

# Manual Técnico

Unidade Curricular: Programação e Integração de Serviços

Projeto: HuntMovies (Plataforma de Gestão de Filmes e Series)

Ano Letivo: 2025/2026

Desenvolvido por : Iuri Silva(2024107621) e Raylson Costa (2024131366).

## Indice

<b>1. Introdução .....</b>	3
<b>2. Instalação e Configuração .....</b>	3
<b>2.1 Requisitos de Sistema.....</b>	3
<b>2.2 Guia de Instalação Rápida.....</b>	3
<b>3. Arquitetura de Software .....</b>	4
<b>3.1 Estrutura de Diretórios.....</b>	4
<b>4. Base de Dados .....</b>	4
<b>4.1 Modelo Relacional.....</b>	4
<b>4.2 Dados.....</b>	4
<b>5. Documentação da API REST .....</b>	5
<b>5.1 Conteúdos (/api/conteudos).....</b>	5
<b>5.2 Utilizadores e Autenticação (/api/auth &amp; /api/admin) .....</b>	5
<b>6. Integração com a API TMDB .....</b>	6

## 1. Introdução

O HuntMovies é uma aplicação web desenhada para a gestão e consulta de filmes e séries. O sistema permite aos utilizadores explorar filmes e séries, enquanto oferece aos administradores ferramentas de importação direta através da API externa The Movie Database (TMDB).

## 2. Instalação e Configuração

### 2.1 Requisitos de Sistema

- Runtime: Node.js
- Base de Dados: MySQL 8.0.
- Gestor de Pacotes: NPM.

### 2.2 Guia de Instalação Rápida

1. Dependências: No terminal do projeto, execute `npm install`.
2. Ambiente: Renomeie ou crie o ficheiro `.env` na raiz com as suas credenciais de base de dados e a sua `TMDB_API_KEY`.
3. Base de Dados: Importe o esquema em `database/dbprojectpis.sql`.
4. Execução: Utilize o comando `npm start`.

## 3. Arquitetura de Software

O sistema segue o modelo Cliente-Servidor. O Frontend comunica com o Backend via requisições assíncronas (Fetch API), e o Backend gere a persistência de dados.

### 3.1 Estrutura de Diretórios

Pasta/Ficheiro	Descrição
/config	Conexões com a BD e Axios para TMDB.
/routes	Endpoints da API REST segregados por módulo.
/middleware	Lógica de segurança (Autenticação JWT).
/public	Ficheiros estáticos (HTML, CSS, JS do cliente).
server.js	Inicialização do servidor Express.

## 4. Base de Dados

A base de dados dbprojectpis foi normalizada para garantir a integridade dos dados e a performance das consultas.

### 4.1 Modelo Relacional

As principais entidades incluem Utilizadores, Conteúdos (Filmes/Séries), Elenco, Reviews e Favoritos.

### 4.2 Dados

- utilizadores: id, username, email, password (hashed), is\_admin.
- conteudos: id, tmdb\_id, titulo, sinopse, poster\_url, tipo (movie/tv).
- reviews: id, utilizador\_id, conteudo\_id, avaliacao (1-10), comentario.

## 5. Documentação da API REST

Abaixo estão listados os principais endpoints para integração.

### 5.1 Conteúdos (/api/conteudos)

Método	Endpoint	Descrição	Acesso
GET	/	Lista todos os filmes e séries na BD local.	Público
GET	/:id	Detalhes completos de um conteúdo específico.	Público
POST	/importar-tmdb	Importa dados da TMDB para a BD local.	Admin
DELETE	/:id	Remove um conteúdo da base de dados.	Admin

### 5.2 Utilizadores e Autenticação (/api/auth & /api/admin)

Método	Endpoint	Descrição	Acesso
POST	/auth/login	Autentica utilizador e devolve Token JWT.	Público
GET	/admin/users	Lista todos os utilizadores registados.	Admin
DELETE	/admin/users/:id	Elimina a conta de um utilizador.	Admin

## 6. Integração com a API TMDB

O sistema utiliza a API da The Movie Database para enriquecer a base de dados local.

1. Pesquisa: O admin pesquisa títulos no backoffice.
2. Fetch: O servidor faz o pedido à TMDB: <https://api.themoviedb.org/3/movie/{id}>.
3. Persistência: Os dados (título, elenco, sinopse) são mapeados e guardados no MySQL para consulta offline.