|  |
| --- |
| **C-EFLE - J+ Digital**  **- FME96SI0004 –**  **Linguagens de Programação - Java** |
| **5425 - Projeto de tecnologias e programação de sistemas de informação** |

**Visualização sincronizada de filmes**



Formador: Sérgio Santos

Formando: Rostyslav RomanyshynÍndice

[1. Introdução 3](#_Toc19279)

[2. ESTRUTURA DO PROJETO 4](#_Toc16324)

[3.Diagrama de Interligação 5](#_Toc8275)

[4.Funcionalidades Principais 7](#_Toc22604)

[5.Spring 14](#_Toc10529)

[1.Dependências 14](#_Toc31574)

[6. Planificação, Calendarização e registo de execução das tarefas 15](#_Toc27085)

[7. Ficha de planeamento 17](#_Toc5523)

[8. Conclusão 18](#_Toc31087)

[9. Fontes de pesquisa (Webgrafia) 19](#_Toc29251)

[10. Anexos 20](#_Toc19649)

# Introdução

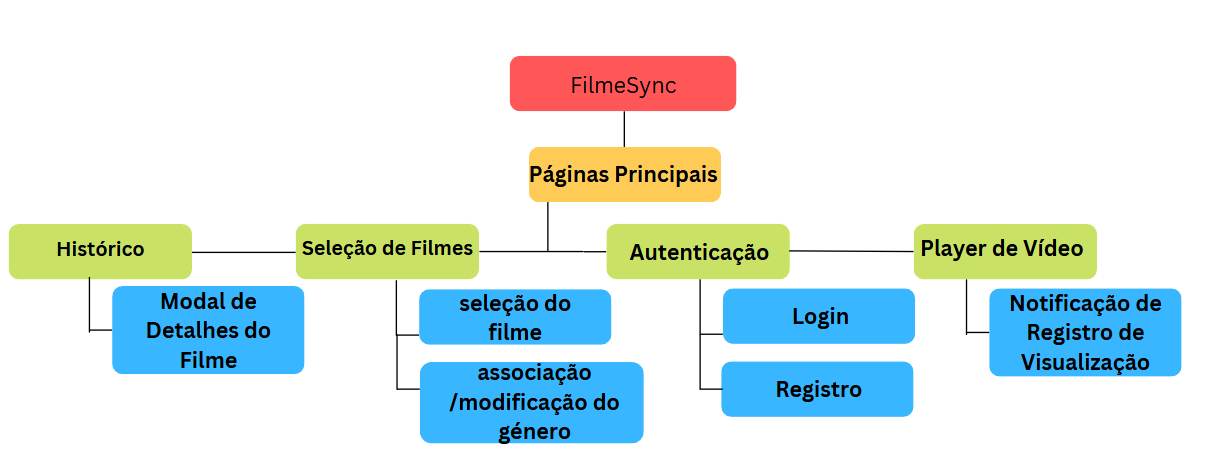
O FilmeSync é uma aplicação web desenvolvida para proporcionar uma experiência personalizada de streaming de vídeos, permitindo aos usuários assistir a filmes armazenados localmente através de uma interface intuitiva e moderna. Este sistema foi concebido como uma alternativa às plataformas de streaming comerciais, oferecendo controle total sobre o conteúdo e eliminando a necessidade de conexão constante à internet para assistir a filmes.

A aplicação foi desenvolvida utilizando o framework Spring Boot, que proporciona uma estrutura robusta para o desenvolvimento de aplicações web em Java. A arquitetura do sistema segue o padrão MVC (Model-View-Controller), garantindo uma separação clara entre a lógica de negócio, a apresentação e o controle de fluxo da aplicação. Para a interface do usuário, foram utilizadas tecnologias web modernas como HTML, CSS e JavaScript, complementadas pelo motor de templates Thymeleaf, que permite a integração perfeita com o backend Java.

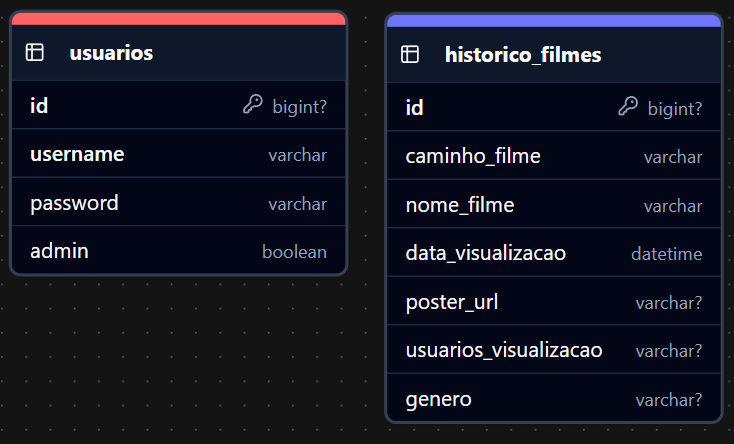
O FilmeSync oferece funcionalidades essenciais para uma plataforma de streaming, incluindo:

