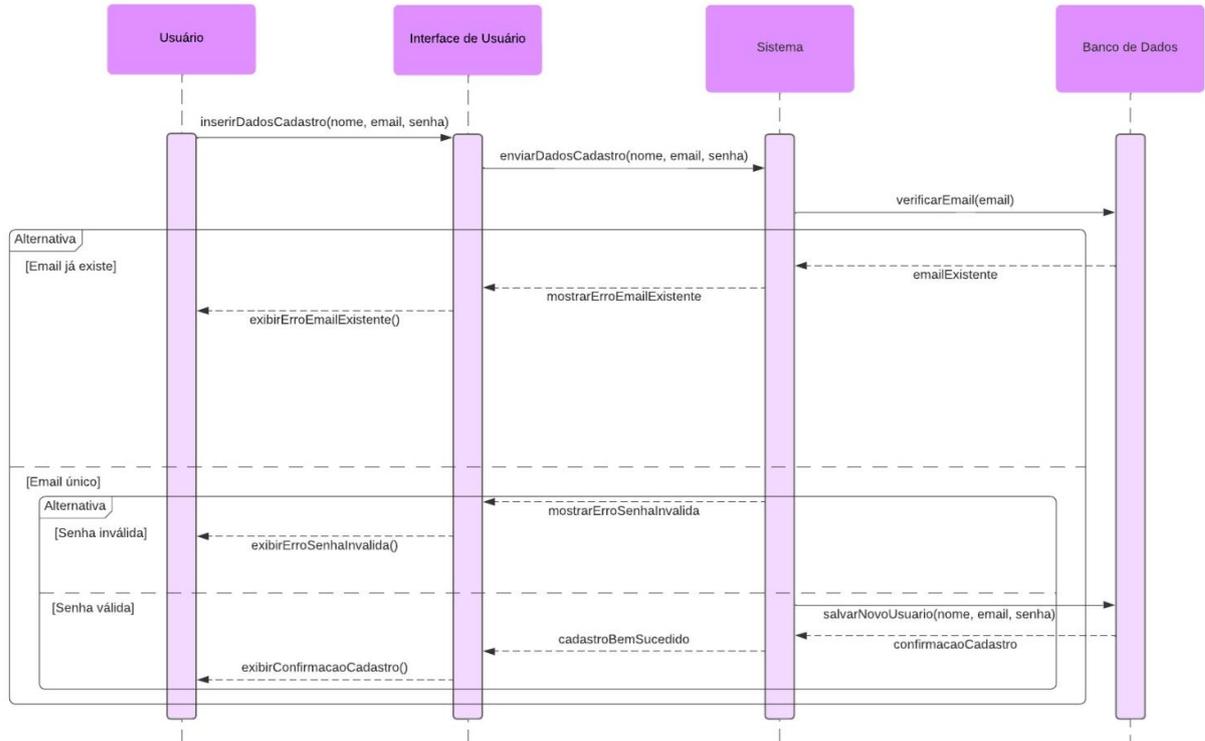


Fonte: Elaborado pelo Autor

3.4 DIAGRAMAS DE SEQUÊNCIA

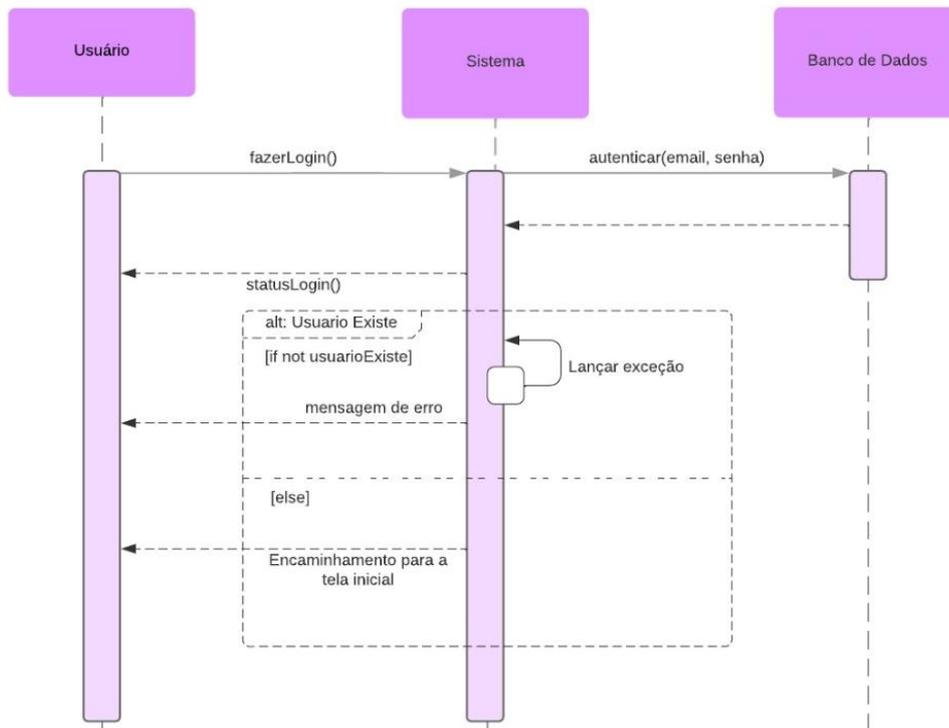
Nessa seção serão apresentados os diagramas de sequência de cada caso de uso, gerados por meio dos requisitos funcionais.

