

Memorial Descritivo do Projeto

API RESTful - Filmes & Premiações + Previsão de Prêmios

Descrição Geral

O projeto tem como objetivo desenvolver um sistema completo para gerenciamento e análise de filmes e premiações, integrando tecnologias modernas e aplicações práticas de Machine Learning. O sistema é composto por:

- Uma API backend desenvolvida em Node.js, utilizando Express e MongoDB.
- Uma API de Machine Learning desenvolvida em Python com FastAPI.
- Um aplicativo mobile criado em Flutter.

O sistema permite o cadastro e autenticação de usuários, listagem e inserção de filmes, registro e consulta de premiações, além da previsão das chances de indicações ou vitórias em premiações como o Oscar e o Globo de Ouro.

Tecnologias Utilizadas

- Node.js + Express
- MongoDB + Mongoose
- Python + FastAPI
- scikit-learn, XGBoost, pandas, joblib
- Flutter (mobile)
- JWT (autenticação)
- Azure VM (deploy)

Orçamento da VM Azure

Para a execução do projeto em ambiente de nuvem, foi utilizado o serviço Microsoft Azure com as seguintes especificações:

- SKU: Standard B1s (1 vCPU, 1 GB RAM)
- Sistema operacional: Ubuntu Server 22.04 LTS
- Disco: 30 GB SSD
- Serviços adicionais: IP público, rede virtual
- Custo estimado mensal: R\$ 75,00 - R\$ 90,00

Total estimado para 3 meses: R\$ 225,00 a R\$ 270,00

Análise de Requisitos

Funcionais:

- RF01: Cadastro e login de usuários com autenticação via JWT.
- RF02: Listagem de todos os filmes cadastrados.
- RF03: Inserção de novos filmes para análise.
- RF04: Listagem de premiações registradas.

- RF05: Previsão de chances de indicação/vitória por meio da API de Machine Learning.
- RF06: Interface mobile para acesso às funcionalidades.

Não Funcionais:

- RNF01: API protegida por autenticação JWT.
- RNF02: Tempo médio de resposta inferior a 2 segundos.
- RNF03: Compatibilidade com dispositivos móveis Android/iOS.
- RNF04: Implantação das APIs em ambiente de nuvem (Azure).
- RNF05: Comunicação segura via HTTPS.

Regras de Negócio:

- RN01: Apenas usuários autenticados podem enviar filmes.
- RN02: O botão de previsão não é exibido para filmes já premiados.
- RN03: Um filme pode receber múltiplas previsões conforme categorias selecionadas.

Justificativa

A proposta deste projeto surge da necessidade de integrar diferentes tecnologias para solucionar um problema prático e realista, unindo o entretenimento (filmes e premiações) com Inteligência Artificial. O desenvolvimento proporciona um ambiente de aprendizagem completo, abordando conceitos de APIs RESTful, autenticação segura, machine learning aplicado, consumo e integração entre serviços, mobile e deploy em nuvem.

Além de agregar ao portfólio acadêmico e profissional dos envolvidos, o projeto utiliza dados públicos e respeita os princípios éticos de desenvolvimento. A utilização da Azure permite simular um ambiente de produção, promovendo contato direto com práticas de DevOps e infraestrutura.

Detalhamento Técnico da API Backend

O backend do sistema é uma API RESTful desenvolvida com Node.js, utilizando o framework Express para gerenciamento das rotas e o banco de dados MongoDB para persistência dos dados. Esta API permite operações de consulta, criação e integração com a API de Machine Learning (FastAPI).

Estrutura de Pastas

```
backend/  
├── controllers/  
├── models/  
├── routes/  
├── middleware/  
├── validators/  
├── docs/  
├── server.js  
└── .env
```

Configuração e Execução

```
npm install
.env com:
MONGO_URI=mongodb://localhost:27017/IMDBest
PORT=5000
npm start
```

Rotas da API

Filmes:

- GET /api/filmes
- POST /api/filmes
- POST /api/filmes/verificar
- POST /api/filmes/classificar

Premiações:

- GET /api/filmes/premiacoes
- POST /api/filmes/premiacoes

Modelos de Dados

Filme: title, year, duration, MPA, rating, votes, meta_score, description, movie_link, writers, directors, stars, languages, awards, createdAt, updatedAt

Premiação: title, year, category, winner

Middlewares

- authMiddleware
- validarFilme
- validarPremiacao

Integração com API FastAPI (ML)

- verificar
- classificar

Tratamento de Erros

- 422: Validação
- 500: Banco de dados
- 400: Requisições malformadas