Antônio Carlos Rodrigues da silva **2DSTB**

1. Contexto e Estratégias

I. Introdução

- **a. Apresentação do projeto:** O **Lixs** é uma plataforma web de catálogo e informações de filmes que permite aos usuários navegar por uma grande biblioteca de títulos, visualizar fichas técnicos e interagir com o conteúdo.

- **b. Objetivo do projeto:** Meu objetivo foi criar uma “comunidade” onde usuários comuns podem não apenas consumir conteúdo, mas também sugerir novos filmes e edições e os administradores gerenciam todo o catálogo para garantir qualidade e a vericidade.

II. Audiência

- **a. Quem lerá essa documentação?**
  Desenvolvedores (Front-end e Back-end), Designers de Interface e a avaliadora do projeto somativo (Mariany Morais).

III. User Research e Público-alvo

- **a. Quem é o usuário?**
  - Entusiastas de cinema que buscam descobrir novos filmes, visualizar informações detalhadas (sinopse, elenco, diretor e etc.) e contribuir com o catálogo.

- **b. Como foi estabelecido?**
  Estabeleci este perfil baseado na missão da empresa de conectar pessoas a histórias que importam cinematograficamente e nos valores de "Comunidade".

IV. Princípios do Design

- **a Qual é o foco do design?**
  - **Imersão na escuridão do novo:** O design utiliza predominantemente fundo preto (#000 e #141414) para remeter à experiência de cinema e reforçar o futurismo através de cores como (#CD39E7 e #00D8FF).

  - **Identidade Visual Futurista:** Uso da fonte Iceland para títulos e logotipos, criando uma estética moderna e tecnológica.

- **Simplicidade:** Navegação clara através de uma Barra de Navegação fixa e Cards de filmes padronizados.

- Acessibilidade: Tudo foi pensado para que fosse acessível a todos os públicos então fiz uso de ferramentas para verificar os níveis de contraste e utilizei a semanticidade ao meu favor.