Figura 2 - CSU 001: Cadastrar



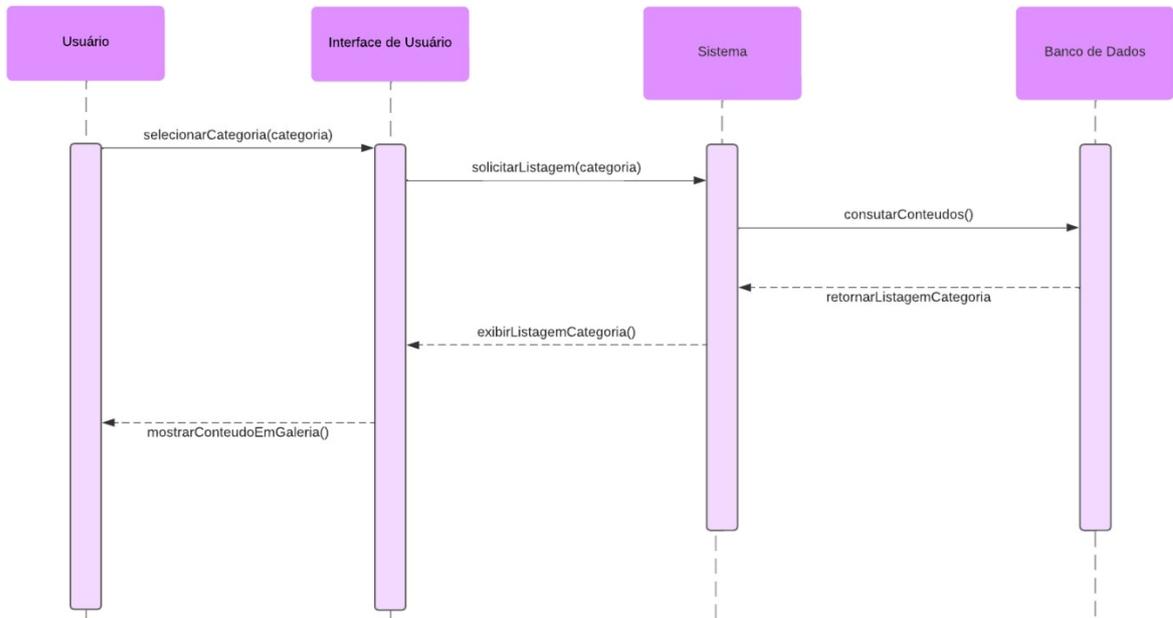
Fonte: Elaborado pelo Autor

Figura 3 - CSU 002: Fazer Login



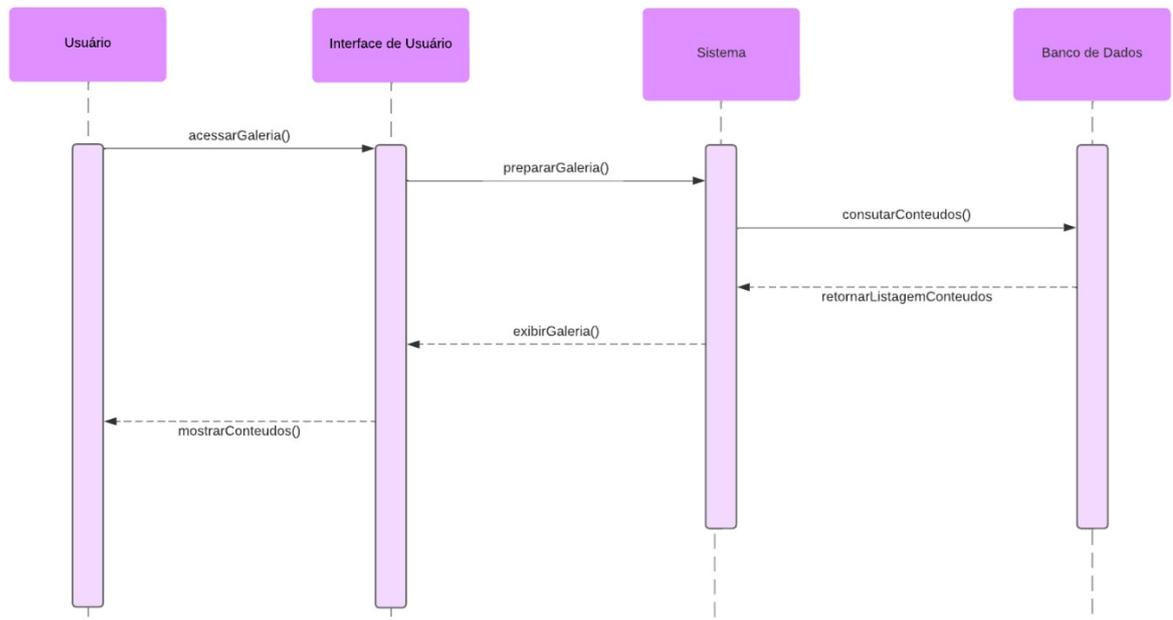
Fonte: Elaborado pelo Autor

Figura 4 - CSU 003: Explorar Conteúdo



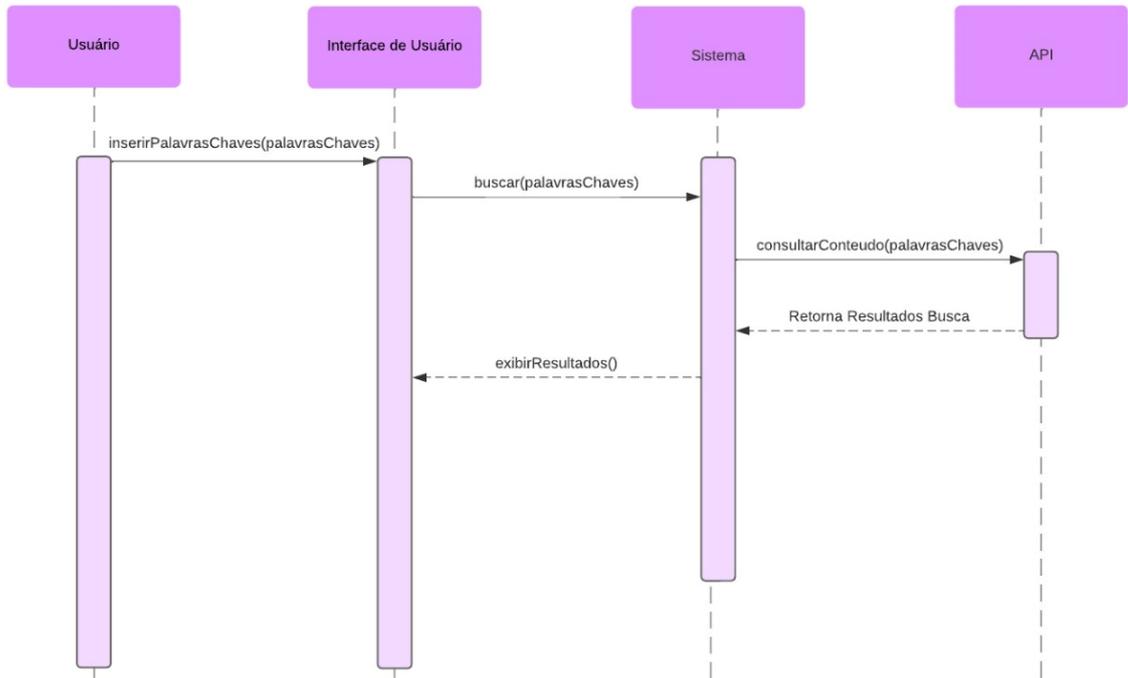
Fonte: Elaborado pelo Autor

Figura 5 - CSU 004: Ver Galeria de Conteúdo



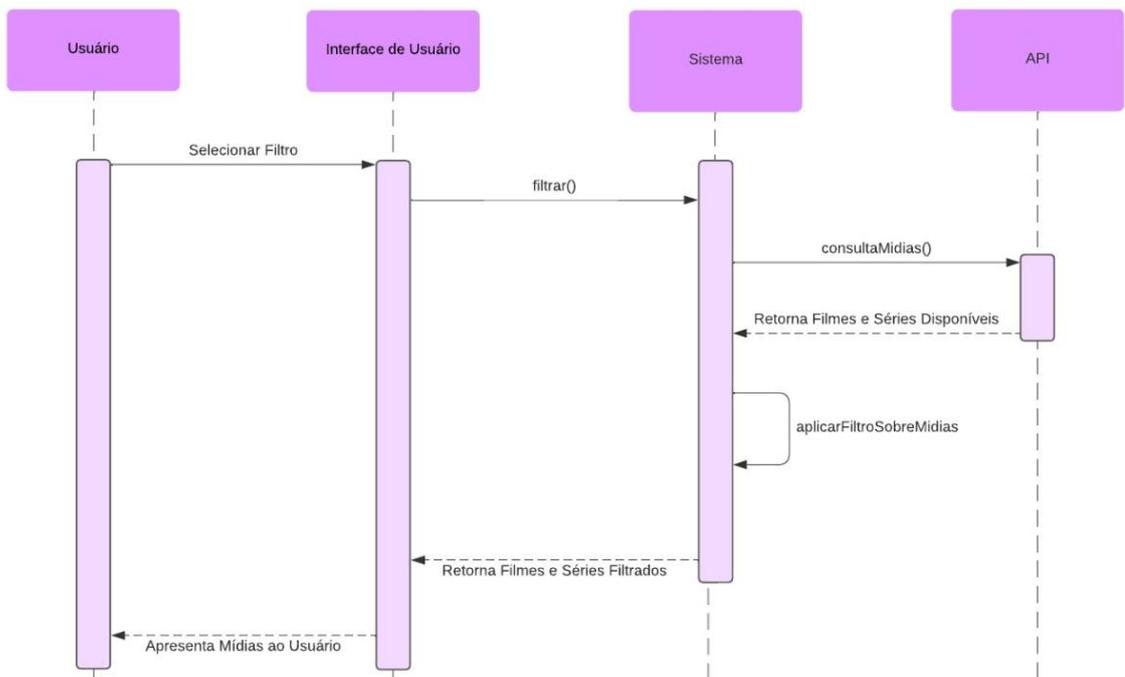
Fonte: Elaborado pelo Autor

Figura 6 - CSU 005: Buscar



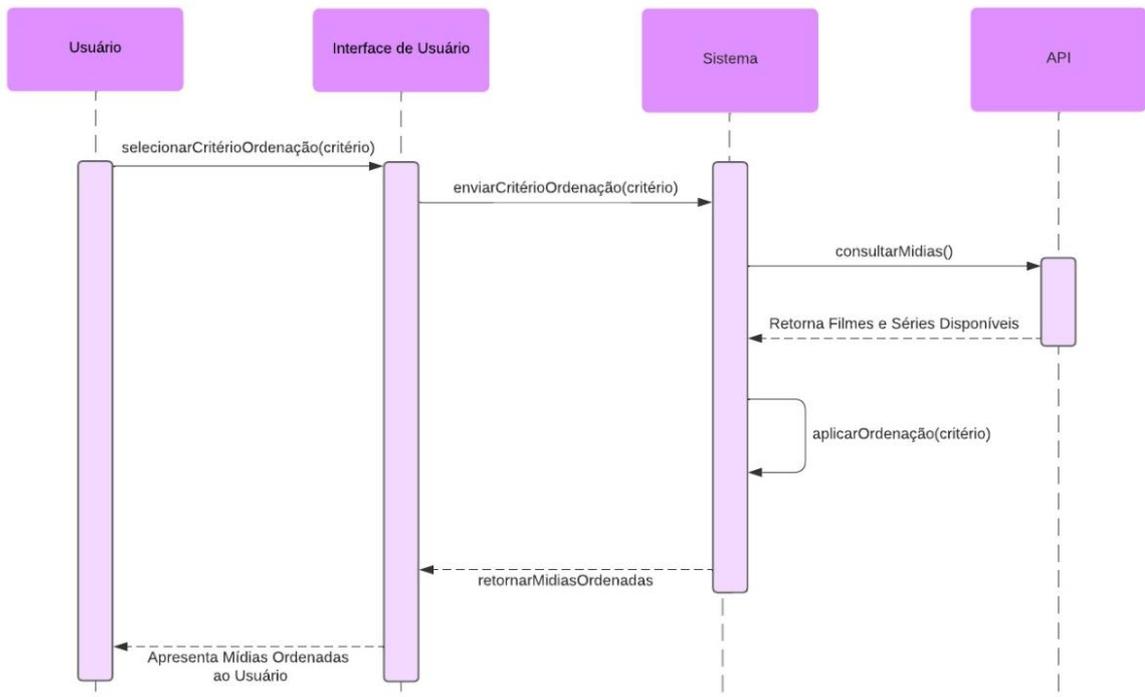
Fonte: Elaborado pelo Autor

Figura 7 - CSU 006: Filtrar



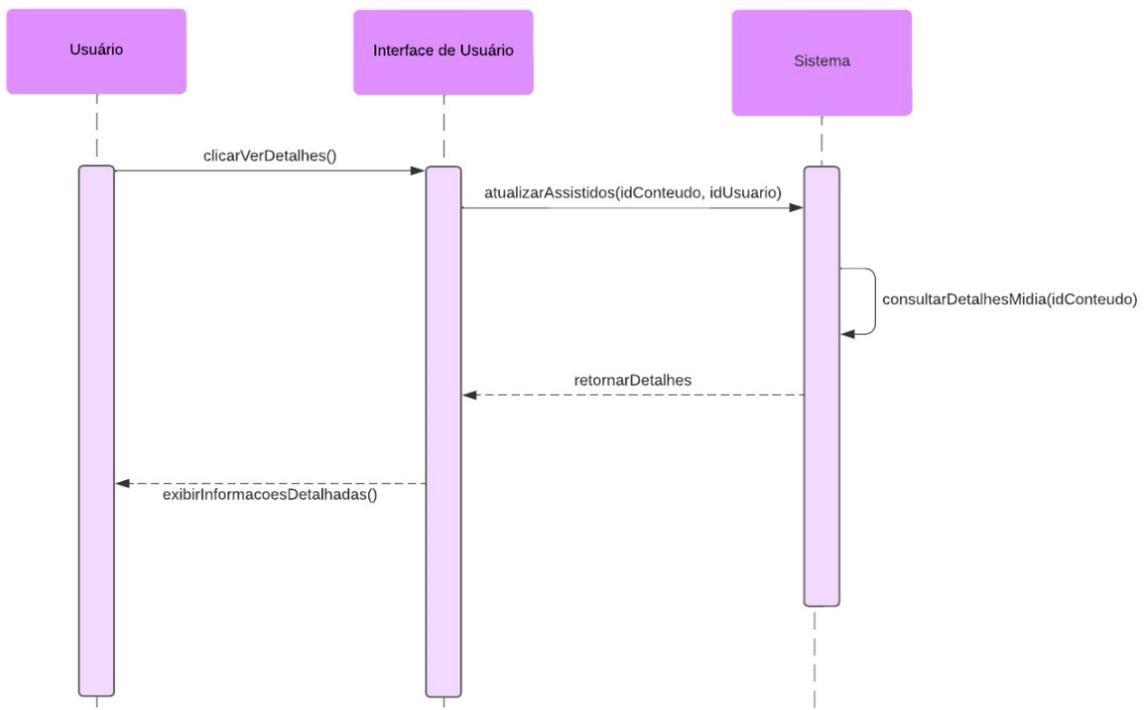
Fonte: Elaborado pelo Autor

Figura 8 - CSU 007: Ordenar



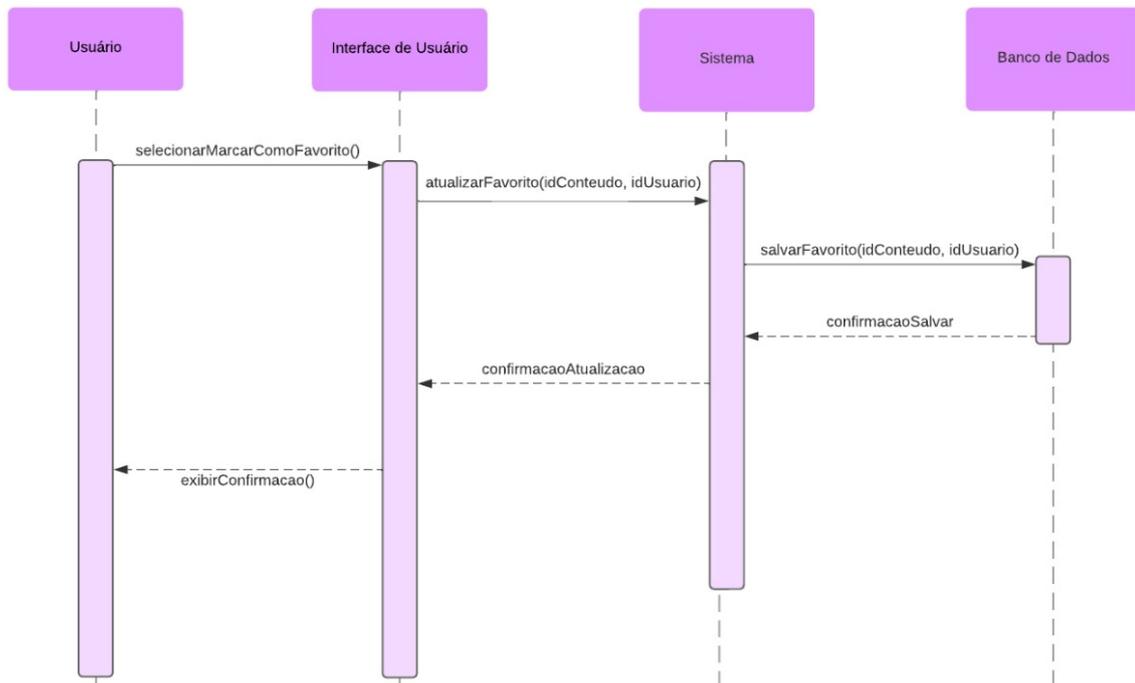
Fonte: Elaborado pelo Autor

Figura 9 - CSU 008: Ver Detalhes



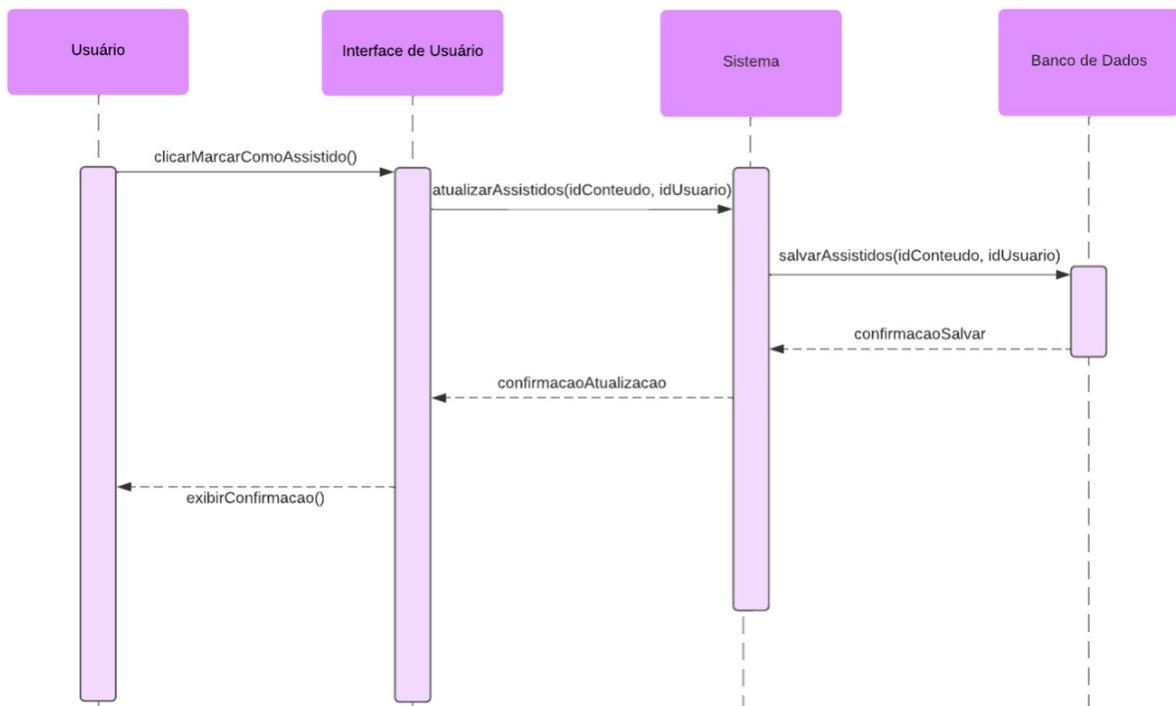
Fonte: Elaborado pelo Autor

Figura 10 - CSU 009: Marcar como favorito



Fonte: Elaborado pelo Autor

Figura 11 - CSU 010: Marcar como assistido



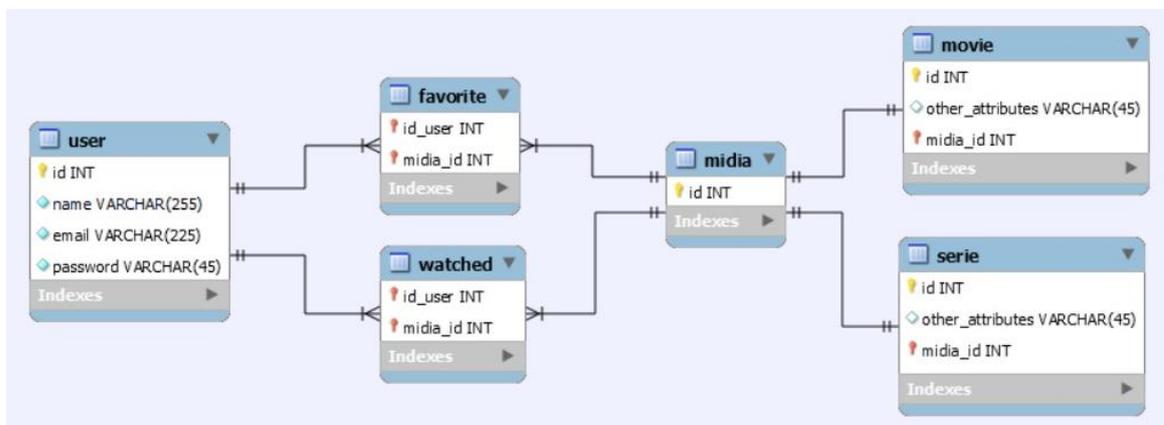
Fonte: Elaborado pelo Autor

3.5 BANCO DE DADOS

3.5.1 MER

Na Figura 12 é apresentado um MER (Modelo Entidade Relacionamento) figurativo pois ilustra as tabelas *mídia*, *movie* e *serie* que não são inclusas no banco de dados, e foram utilizadas com o intuito de representar tabelas da API utilizada TMDb. É valido ressaltar que *favorite* e *watched*, são camadas intermediárias, e por isso utilizam chave privada composta, que é uma chave composta por mais de uma chave. A tabela *mídia* também é intermediária e foi colocada com o intuito de ilustrar a ligação entre *favorite* ou *watched* com *movie* ou *serie*, evitando um relacionamento de muitos para muitos.

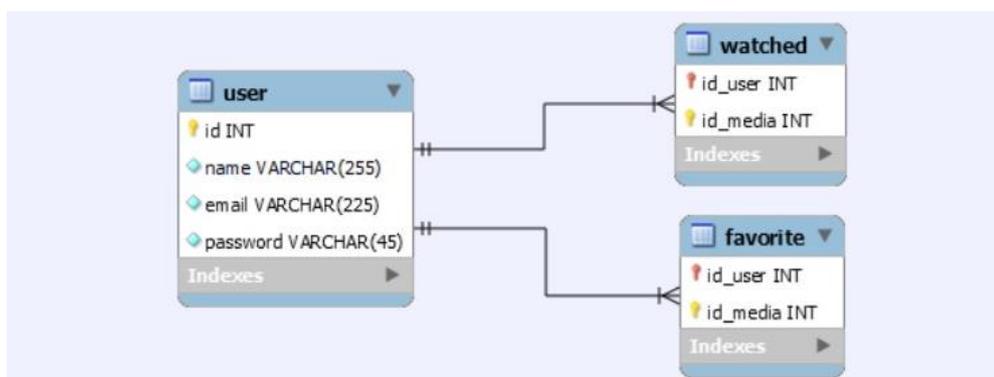