1. Autenticação e Autorização: Sistema de login e registro que diferencia entre usuários comuns e administradores, garantindo que apenas usuários autorizados possam acessar determinadas funcionalidades.
2. Seleção de Filmes: Interface para navegação e seleção dos filmes disponíveis, com suporte para filtragem e ordenação por diferentes critérios, como gênero e nome.
3. Player de Vídeo: Um reprodutor de vídeo integrado que oferece controles de reprodução e rastreia o progresso de visualização.
4. Histórico de Visualização: Registro automático dos filmes assistidos, incluindo informações como data de visualização e usuários que assistiram ao conteúdo.
5. Categorização por Gênero: Possibilidade de atribuir e filtrar filmes por gêneros, facilitando a organização e descoberta de conteúdo.
6. Integração com APIs Externas: Conexão com a API do IMDb para obtenção de posters e informações adicionais sobre os filmes.

# ESTRUTURA DO PROJETO



# 3.Diagrama de Interligação



**Tabela: Usuario**

|  |  |  |
| --- | --- | --- |
| Campo | Tipo | Descrição |
| id | BIGINT | Identificador único (Chave Primária) |
| username | VARCHAR | Nome de usuário único para login |
| password | VARCHAR | Senha do usuário |
| admin | BOOLEAN | Flag que indica se é administrador (true/false) |

**Tabela: HistoricoFilme**

|  |  |  |
| --- | --- | --- |
| Campo | Tipo | Descrição |
| id | BIGINT | Identificador único (Chave Primária) |
| caminhoFilme | VARCHAR | Caminho do arquivo de vídeo no sistema |
| nomeFilme | VARCHAR | Nome do filme extraído do caminho |
| dataVisualizacao | DATETIME | Data e hora da última visualização |
| posterUrl | VARCHAR(1000) | URL da imagem do poster obtida da API IMDb |
| usuariosVisualizacao | VARCHAR(1000) | Lista de usuários que assistiram ao filme |
| genero | VARCHAR | Gênero do filme (Ação, Comédia, etc.) |

**Relacionamentos**

Relacionamento Lógico: Usuario -> HistoricoFilme

Tipo: Muitos para Muitos (implementado através do campo usuariosVisualizacao)

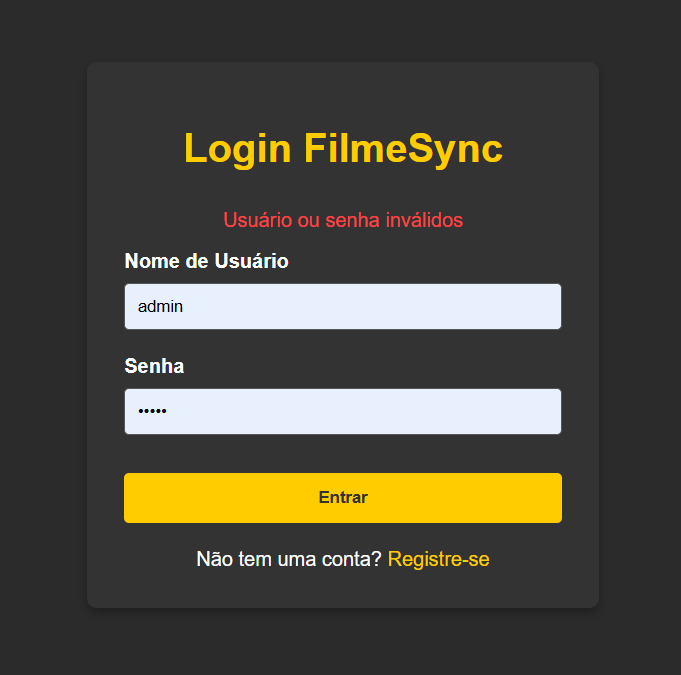
Descrição: Um usuário pode assistir a vários filmes, e um filme pode ser assistido por vários usuários

Implementação: O campo usuariosVisualizacao na tabela HistoricoFilme armazena uma lista de nomes de usuários