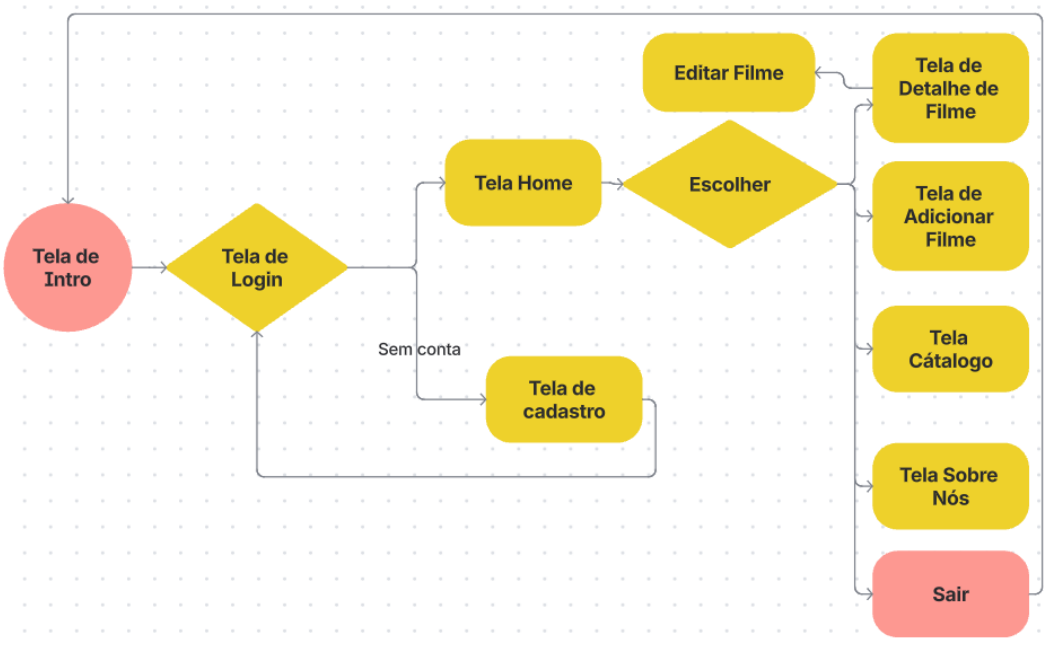
2. Ideação e Estrutura

I. Exploração do Conceito

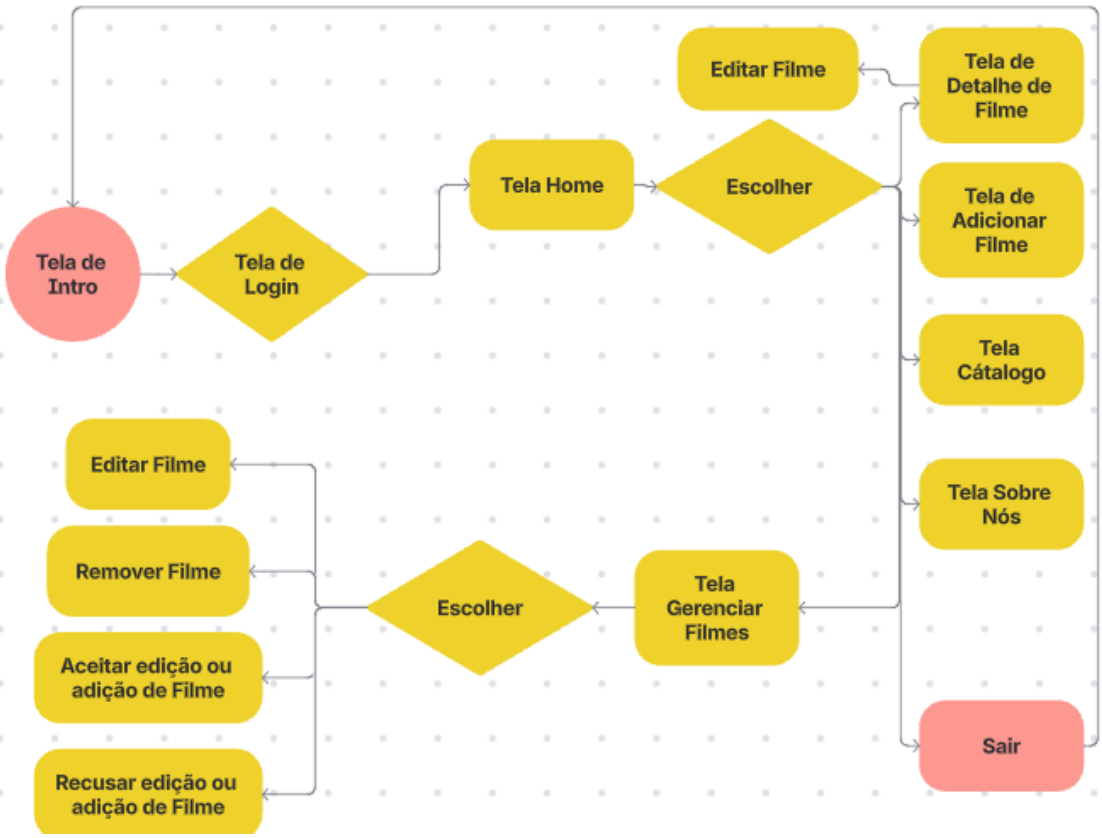
- **a. Ideias iniciais:** A ideia veio a partir de uma atividade pedagogica com algumas regras então para que isso pudesse ser feito utilizei componentes react e os organizamos nas paginas de acordo com o que achei ideal para a navegação, utilizamos no Back-End Python puro e Mysql para o nosso banco, para se adequar as regras e a partir disso criei a lixs com aspecto futurista .

II. Como a arquitetura funciona

- **a. Mapa do site (Estrutura de Navegação):**
- **Usuário Comum:**



- **Usuário Administrador:**



III. Versionamento e iterações

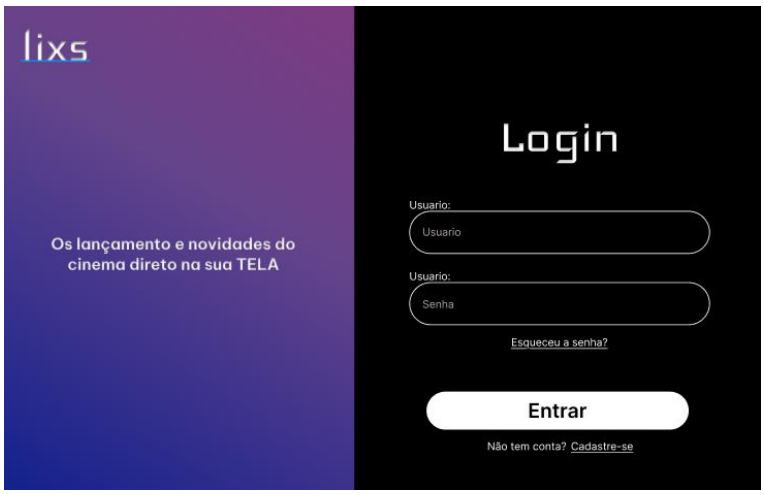
- **a. Mudanças no projeto:**
  O sistema evoluiu para incluir um fluxo de aprovação, onde filmes adicionados ou editados por usuários comuns entram como "Pendentes" no banco de dados antes de irem para o catálogo oficial (`filme`).