Figura 12 - Modelo entidade-relacionamento ilustrativo



Fonte: Elaborado pelo Autor

Na Figura 13 abaixo, está demonstrado o banco de dados definitivo, utilizado para manter os dados do usuário e seus filmes favoritos e assistidos.

Figura 13 - Modelo entidade-relacionamento



Fonte: Elaborado pelo Autor

3.5.2 Descrição de dados do sistema

Essa seção apresenta todas as informações que compõem a base de dados do sistema.

Tabela 1 - DD 001: Dados do usuário

Identificador

Descrição

DD 001

Dados do usuário

Identificador	Tamanho	Tipo	Formato	Descrição e observações
ID	5	Numérico	xxxxx	Campo único com somente número e com autoincremento.
Nome	225	Texto	Formato livre	Campo obrigatório.
E-mail	225	Texto	Formato livre	Campo obrigatório.
Senha	45	Texto	Formato livre	Campo obrigatório.

Fonte: Elaborado pelo Autor

Tabela 2 - DD 002: Dados dos títulos favoritados

Identificador

Descrição

DD 002

Dados dos títulos favoritados

Identificador	Tamanho	Tipo	Formato	Descrição e observações
ID do Usuário	5	Numérico	xxxxx	Campo único com somente número e com autoincremento.
ID da mídia	5	Numérico	xxxxx	Campo único com somente número e com autoincremento.

Fonte: Elaborado pelo Autor

Tabela 3 - DD 003: Dados dos títulos assistidos

Identificador

Descrição

DD 003	Dados dos títulos assistidos
---------------	------------------------------

Identificador	Tamanho	Tipo	Formato	Descrição e observações
ID do Usuário	5	Numérico	xxxxx	Campo único com somente número e com autoincremento.
ID da mídia	5	Numérico	xxxxx	Campo único com somente número e com autoincremento.

Fonte: Elaborado pelo Autor

3.6 *PROTOTIPAGEM*