# 4.Funcionalidades Principais

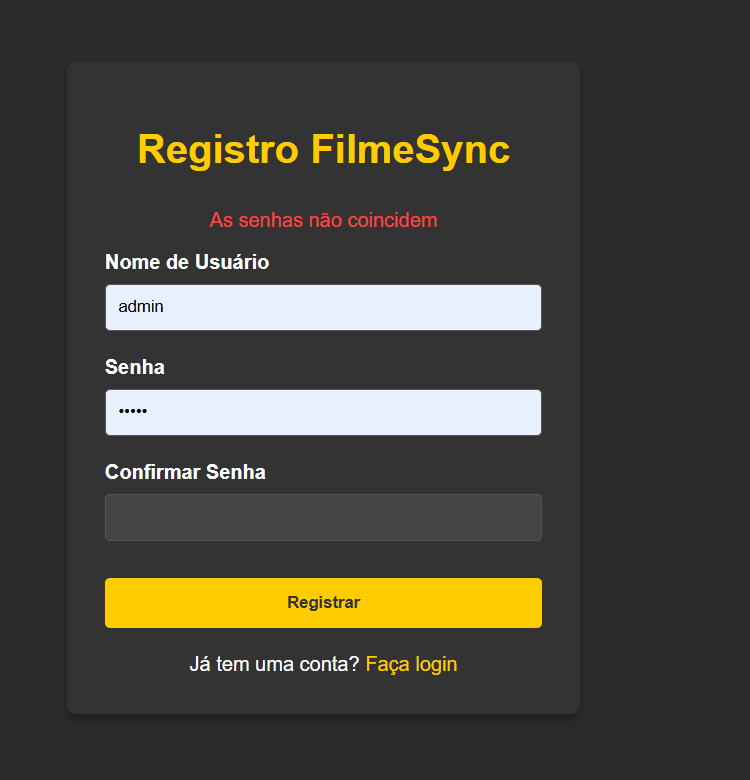
#### Página de Login

* Autenticação de usuários com nome de usuário e senha
* Redirecionamento automático para a página principal após login bem-sucedido
* Mensagens de erro para credenciais inválidas
* Link para registro de novos usuários