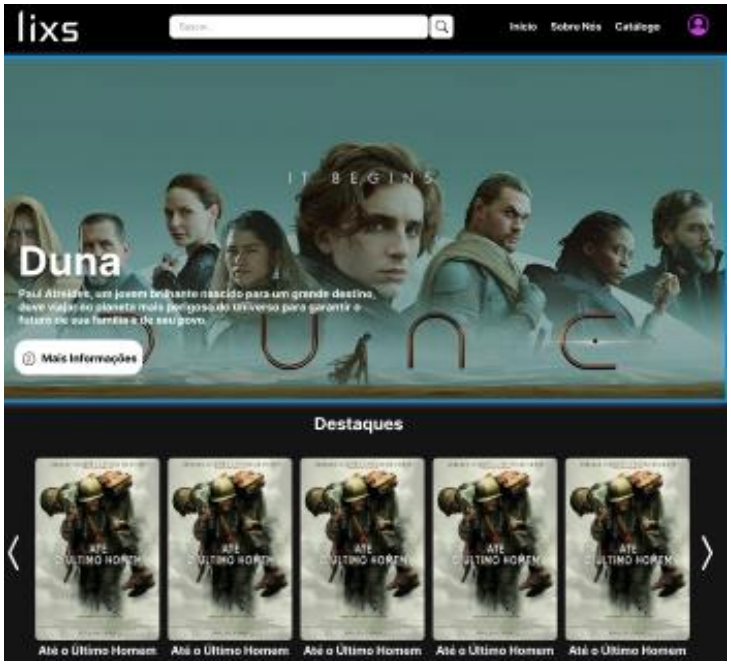
  A ideia é que os botões principais fossem um gradiente nas nossas cores mas por uma questão de contrate as cores dos botões foram reformuladas.

  Logo tambem foi alterado para branca por conta de contraste antes ela seria gradiente.
  Importante resaltar que durante todo o projeto tiveram refatorações para se adptar aos criterios de avalição desde criação de novas tabelas até a implementação do singletom no backend

3. Artefatos de Design

I. Wireframes e Protótipos

- **a. Ferramenta:** O design visual e o fluxo foram prototipados utilizando o Figma.
- **Link do Projeto:** Figma - Lixs
  Login
  

  Home
  

II. Error States (Estados de Erro)

- **a. Feedback ao usuário:**
  - **Login/Cadastro:** Mensagens de erro aparecem em vermelho (#ff6b6b) logo abaixo dos campos de input quando há falha na autenticação ou senhas divergentes.

- **Carregamento de Dados:** Se a API falhar, uma mensagem centralizada informa "Erro de conexão. O servidor Python está rodando?" ou a mensagem específica retornada pelo back-end.
- **Imagem Quebrada:** Nos cards de filmes, se a imagem do pôster falhar, um *placeholder* genérico é exibido automaticamente.

III. Acessibilidade

- **a. Critérios seguidos:**
  - Uso de tags semânticas (`<nav>`, `<main>`, `<article>`, `<figure>`, `<footer>`).
  - Atributos `aria-label` em botões que contêm apenas ícones (ex: botões de editar/remover e setas do carrossel).
  - Contraste alto entre fundo preto e texto branco/cinza claro.
  - Atributos alt em imagens.

4. Sistemas de Design e Padrões Visuais

I. Componentes do Sistema de Design

- **a. Qual o sistema seguido?** Sistema Próprio (CSS) com auxílio de bibliotecas específicas.
  - **Ícones:** Bootstrap.
  - **Carrossel:** React Slick.