A fase de prototipação desempenha um papel fundamental no desenvolvimento web, permitindo a visualização e teste da aparência e funcionalidades da interface do usuário precedendo a fase de implementação final. Esta seção consiste no detalhamento das decisões e processos relacionados à prototipação do catálogo de filmes web, incluindo a escolha da ferramenta, a identidade visual, o sistema de grid, a criação do estilo local, a utilização de frames e componentes, a criação de um protótipo interativo e garantia de responsividade.

3.6.1 **Identidade Visual**

A identidade visual do catálogo de filmes foi projetada para transmitir conceitos de inovação, exploração, modernidade e confiabilidade. Ela foi inspirada na empresa Norven, que foi responsável por lançar o desafio da implementação de um catálogo de filmes e possui os mesmos princípios em sua identidade visual. O logotipo também foi inspirado na marca, utilizando a mesma tipografia e proporções, acrescentando o símbolo do player de um filme e a ideia de movimento, o logotipo está exposto abaixo na Figura 14.

Figura 14 – Logotipo Movie



Fonte: Elaborado pelo Autor

Além disso, a escolha de utilizar os tons mais escuros da paleta como os prevaletentes na página web, juntamente com o roxo mais claro, lembra a sensação de estar em uma sala escura de cinema, com a iluminação vindo apenas do brilho da tela. Além disso, na psicologia das cores, o roxo é considerado capaz de inspirar criatividade e calma e carrega consigo uma aura de mistério e sofisticação. Em filmes, a tonalidade roxa é frequentemente empregada em cenas que buscam evocar uma atmosfera de fantasia, magia ou enigma, e foi utilizado no intuito de destacar a magia do cinema. Confira a paleta de cores na Figura 15, abaixo.