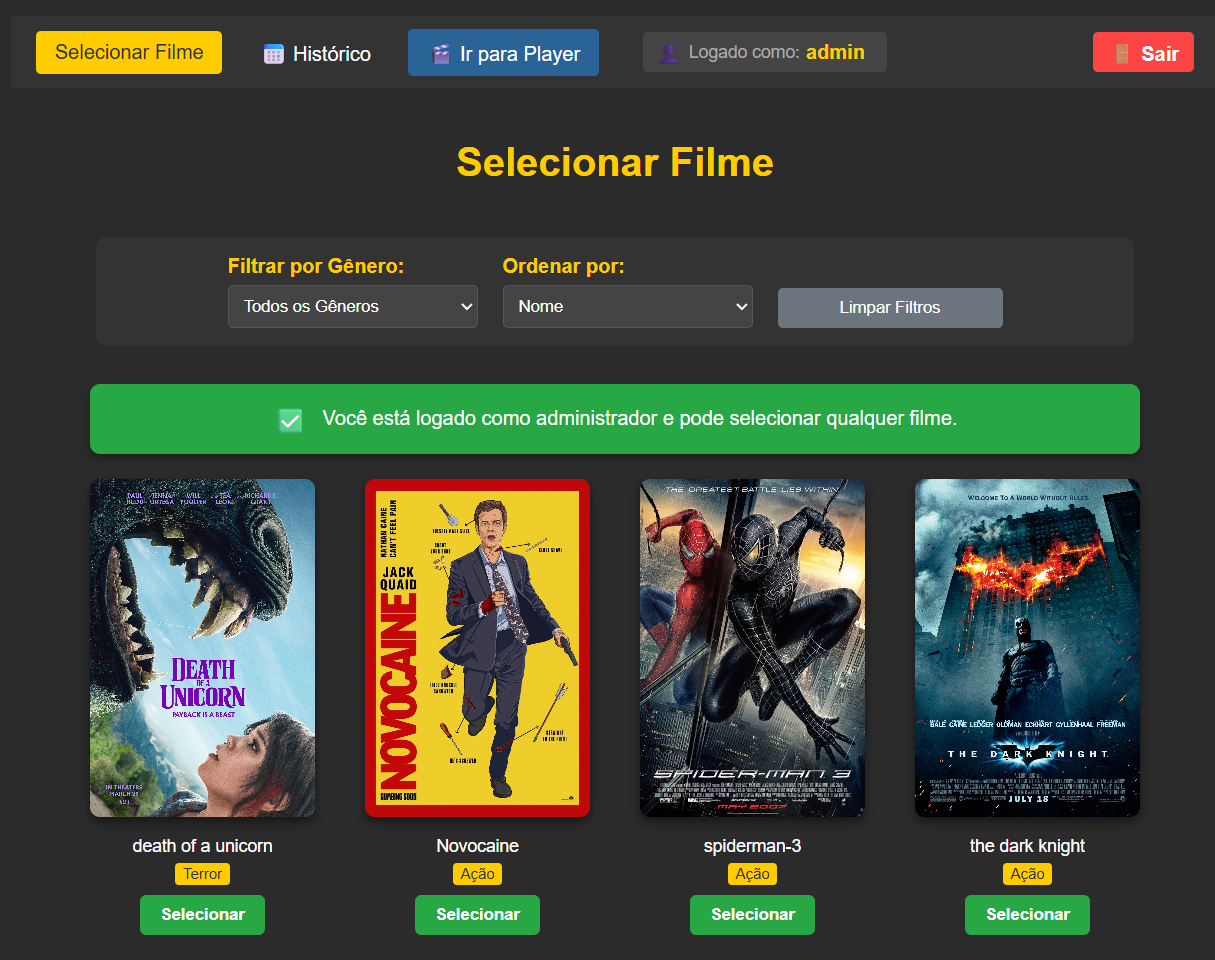
#### Página de Registro

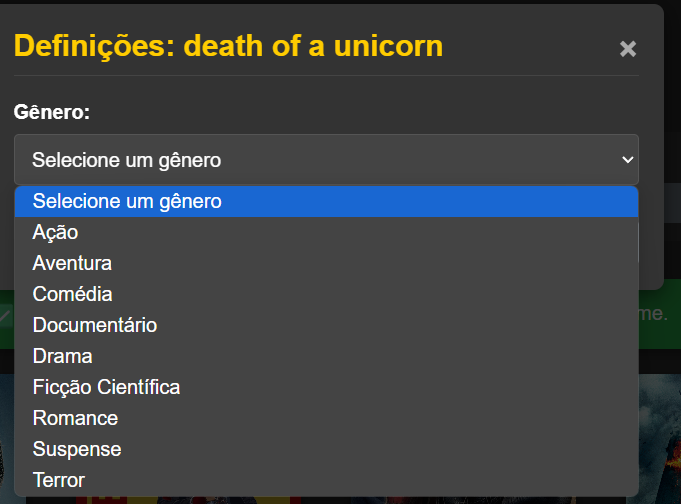
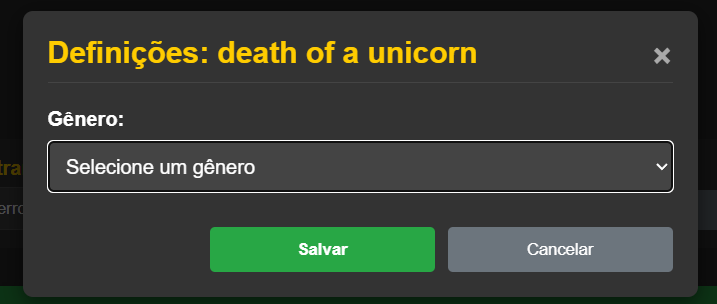
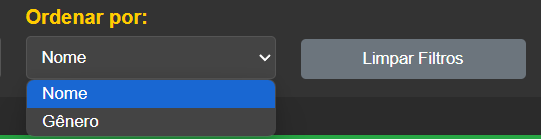
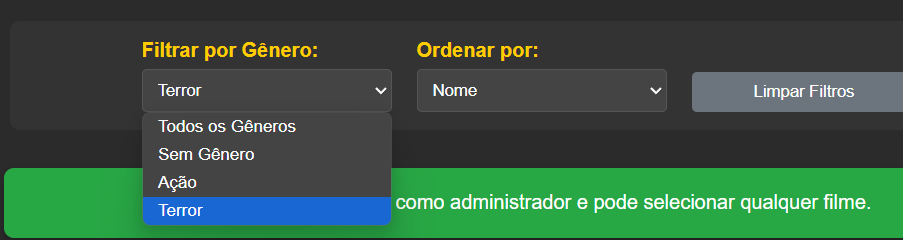
* Criação de novas contas de usuário
* Validação de dados de entrada (nome de usuário único, senha válida)
* Mensagens de feedback sobre o sucesso ou falha do registro
* Redirecionamento para a página de login após registro bem-sucedido