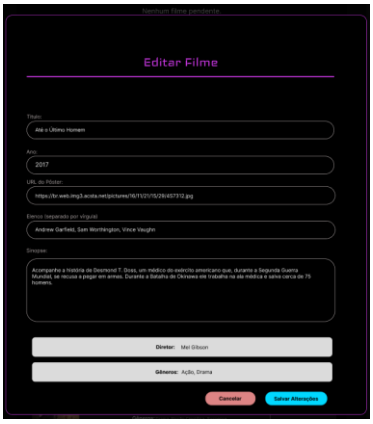
II. UI Patterns (Padrões de Interface)

- **a. Elementos repetidos:**
  - **Nav Bar:**

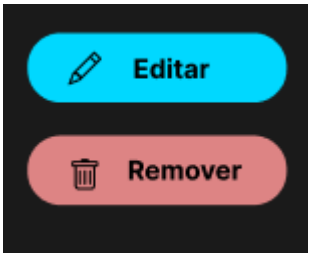
  - **Footer:**

- **Cards de Filmes:**

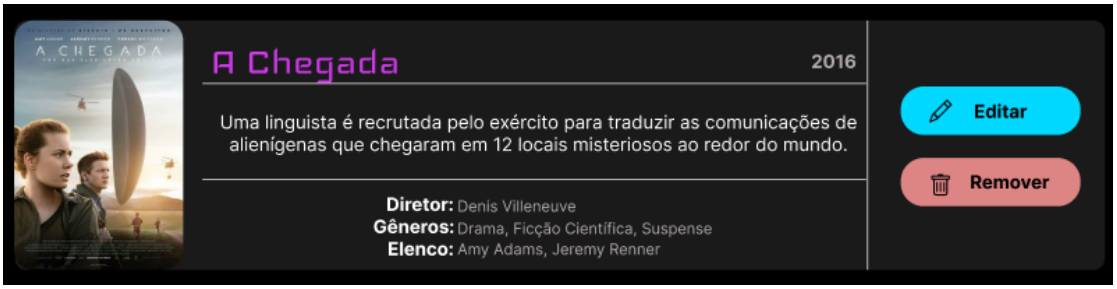
- **Modais:**



- **Botões:**



- **Card de Edição de Filme:**

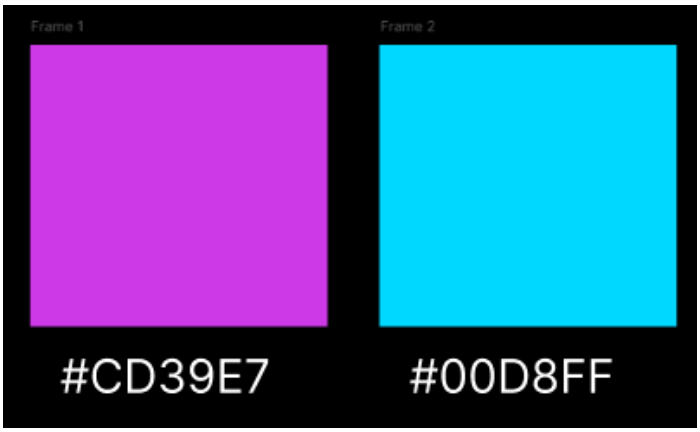


III. Guia de Estilo

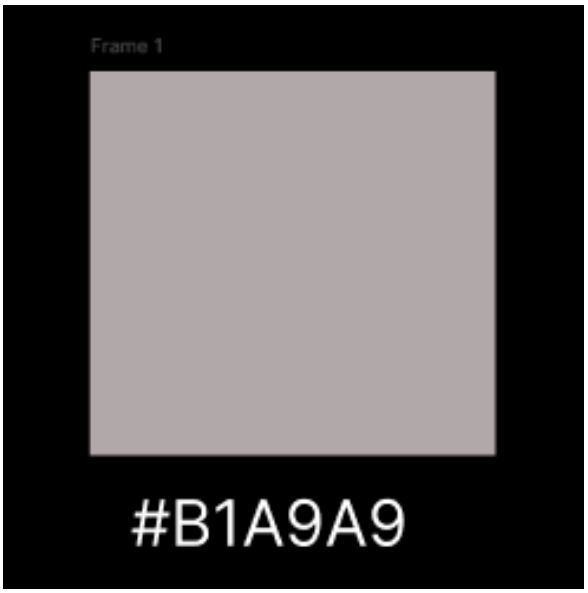
- **a. Cores usadas:**
  - **Primária (Fundo):** #000000 (Preto) e #141414 (Cinza).