Figura 15 – Paleta de cores

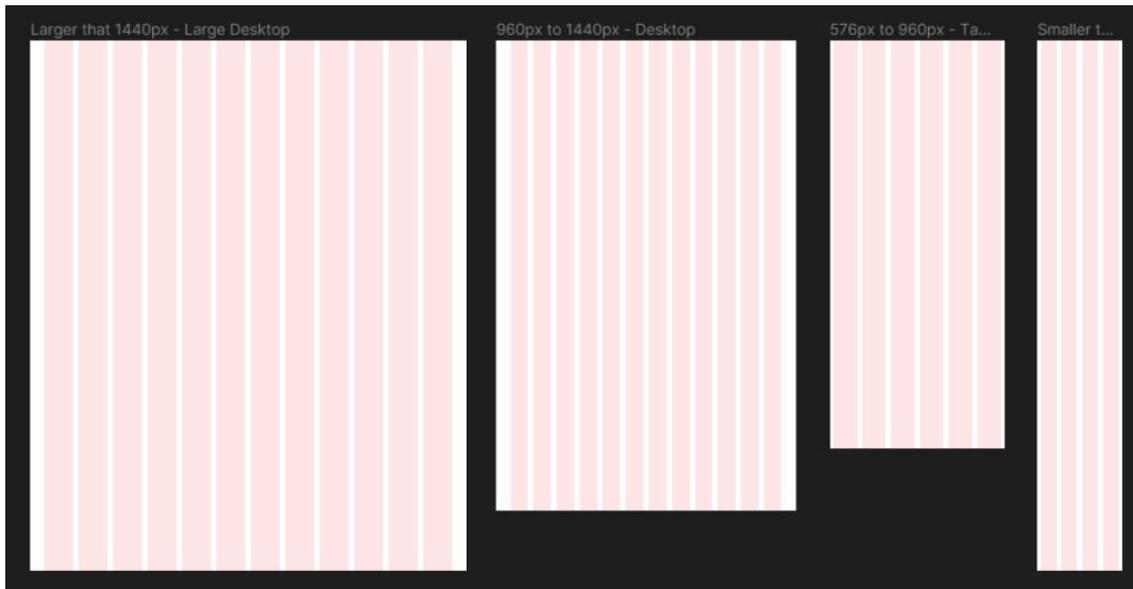


Fonte: Elaborado pelo Autor

3.6.2 Sistema de Grid de 8pts

Para garantir a consistência no design, facilitar a criação de designs responsivos e proporcionar uma melhor experiência do usuário, foi implementado um sistema que adota o grid de 8 pontos, explicado com mais detalhe no tópico 2.3.1 do referencial teórico. No desenvolvimento dos protótipos responsivos, mantendo o grid de 8 pontos, foram utilizadas 12 colunas para desktop. Esse grid foi fundamental na escolha de tamanhos de elementos e espaçamentos, seguindo a abordagem de “light grid”, também citada no referencial teórico. Confira os grids na Figura 16.

Figura 16 – Grid de 8 pontos



Fonte: Elaborado pelo Autor

3.6.3 Projeto no Figma

O Figma foi o software escolhido para a realização do protótipo, suas principais funcionalidades estão descritas no tópico 2.3.2. Deste documento. Os pontos de foco no processo foram: a utilização do estilo local na criação de todos os componentes, a utilização de frames e componentização.

visando manter a harmonia e identidade visual foram utilizados componentes mais retos, com uma leve curvatura, apenas para suavizar e dar ideia de movimento, realçando a identidade visual da marca. Também foram escolhidas as cores da identidade visual na criação do estilo local.

As imagens utilizadas foram escolhidas pensando em trazer ao usuário a sensação de estar em uma sala de cinema. Todos os elementos da tela são componentes e estão concentrados na pasta *assets*, separados por tamanhos de telas.

Todos os componentes são interativos, para melhor visualização do comportamento que o sistema deve ter, assim como a interação entre as telas. Os componentes foram feitos utilizando frames e auto layout sempre que possível. Foram feitos design para telas *mobile* (<576px), *tablet* (576px - 960px), *desktop* (960px – 1440px) e *large desktop* (>1440px).

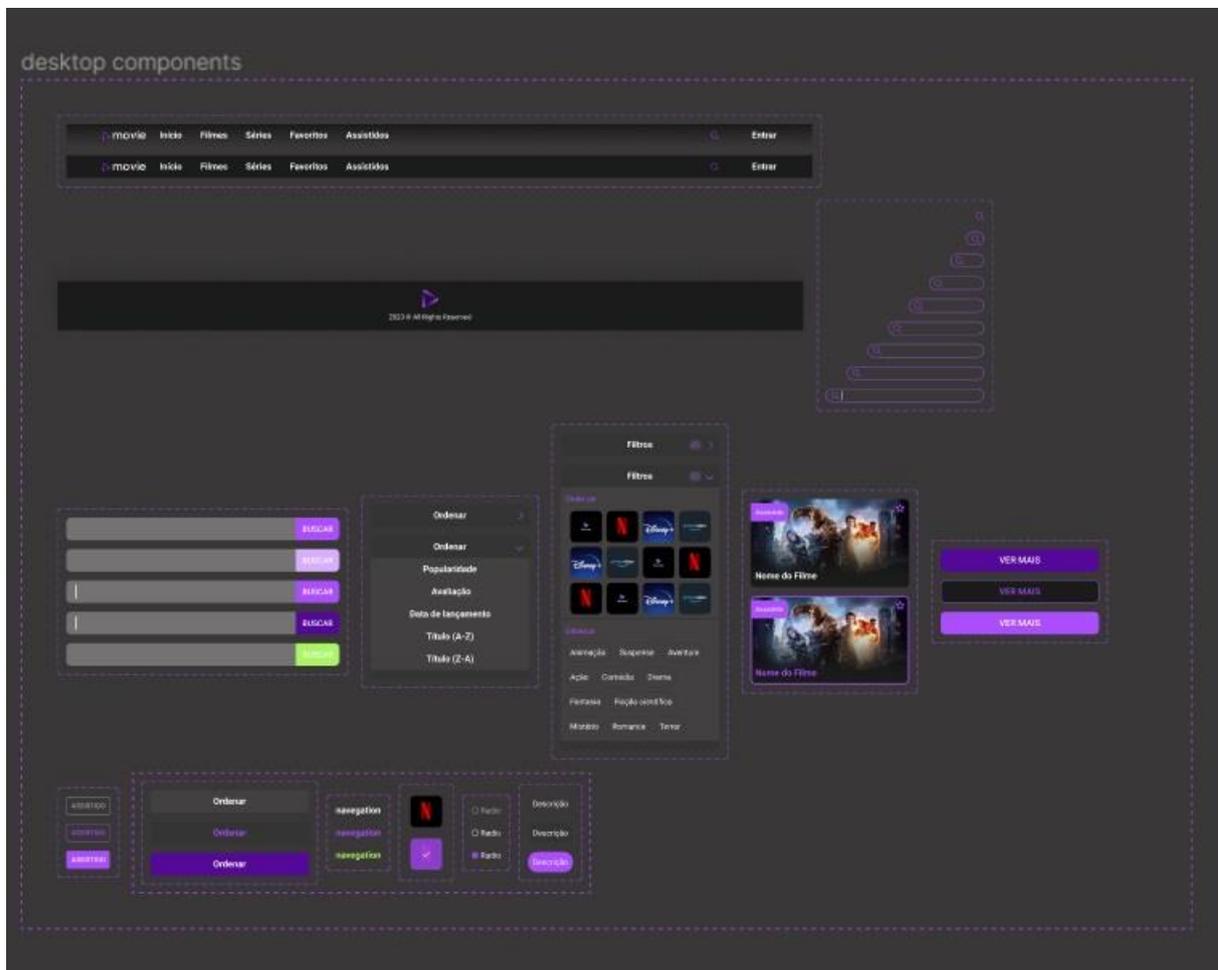
Abaixo, a figura 18 mostra os componentes e seus estados, para cada tamanho de tela foram feitos seus respectivos componentes. Na Figura 17 é possível visualizar a prototipagem das telas em seus diferentes tamanhos.