#### Página de Seleção de Filmes

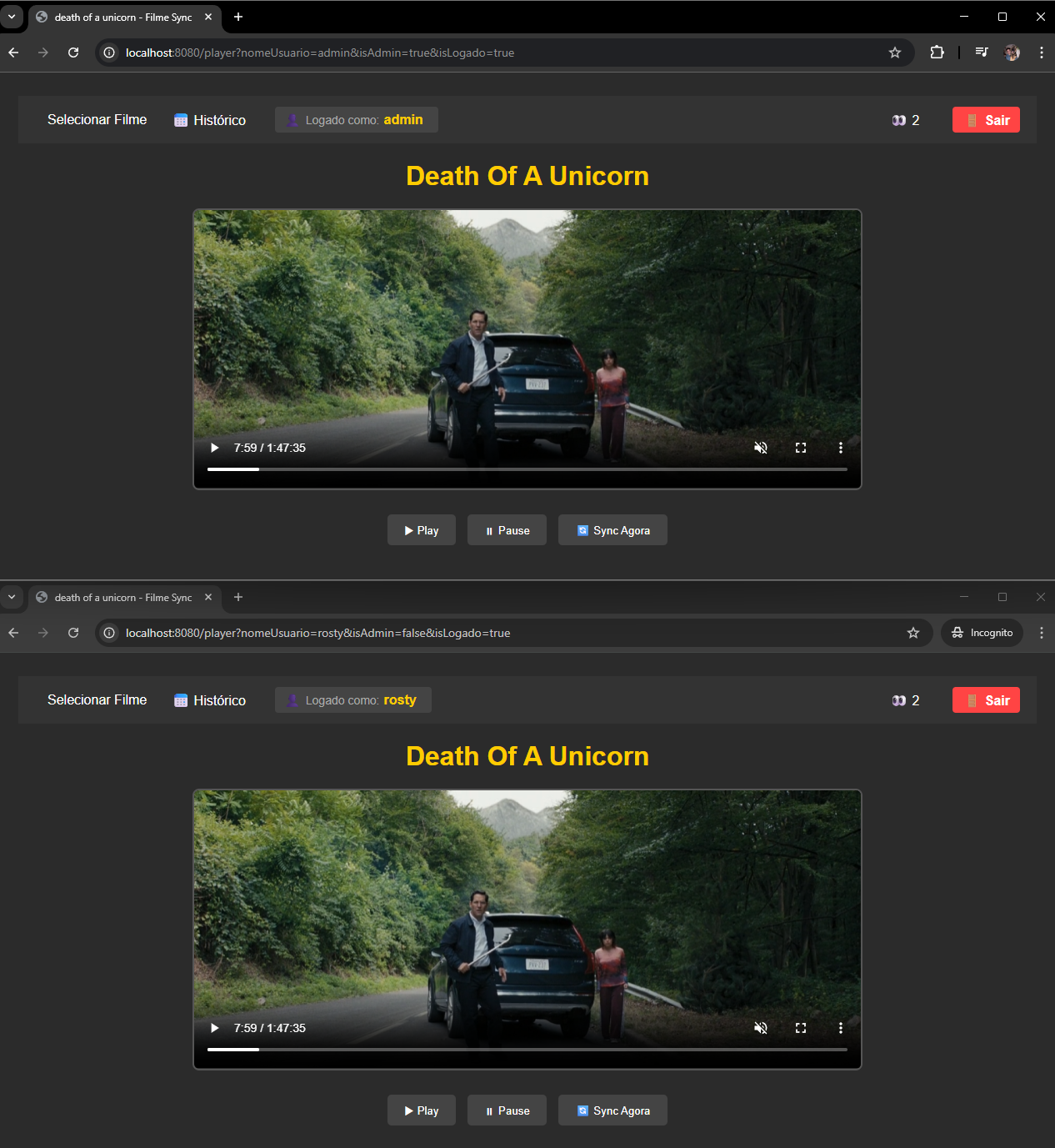
* Listagem de filmes disponíveis com posters obtidos da API do IMDb
* Filtragem de filmes por gênero através de um menu dropdown
* Ordenação de filmes por nome ou gênero)
* Botão de seleção para escolher o filme a ser reproduzido (apenas para administradores)
* Modal de configurações ao clicar na imagem do filme para:
  + Definir ou alterar o gênero do filme
  + Salvar as alterações no banco de dados
* Botão "Limpar Filtros" para resetar as opções de filtragem e ordenação