  - **Destaque (Intro):** Gradiente #0C38C9 (Azul) para #8D0B99 (Roxo).

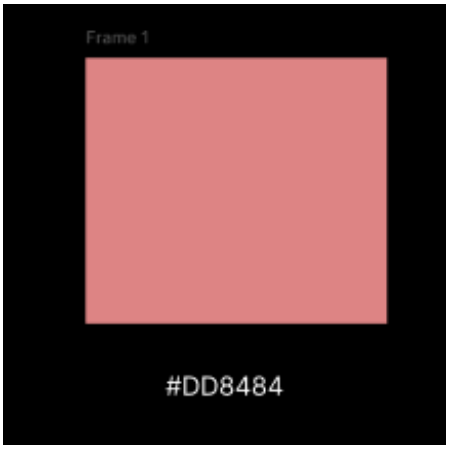
- **Ação/Foco:** #00d8ff (Ciano Neon) e #CD39E7 (Roxo Neon).



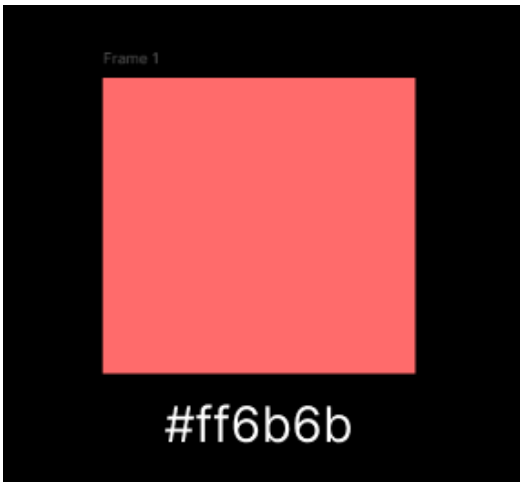
- **Texto Secundário:** #B1A9A9 (Cinza Claro).



- **Botões de rejeição/exclusão:** #DD8484(Vermelho muito Suave).



- **Erro:** #ff6b6b (Vermelho Suave).



- **b. Fontes:**
  - **Títulos/Logo:** 'Iceland', sans-serif (Estilo futurista).