Figura 17 – Telas Figma



Fonte: Elaborado pelo Autor

Figura 18 – Componentes Figma



Fonte: Elaborado pelo Autor

O protótipo completo do figma, contendo todas as telas componentes e interações está disponível no link:

<https://www.figma.com/file/x1m6elmGc5HX6Teo1RmJf2/TCC---cat%C3%A1logo-de-filmes?type=design&node-id=0%3A1&mode=design&t=6fZP31fhnHL1w4g8-1>

4 DESENVOLVIMENTO DO SOFTWARE

4.1 BACK-END E ARMAZENAMENTO DE DADOS

O programa realiza requisições à API TMDb para obter informações sobre filmes e séries. Isso é feito utilizando a biblioteca Axios, que facilita a comunicação com APIs. As requisições são gerenciadas nas *actions* da Vuex store. A figura 19 apresenta como são feitas as requisições utilizando Axios nas *actions*:

Figura 19 – Requisições no Axios

```
},
loadMovies({ commit }) {
  movieDBInstance
    .get('discover/movie')
    .then((response) => {
      response.data.results.forEach((element) => {
        element.media_type = 'movie';
      });
      commit('loadMovies', response.data.results);
    })
    .catch((err) => console.error(err));
},
```

Fonte: Elaborado pelo Autor

Ao receber os dados da API, essas *actions* utilizam o método *commit* para acionar *mutations*, que por sua vez tratam os dados e os enviam para o *state* da store. Dessa forma, os componentes podem acessar esses dados por meio do método *dispatch* para as *actions* e utilizando os *states* para exibição na interface. A estrutura da store e as chamadas das *actions* e *mutations*, podem ser vistas abaixo nas Figuras 20 e 21.

Figura 20 – Estrutura da Store

```
export default createStore({
  state: {...},
  mutations: {...},
  actions: {...},
});
```

Fonte: Elaborado pelo Autor

Figura 21 – Chamadas da Store

```
Chamada do dispatch
this.$store.dispatch('setFavorites', item);

Chamada do commit
commit('loadMovies', response.data.results);

Chamada do state
return this.$store.state.favorites;
```

Fonte: Elaborado pelo Autor

O sistema conta com um banco de dados MySQL composto por três tabelas: 'user', 'favorites' e 'watched'. A tabela 'user' armazena informações essenciais do usuário, enquanto as tabelas 'favorites' e 'watched' relacionam o usuário às mídias que ele marcou como favoritas ou assistidas. Na seção 3.4 é possível visualizar com clareza a estrutura do banco de dados.

A conexão com o banco de dados é realizada pelo Node.js, e na pasta 'controllers', (observe a Figura 22) foram implementadas funções que utilizam essa conexão para executar operações como DELETE, INSERT e SELECT. As rotas associadas a essas operações são definidas no arquivo 'routes' (observe a Figura 23), permitindo que as funções utilizem o Axios para fazer requisições ao banco de dados.

Figura 22 – *Controllers Favorite*

```
import { db } from '../db.js';
export const getFavorites = (_, res) => {
  const q = 'SELECT * FROM favorite';
  db.query(q, (err, data) => {
    if (err) return res.json(err);
    return res.status(200).json(data);
  });
};
export const addFavorites = (req, res) => {
  const q = 'INSERT INTO favorite(id_user, id_media) VALUES(?)';
  const values = [req.body.idUser, req.body.idMedia];
  db.query(q, [values], (err) => {
    if (err) return res.json(err);
    return res.status(200).json('favorito adicionado com sucesso.');
```

Fonte: Elaborado pelo Autor

Figura 23 – Routes Favorite

```
JS favorites.js M X
src > API > routes > JS favorites.js > default
1 import express from 'express';
2 import { getFavorites, addFavorites, deleteFavorites } from '../controllers/favorites.js';
3 const router = express.Router();
4 router.get('/', getFavorites);
5 router.post('/', addFavorites);
6 router.delete('/:idUser/:idMedia', deleteFavorites);
7 export default router;

},
getFavorites({ commit }) {
  axios
    .get('http://localhost:8800/favorites')
    .then((res) => {
      commit('getFavorites', res.data);
    })
    .catch((err) => console.error(err));
},
```

Fonte: Elaborado pelo Autor

A persistência local de dados é implementada por meio do Vuex, utilizando o local store. Todos os atributos da store, que representam os dados da aplicação, são armazenados localmente. Isso garante que as informações persistam entre sessões e que o usuário possa retomar a interação de onde parou. A Figura 24 demonstra o armazenamento desses dados:

Figura 24 – Local Storage

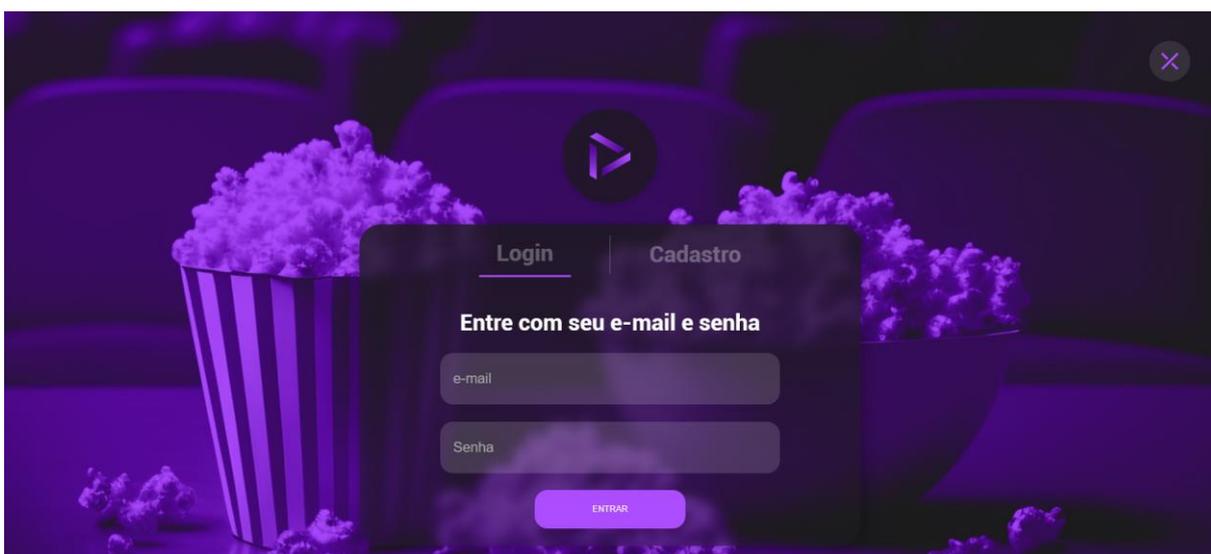
Storage	Key	Value
Local storage	sortOption	"popularity"
http://localhost:8080	selectedGenres	[99,37,14,27]
Session storage	favorites	[{"adult":false,"backdrop_pat...
http://localhost:8080	currentItem	{"adult":false,"backdrop_pat...
IndexedDB	watched	[{"adult":false,"backdrop_pat...
Web SQL	user	{"id":1,"name":"tayna","email...

Fonte: Elaborado pelo Autor

4.2 IMPLEMENTAÇÃO FRONT-END

O login torna-se imperativo para associar as interações do usuário, como favoritar ou marcar como assistido, aos seus dados específicos. Caso o usuário não esteja autenticado, é possível navegar pelo site, mas ações de favoritar ou marcar como assistido são salvas apenas localmente, com o sistema alertando sobre a possibilidade de perda desses dados caso o login não seja realizado. A tela de cadastro e login pode ser vista abaixo, na Figura 25.