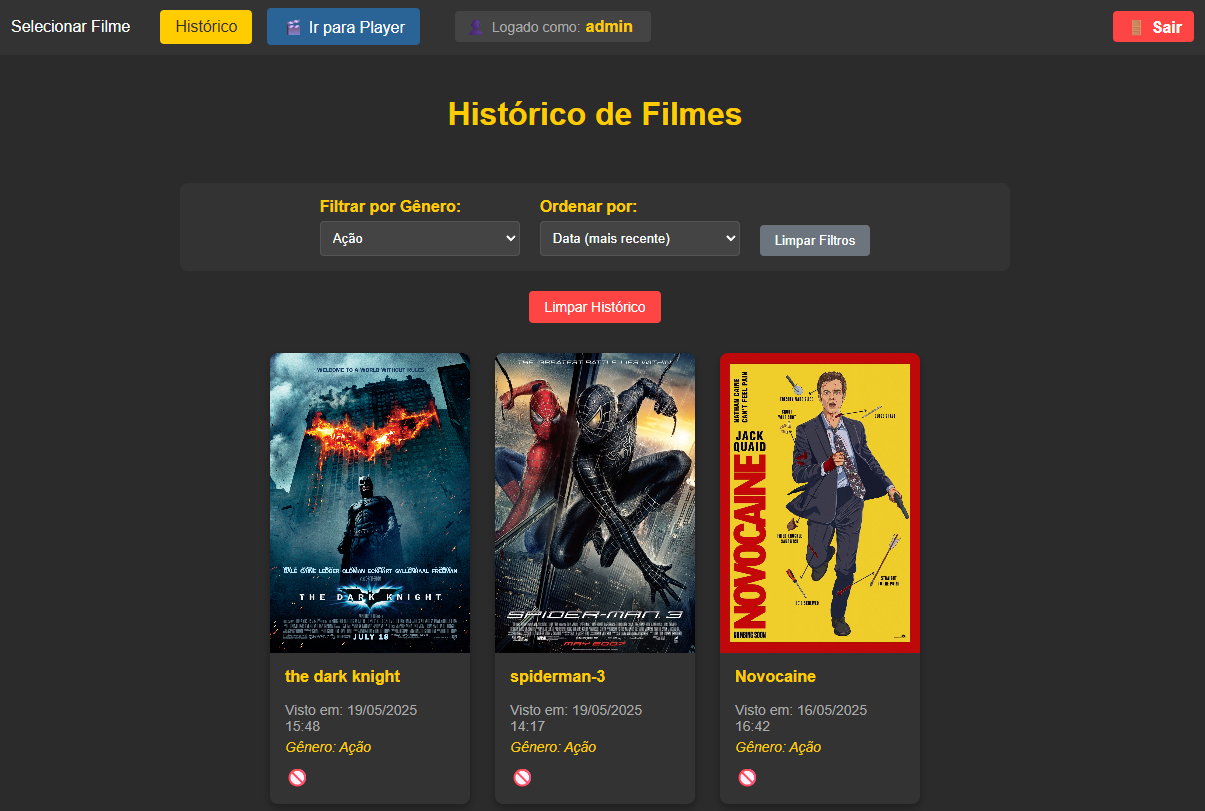
#### Página do Player de Vídeo

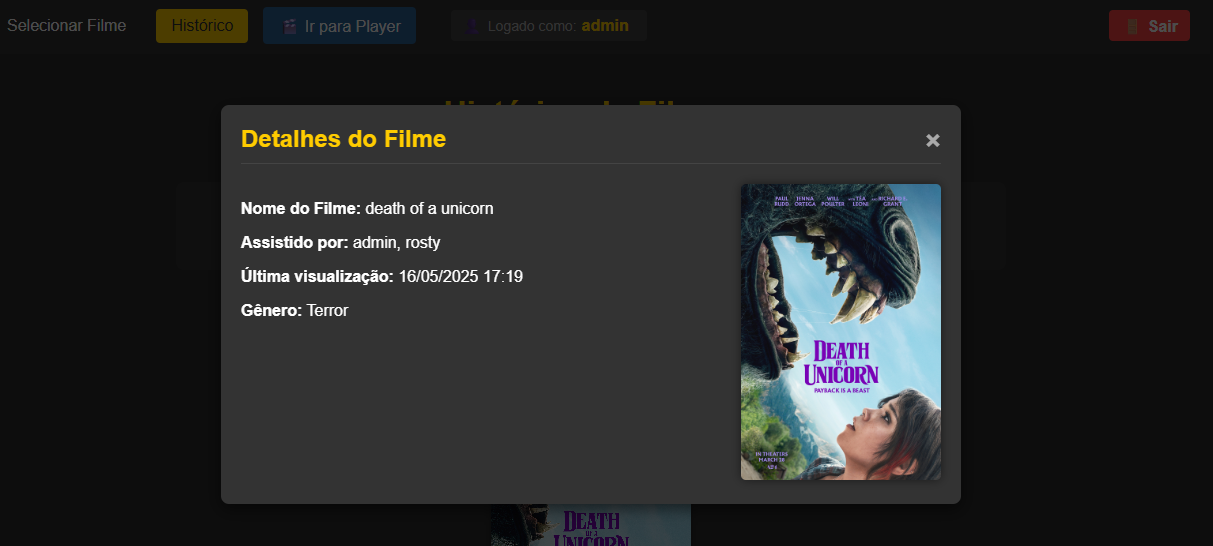
* Reprodução do filme selecionado em player HTML5 nativo
* Controles de reprodução (play, pause, volume, tela cheia)
* Exibição do nome do filme atual
* Contador de visualizações que rastreia o número de espectadores
* Registro automático de visualização quando o usuário assiste a pelo menos metade do filme e notificação discreta quando a visualização é registrada
* Exibição do usuário logado e seu status (admin ou usuário comum)