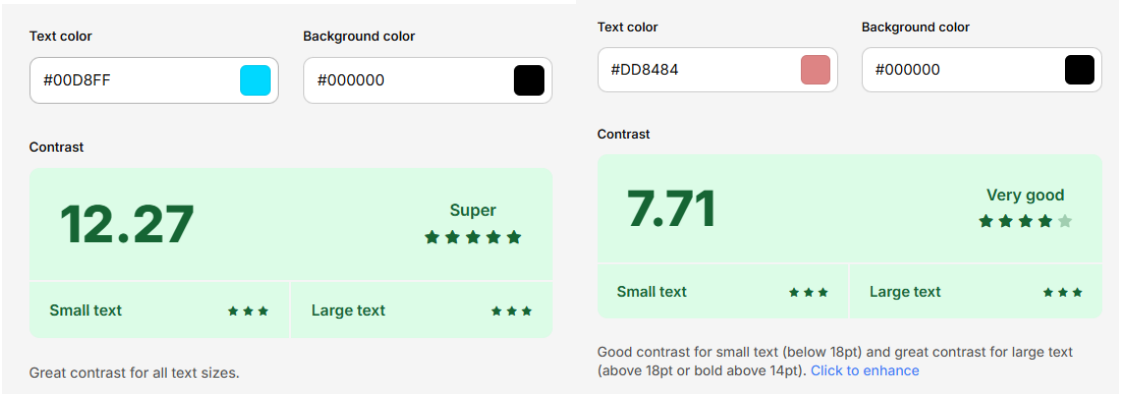
Iceland

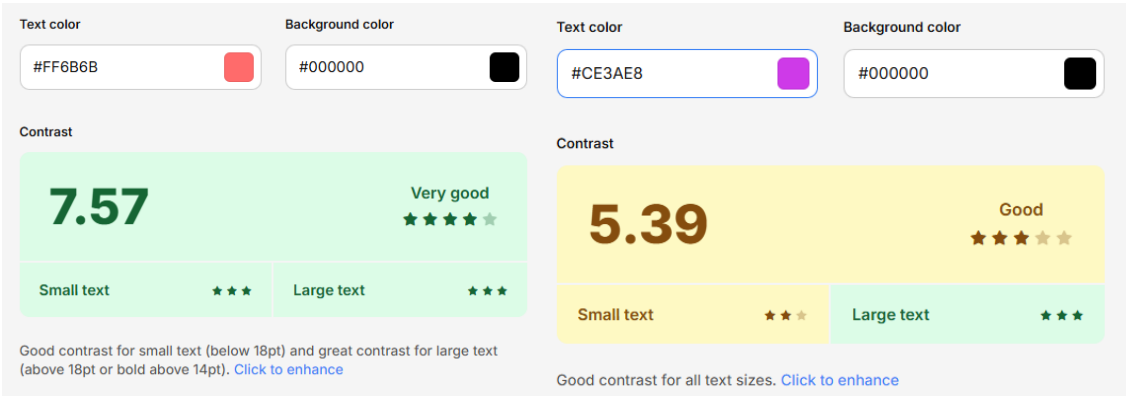
  - **Corpo:** System UI, Arial, sans-serif.

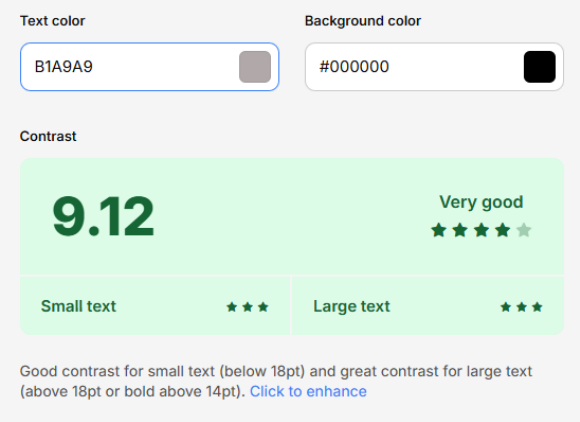
Sans Serif / inter

- **c. Logos:**
  - Texto "lixs" em minúsculo, usando a fonte Iceland, com aplicação de gradiente ou cor branca .
  - 

Testes de Contraste







ainda dentro das normas AA da WCAG!

5. Validação e Iteração

I. Estratégias de Validação

Testes de Usuário

O escopo atual de desenvolvimento não teve testes práticos ou entrevistas com usuários finais para validação da usabilidade.

Auditoria Técnica e Conformidade (feita por mim)

Para assegurar a qualidade da experiência sem os testes com usuários, o projeto foi submetido a uma auditoria técnica utilizando as ferramentas de desenvolvimento do Google Chrome (Lighthouse).

Esta análise focou em garantir que a estética "Dark" do sistema não comprometesse sua funcionalidade por conta do contraste:

- **Contraste e Legibilidade:** A auditoria priorizou a verificação das relações de cores entre o fundo escuro e os elementos de texto, assegurando que o contraste estivesse adequado para uma leitura confortável. Também foi validado a inclusão de textos alternativos (alt text) nos componentes visuais, como os pôsteres dos filmes e nas demais imagens.

- **Otimização de Desempenho:** Considerando a natureza de uma plataforma web, a velocidade de resposta foi um critério. A escolha do **Vite** para a aplicação garantiu tempos de carregamento menor sendo assim rapido.

- **Arquitetura Semântica (SEO):** A estrutura do código foi analisada para confirmar o uso apropriado das tags HTML. A hierarquia correta de elementos foi validada para otimizar e facilitar a navegação via teclado ou leitores de tela.

II. Arquitetura Tecnológica

O ecossistema do **Lixs** foi estruturado sobre três tecnologias diferentes, cada um gerenciando uma parte da aplicação:

- **Camada de Apresentação (Frontend):** Toda a interatividade e o visual da aplicação foram construídos com a biblioteca **React**. Para modernizar e acelerar o ambiente de desenvolvimento, utilizou-se o **Vite**, que oferece uma experiência de construção mais leve e performática.

- **Lógica de Servidor (Backend):** O processamento das requisições, a gestão de sessões (JWT) e as regras de negócio foram implementados em **Python** puro. A aplicação opera sem frameworks, utilizando apenas as bibliotecas padrão (http.server, socketserver) para manter a leveza e o controle total sobre o fluxo de dados (foi escolhida porque era regra).
- **Persistência de Dados:** O gerenciamento e armazenamento seguro das informações desde o catálogo de filmes até as credenciais de usuários e logs de edição são realizados através dode banco de dados **MySQL**.