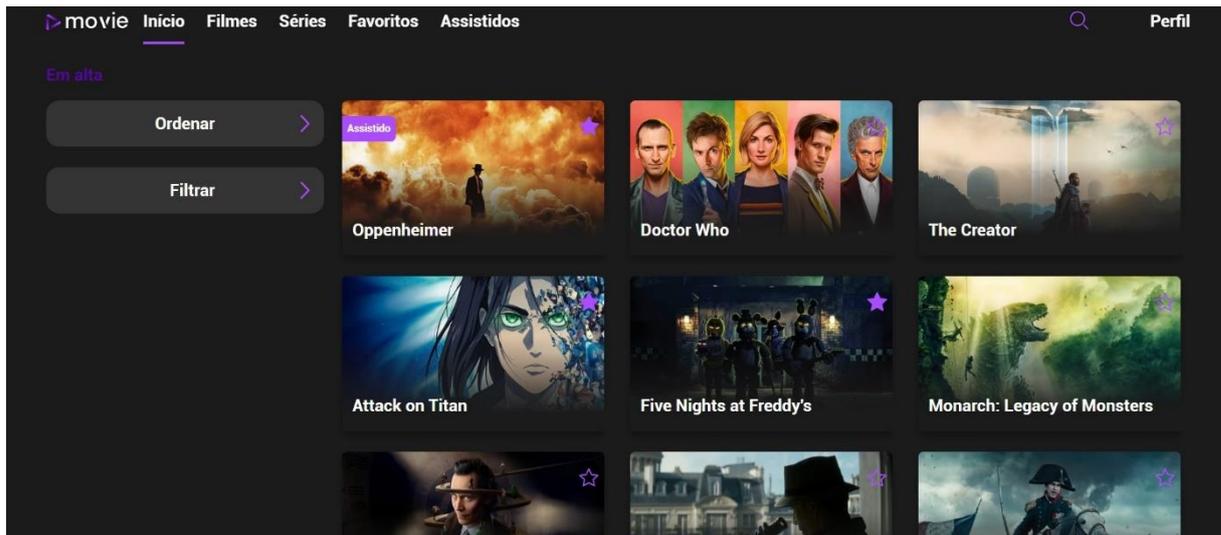
Figura 25: Login e cadastro



Fonte: Elaborado pelo Autor

As telas principais, como home, filmes, séries, favoritos e assistidos, compartilham uma estrutura comum, com a exceção da home, que possui uma seção adicional chamada "hero". Todas apresentam opções de ordenação e filtragem no canto superior esquerdo, e uma galeria exibindo títulos com informações como nome, imagem, *tag* de assistido (se aplicável) e um ícone de favorito preenchido quando o item é marcado como favorito. Observe a Figura 26:

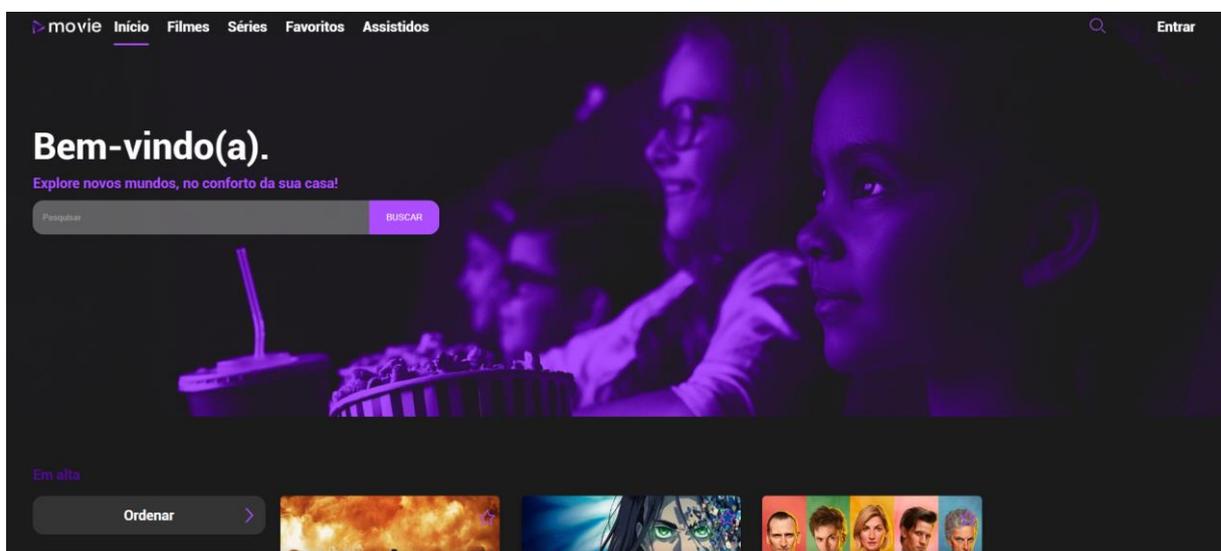
Figura 26: Tela Inicial



Fonte: Elaborado pelo Autor

A seção hero utiliza uma imagem de crianças assistindo filme no cinema (observe a Figura 27), trazendo a ideia da magia do cinema, e por cima, há uma frase de boas-vindas e outro modelo de barra de busca. Ao clicar em qualquer filme ou série, os usuários são direcionados para uma tela de detalhes. Nela, é possível explorar informações mais aprofundadas, como título, sinopse, data de lançamento, país de origem, duração, avaliação, gêneros, diretor, ator principal e a opção de marcar como assistido ou não.

Figura 27: Seção Hero



Fonte: Elaborado pelo Autor

A aplicação adota fortemente a componentização e a modularização. Filtros, ordenações, galerias de conteúdo, botões, barra de pesquisa, *footer*, *header* e outros elementos são componentes armazenados na pasta "componentes". Todos foram desenvolvidos utilizando Vue e a Options API, garantindo uma estrutura modular que facilita a manutenção e promove a reutilização de código.

As telas principais, incluindo home, filmes, séries, favoritos, assistidos, login e cadastro, estão organizadas na pasta "view". Todas foram construídas com Vue 3 e Options API, mantendo a modularização nas interações entre elas. Isso não apenas facilita o desenvolvimento contínuo, mas também promove uma arquitetura coesa e escalável.

Por fim, o site foi disponibilizado na internet, por meio da hospedagem gratuita no Netlify, citado no tópico 2.3.10 dessa monografia. Para hospedar o banco de dados, foi utilizado o Planetscale, pois além de fornecer várias ferramentas de forma gratuita, é possível fazer ligação entre ele e o site diretamente do Netlify, suas funcionalidades também estão descritas no tópico 2.3.11 desse trabalho. É possível acessar o site com o conteúdo do trabalho em <<https://steady-cobbler-6ff91e.netlify.app/>>

5 CONSIDERAÇÕES FINAIS

5.1 LIÇÕES APRENDIDAS

Durante o desenvolvimento deste projeto, diversas lições valiosas foram assimiladas. Primeiramente, a importância do aprendizado das ferramentas e de buscar estar sempre atualizada sobre as técnicas a serem utilizadas no desenvolvimento da aplicação. Além disso, a relevância do design e usabilidade ficou evidente, pois são eles que determinam a eficácia com que os usuários interagem com o catálogo.

O gerenciamento eficaz do projeto também se mostrou crucial, especialmente no que tange à organização do tempo e recursos disponíveis. Por fim, a colaboração e o feedback foram essenciais para o refinamento do projeto, provando que a construção de uma boa interface é um processo iterativo e colaborativo.

5.2 DIFICULDADES ENCONTRADAS

O percurso deste projeto não foi sem desafios. Entre as principais dificuldades encontradas, destacam-se os obstáculos técnicos, como problemas de compatibilidade e bugs, que exigiram uma abordagem detalhada para solução. A integração com APIs externas, particularmente com a do The Movie Database, também apresentou desafios significativos, testando a capacidade de adaptação e resolução de problemas. Além disso, um grande desafio foi distribuir o tempo entre o Trabalho de Conclusão de Curso, estágio e demais disciplinas da faculdade, mas com a ajuda dos professores e do serviço, foi possível finalizar chegar até aqui sem que nenhum lado fosse prejudicado.

5.3 IMPLEMENTAÇÕES FUTURAS

Olhando para o futuro, várias implementações podem ser consideradas para enriquecer o catálogo de filmes. Uma expansão das funcionalidades, como a inserção de mais filtros e a possibilidade do usuário comentar e avaliar filmes e séries. Outra funcionalidade possível é a personalização do perfil, que poderia oferecer uma experiência mais rica e diversificada aos usuários.

Melhorias na interface do usuário, por meio da implementação da responsividade, permitindo a adaptação a diferentes tamanhos de telas. Outras alterações baseadas no feedback recebido, poderiam tornar a navegação mais intuitiva e agradável.

Também é válido ressaltar que adotar um método de criptografia para senhas e dados importantes no banco de dados é uma funcionalidade de grande importância. A prática não foi adotada no seguinte projeto pois ele tem o intuito de ser apenas educativo, mas é imprescindível em softwares comerciais.