#### Página de Histórico

* Listagem de todos os filmes assistidos com posters, nomes e datas
* Filtragem do histórico por gênero através de um menu dropdown
* Ordenação do histórico por data, nome ou gênero
* Exibição do gênero de cada filme (se definido)
* Modal de detalhes ao clicar em um filme, mostrando:
  + Nome do filme
  + Lista de usuários que assistiram ao filme
  + Data da última visualização
  + Gênero do filme (se definido)
* Botão para remover filmes individuais do histórico e Botão "Limpar Histórico" para remover todos os registros
* Botão "Limpar Filtros" para resetar as opções de filtragem e ordenação





# 5.Spring

## ****1.Dependências****

* Spring Web → Framework para desenvolvimento de aplicações web, fornecendo servidor embutido e suporte a REST
* Spring Data JPA → Facilita acesso a dados e operações com banco de dados através de abstrações de alto nível
* Thymeleaf → Motor de templates para geração de páginas HTML dinâmicas no servidor
* MySQL Connector → Driver para conexão com banco de dados MySQL
* Spring DevTools → Ferramentas para desenvolvimento com reinicialização automática da aplicação
* Lombok → Reduz código boilerplate gerando getters, setters e construtores via anotações
* jQuery → Biblioteca JavaScript para manipulação do DOM e requisições AJAX
* Jackson → Processamento de JSON para comunicação com APIs externas
* Spring Test → Suporte para testes unitários e de integração
* Bootstrap (opcional) → Framework CSS para interfaces responsivas
* HttpClient (opcional) → Cliente para requisições HTTP à API do IMDb
* Spring Security (opcional) → Framework para autenticação e autorização

# 6. Planificação, Calendarização e registo de execução das tarefas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PLANIFICAÇÃO - CALENDARIZAÇÃO - REGISTO** | | | | | |
|  |  |  |  |  |  |
| **Nome do aluno** | | **Rostyslav Romanyshyn** | |  |  |
|  |  |  |  |  |  |
| Nr | Data | Titulo da tarefa | Descição da tarefa | Data Inicio da tarefa | Data Fim da tarefa |
| 1 | 29/04/25 | Escolha do projeto | Planeamento do projeto e das tecnologias que vão ser usadas para a implementação | 29/04/25 | 29/04/25 |
| 2 | 05/05/25 | Criação da DB | Instlação e imprementação do SQLite | 05/05/25 | 05/05/25 |
| 3 | 07/05/25 | --------- | Exames médicos - Não fiz nada | 07/05/25 | 07/05/25 |
| 4 | 09/05/25 | Troca para DB MySQL e crição de Controler | Troca de SQLite para MySQL que se enconta num Docker por falta de compatebilidade , Criação de metodos GET,POST,DELETE no Controler no Spring(MVC architecture ) | 07/05/25 | 07/05/25 |
| 5 | 12/5/25 | Implementação do player + WebSocket | Com a ajuda da dependency “spring-boot-starter-websocket” implementei um Sync entre os dispositivos quando dão start ou stop do filme e mudar o tempo de filme.   - Problemas encontrados:  O filme tem de começar em Mute(silencio), pois os browsers de hoje não permitem autoplay com o som para isso a solução foi começar sem som  - | 12/5/25 | 12/5/25 |
| 6 | 14/5/25 | Pagina selecionar-filme | Criação de uma pagina que mostre todos os filmes disponiveis, e ao selecionar um deles isso mais ser mostrado na pagina principal | 14/5/25 | 14/5/25 |
| 7 | 16/5/25 | Login | Criação de pagina para fazer login e registar. Agora só o utilizador admin conseguie escolher o filme | 16/5/25 | 16/5/25 |
| 8 | 19/5/2025 | Gênero + Filtragem | Na página de /selecionar-filme podemos filtrar por gênero e ao selecionarmos a capa do filme podemos escolher/modificar o gênero do filme | 19/5/2025 | 19/5/2025 |
| 9 | 20/5/2025 | Relatório | Relatório | 20/5/2025 | 20/5/2025 |
| 10 | 23/5/2025 | Relatório | Relatório e melhoria do código | 23/5/2025 | 23/5/2025 |
| 11 | 26/5/2025 | Relatório | melhoria do código | 26/5/2025 | 26/5/2025 |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |
| 17 |  |  |  |  |  |
| 18 |  |  |  |  |  |
| 19 |  |  |  |  |  |
| 20 |  |  |  |  |  |
| 21 |  |  |  |  |  |
| 22 |  |  |  |  |  |

