

Instituto Politécnico de Setúbal  
Curso Técnico de Gestão e Programação de Sistemas Informáticos

# Manual Técnico



**CinePIS**

**CTeSP em Tecnologias e Programação de Sistemas de Informação  
Programação e Integração de Serviços**

**Samuel Furtado, Diogo Rolo, Rodrigo Milagre**

**Docente:** Fabiana Gloria



**REPÚBLICA  
PORTUGUESA**



# Índice

Introdução e Tecnologias.....	3
Estrutura do Projeto (File System) .....	4
Arquitetura de Dados (MySQL) .....	4
Web Services e Especificações Rest .....	5
Serviços Elementares (Gestão Interna) .....	5
Serviços Compostos (Integração API TMDB) .....	5
Instalação e Configuração .....	5
Segurança e Boas Práticas.....	5

# Introdução e Tecnologias

Este documento descreve a arquitetura técnica da plataforma CinePis, desenvolvida para a unidade curricular de Programação e Integração de Serviços. A solução baseia-se numa arquitetura cliente-servidor utilizando tecnologias modernas de desenvolvimento web.

- Runtime: Node.js.
- Framework Web: Express.js.
- IDE: Visual Studio Code
- Motor de Template: EJS (Embedded JavaScript) para renderização dinâmica.
- Base de Dados: MySQL para persistência de dados.
- Comunicação: Formato JSON para troca de mensagens entre cliente e servidor.
- Versionamento: Git (GitHub).

# Estrutura do Projeto (File System)

A organização do código segue o padrão de separação de responsabilidades para garantir a escalabilidade do sistema:

- /controllers: Contém a lógica da API e a exportação dos dados da mesma (ex: mainController.js).
- /public: Ficheiros estáticos (CSS, imagens, JS do cliente).
  - /BD\_Info: Contém o script de criação BD\_Create.txt e a configuração da ligação db.js.
- /routes: Definição dos Web Services REST (utilizadores, conteúdos, listas, géneros).
- /views: Modelos EJS para o Frontoffice e Backoffice.
- .env: Gestão dos dados da Base de dados (credenciais de BD).

# Arquitetura de Dados (MySQL)

A base de dados bd\_projetopis foi desenhada para suportar as entidades principais do sistema:

- utilizadores: Gere contas e permissões (admin/utilizador).
- conteudo: Armazena metadados de filmes e séries importados da API (TMDB ID, título, poster).
- reviews: Regista classificações, críticas e votos de utilidade.
- listas\_personalizadas e lista\_conteudos: Implementam a funcionalidade de coleções de utilizador.
- favoritos e vistos: Tabelas de relação para marcação de conteúdos

# Web Services e Especificações Rest

A plataforma utiliza uma arquitetura baseada em Web Services REST, separando a lógica de rotas por domínios de funcionalidade

## Serviços Elementares (Gestão Interna)

Implementados através de rotas que comunicam diretamente com o MySQL:

- Utilizadores (/auth): Serviços de registo, login e gestão administrativa (Backoffice) de contas.
- Listas e Interação (/api/listas): Endpoints para a criação de listas personalizadas e adição de conteúdos.
- Conteúdos Locais (/api/conteudo): Gestão de reviews e votos de utilidade armazenados localmente.

## Serviços Compostos (Integração API TMDB)

A plataforma consome a API externa do The Movie Database para operações de dados em larga escala:

- Pesquisa e Listagem (/): As rotas principais utilizam serviços compostos que consultam a API externa para apresentar tendências e resultados de pesquisa.
- Metadados Avançados: Importação dinâmica de detalhes como elenco (atores), diretores e géneros cinematográficos diretamente da fonte externa.

<https://developer.themoviedb.org/docs/getting-started>

# Instalação e Configuração

Para replicar o ambiente de desenvolvimento:

1. Clonar o Repositório: `git clone https://github.com/oozie1/Projeto-PIS.git`.
2. Instalar Dependências: Executar `npm install express bcrypt express-session dotenv axios`.
3. Configurar Base de Dados: Executar o script `BD_Create.txt` (localizado em `public/BD_Info/`) no MySQL.
4. Variáveis de Ambiente: Criar um ficheiro `.env` na raiz com:
  - 5. `DB_HOST`, `DB_USER`, `DB_PASS`, `DB_NAME`.
  - 6. `API_KEY` (chave da API TMDB).
7. Execução: `npm start` ou `node index.js`.

# Segurança e Boas Práticas

- Autenticação: Uso de express-session para gestão de sessões de utilizador.
- Criptografia: Palavras-passe armazenadas como hashes utilizando a biblioteca bcrypt.
- Validação: Verificação de privilégios de administrador em rotas críticas (Backoffice).