Além disso, a otimização do desempenho do site, visando um carregamento mais rápido e eficiente, é uma meta importante. Por fim, a implementação de práticas de acessibilidade e design inclusivo garantiria que o catálogo seja utilizável por uma gama mais ampla de usuários, independente de suas limitações ou necessidades especiais.

A implementação completa do sistema está disponível em: <https://github.com/Tayna-Carvalho/movie-catalog>

5.4 CONCLUSÃO

Este Trabalho de Conclusão de Curso representou uma jornada significativa no desenvolvimento pessoal, destacando a interseção entre a tecnologia avançada e a experiência do usuário no contexto de catálogos de filmes web. O projeto não apenas abordou a implementação técnica utilizando ferramentas contemporâneas como Vue3, Vuex, Vue Router, HTML5, CSS3, e JavaScript, mas também enfatizou a importância de uma abordagem centrada no usuário, refletindo sobre como o design e a funcionalidade impactam diretamente a experiência do usuário.

Através da análise comparativa de plataformas de streaming existentes, foi possível identificar tendências atuais e requisitos essenciais para um catálogo de filmes eficaz. Este estudo destacou a relevância de uma interface intuitiva e uma navegação fluída. A escolha da paleta de cores e o layout foram cuidadosamente pensados para maximizar a usabilidade e o apelo estético, garantindo que a plataforma seja não apenas funcional, mas também visualmente atraente.

A integração e implementação de tecnologias *front-end* mostraram-se desafiadoras, porém extremamente recompensadoras. O processo de desenvolvimento revelou a importância da arquitetura modular e do código reutilizável, enfatizando a eficiência e a escalabilidade no desenvolvimento de aplicações web e a utilização de APIs.

Este TCC também serve como um estudo de caso sobre como práticas de desenvolvimento web pode ser aplicadas no setor de entretenimento. Ele ilustra como as soluções tecnológicas podem ser utilizadas para criar plataformas digitais que valorizam o desenvolvimento e buscam uma boa experiência do usuário.

Em conclusão, este trabalho buscou destacar. Este projeto não apenas buscou fornecer um modelo valioso para futuros desenvolvimentos na área, assim como reafirmar o potencial ilimitado da ciência da computação na melhoria e inovação da indústria do entretenimento digital.

6 REFERENCIAS

Axios. Documentação. Disponível em: <https://axios-http.com/ptbr/docs/intro/>. Acesso em: 02 de dezembro de 2023.

Beard, J. (2020). The Principles of Beautiful Web Design (4ª ed.). SitePoint. ISBN 9781925836370.

Bleicher, S. (2023). Cor Contemporânea: Teoria e Uso (3ª ed.). Routledge. <https://doi.org/10.4324/9781003242741>.

Collins, M. J. (2017). Pro HTML5 with CSS, JavaScript, and Multimedia: Complete Website Development and Best Practices. Apress. ISBN 978-1-4842-2462-5.

Date, C. J. (1989). A Guide to the SQL Standard: A User's Guide to the Standard Relational Language SQL. Addison-Wesley Pub. Co. ISBN 0201502097.

Figma. (2023). Best Practice Guides. Disponível em: <https://www.figma.com/best-practices/guides/>. Acesso em: 30 de setembro de 2023.

HBO Max. Disponível em: <https://www.hbomax.com/br/pt/>. Acesso em: 02 de dezembro de 2023.

Hughes-Croucher, T.; Wilson, M. (2012). Node: Up and Running: Scalable Server-Side Code with JavaScript. O'Reilly Media. ISBN 9781449398583.

IMDb. Disponível em: <https://www.imdb.com/>. Acesso em: 02 de dezembro de 2023.

Kenzie. 8pt grid: o que é e para que serve? Disponível em: <https://kenzie.com.br/blog/8-pt-grid-o-que-e-e-para-que-serve/>. Acesso em: 02 de dezembro de 2023.

Letterboxd. Disponível em: <https://letterboxd.com/>. Acesso em: 02 de dezembro de 2023.

MDN. (2023a). HTML: HyperText Markup Language. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/HTML>. Acesso em: 30 de setembro de 2023.

MDN. (2023b). CSS: Cascading Style Sheets. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/CSS>. Acesso em: 30 de setembro de 2023.

MDN. (2023c). JavaScript. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Acesso em: 30 de setembro de 2023.

MySQL. Documentação. Disponível em: <https://www.mysql.com/>. Acesso em: 02 de dezembro de 2023.

Netflix. Disponível em: <https://www.netflix.com/>. Acesso em: 02 de dezembro de 2023.

Netlify (2023). *Connect everything. Build anything.* Disponível em: <https://www.netlify.com/>. Acesso em 30 de novembro de 2023

Node.js. Documentação. Disponível em: <https://nodejs.org/en/docs/>. Acesso em: 02 de dezembro de 2023.

PlanetScale (2023). *PlanetScale is the world's most advanced MySQL platform.* Disponível em: <https://app.planetscale.com/>. Acesso em 30 de novembro de 2023

Prime Video. Disponível em: <https://www.primevideo.com/>. Acesso em: 02 de dezembro de 2023.

Robbins, Jennifer Niederst. Learning Web Design: A Beginner's Guide to (X)HTML, Style Sheets, and Web Graphics. O'Reilly Media, 2007. ISBN 9780596527525.

Sharma, V.; Tiwari, A. K. A Study on User Interface and User Experience Designs and its Tools. World Journal of Research and Review (WJRR), ISSN: 2455-3956, v. 12, n. 6, p. 41-44, jun. 2021.

Tegmark, Max. Life 3.0: Being Human in the Age of Artificial Intelligence. London: Vintage, 2018.

Tidwell, Jenifer. Designing Interfaces. 2ª ed. Sebastopol, CA: O'Reilly Media, Inc., 2011.

TMDb. Disponível em: <https://www.themoviedb.org/>. Acesso em: 02 de dezembro de 2023.

TMDb-API. Documentação. Disponível em: <https://developer.themoviedb.org/reference/discover-movie/>. Acesso em: 02 de dezembro de 2023.

Vue Router. (2023). The oficial Router for Vue.js. Disponível em: <https://router.vuejs.org/>. Acesso em: 30 de setembro de 2023.

Vuex. (2023). What is Vuex?. Disponível em: <https://vuex.vuejs.org/>. Acesso em: 30 de setembro de 2023.

Vue.js. (2023a). Vue.js. Disponível em: <https://vuejs.org/guide/introduction.html>. Acesso em: 30 de setembro de 2023.

Vue.js. (2023b). Options API. Disponível em: <https://vuejs.org/guide/introduction.html>. Acesso em: 30 de setembro de 2023.

Vue.js. (2023c). Components. Disponível em: <https://vuejs.org/guide/components/props.html>. Acesso em: 30 de setembro de 2023.

Vuex. (2023). What is Vuex?. Disponível em: <https://vuex.vuejs.org/>. Acesso em: 30 de setembro de 2023.

Vuex. (2023). What is Vuex?. Disponível em: <https://vuex.vuejs.org/>. Acesso em: 30 de setembro de 2023.

RESOLUÇÃO nº 038/2020 – CEPE

ANEXO I

APÊNDICE ao TCC

Termo de autorização de publicação de produção acadêmica

O(A) estudante Tayná Rodrigues de Sousa Carvalho
do Curso de Ciência da Computação, matrícula 20221002801242,
telefone: (77) 98123-6099 e-mail tayrs.carvalho@gmail.com,
na qualidade de titular dos direitos autorais, em consonância com a Lei nº 9.610/98 (Lei dos Direitos do Autor), autoriza a Pontifícia Universidade Católica de Goiás (PUC Goiás) a disponibilizar o Trabalho de Conclusão de Curso intitulado Explorando o desenvolvimento front-end: Um estudo de caso em catálogo de filmes, gratuitamente, sem ressarcimento dos direitos autorais, por 5 (cinco) anos, conforme permissões do documento, em meio eletrônico, na rede mundial de computadores, no formato especificado (Texto(PDF); Imagem (GIF ou JPEG); Som (WAVE, MPEG, AIFF, SND); Vídeo (MPEG, MWV, AVI, QT); outros, específicos da área; para fins de leitura e/ou impressão pela internet, a título de divulgação da produção científica gerada nos cursos de graduação da PUC Goiás.

Goiânia, 06 de Dezembro de 2023.

Assinatura do autor: Tayná Carvalho

Nome completo do autor: Tayná Rodrigues de Sousa Carvalho

Assinatura do professor-orientador: _____

Nome completo do professor-orientador: _____