# 7. Ficha de planeamento

**Ficha de Planeamento do Projeto**

|  |
| --- |
| **Título do Projeto:** |
| Visualização sincronizada de filmes |
| **Endereço do Projeto:** |
| Por defenir |
| **Nome do aluno** |
| Rostyslav Romanyshyn |

|  |
| --- |
| **Destinatários** |
| Feito para uso pessoal, mas vou meter no git publico no caso que alguém queira implementar /melhorar |

|  |
| --- |
| **Objetivo do Projeto** |
| O FilmeSync é uma aplicação web que permite visualizar e organizar filmes armazenados localmente, oferecendo uma experiência similar a plataformas de streaming comerciais. O sistema proporciona reprodução de vídeos, categorização por gênero, histórico de visualizações e gerenciamento de usuários, eliminando a necessidade de conexão constante à internet enquanto mantém recursos essenciais como posters de filmes e controle de acesso baseado em perfis. |

# Conclusão

O projeto FilmeSync alcançou o seu objetivo de criar uma plataforma de streaming local com interface intuitiva com funcionalidades essenciais. A implementação bem-sucedida de recursos como reprodução de vídeos, categorização por gênero, histórico de visualizações e controle de acesso demonstra a viabilidade de uma solução personalizada para gerenciamento e exibição de conteúdo audiovisual. O sistema oferece uma alternativa prática às plataformas comerciais, permitindo aos usuários assistir a filmes sem dependência constante de internet, enquanto mantém uma experiência visual atraente com posters e informações obtidas da API do IMDb. As tecnologias escolhidas, como Spring Boot e Thymeleaf, proporcionaram uma base sólida para o desenvolvimento, resultando em uma aplicação robusta e funcional que atende às necessidades tanto de administradores quanto de usuários comuns. O FilmeSync representa uma solução eficiente para quem busca organizar e desfrutar de sua coleção pessoal de filmes em um ambiente digital estruturado.

# 9. Fontes de pesquisa (Webgrafia)

<https://spring.io/projects/spring-framework>

<https://www.thymeleaf.org/documentation.html>

<https://www.w3schools.com/html/html5_video.asp>

<https://dev.mysql.com/doc/refman/8.0/en/>

# 10. Anexos

-- Criação do banco de dados FilmeSync  
CREATE DATABASE IF NOT EXISTS filmesync CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

-- Usar o banco de dados  
USE filmesync;

-- Tabela de Usuários  
CREATE TABLE usuarios (  
Id BIGINT AUTO\_INCREMENT PRIMARY KEY,  
username VARCHAR(255) NOT NULL UNIQUE,  
password VARCHAR(255) NOT NULL,  
 admin BOOLEAN NOT NULL DEFAULT FALSE  
);

-- Tabela de Histórico de Filmes  
CREATE TABLE historico\_filmes (  
 id BIGINT AUTO\_INCREMENT PRIMARY KEY,  
 caminho\_filme VARCHAR(255) NOT NULL,  
 nome\_filme VARCHAR(255) NOT NULL,  
 data\_visualizacao DATETIME NOT NULL,  
 poster\_url VARCHAR(1000),  
usuarios\_visualizacao VARCHAR(1000),  
genero VARCHAR(100)  
);

-- Inserção do usuário administrador padrão  
INSERT INTO usuarios (username, password, admin)   
VALUES ('admin', 'admin', TRUE)  
ON DUPLICATE KEY UPDATE admin = TRUE;