

## Decide-Aí

O "Decide Ai" é uma plataforma inovadora de recomendação de filmes, projetada para ajudar os usuários a escolherem o filme perfeito com base em suas preferências pessoais. A aplicação funciona de maneira intuitiva e eficiente, sendo composta por três principais componentes: uma interface **frontend**, uma **API Rest** e um **banco de dados**.

<div align="center">



</div>

A interface frontend é onde a mágica começa. É através dela que os usuários interagem diretamente com o sistema. A interface é projetada para ser amigável e de fácil navegação, permitindo que os usuários insiram suas preferências de filmes, como gênero, atores favoritos, diretores, entre outros critérios específicos. A simplicidade e a estética da interface garantem uma experiência agradável e envolvente para o usuário.

Em segundo lugar, temos a API Rest, que serve como o cérebro da operação. Esta API é responsável pela comunicação com várias APIs de filmes disponíveis na internet. Ao receber as preferências do usuário, a API Rest consulta essas fontes externas para coletar informações relevantes sobre os filmes que correspondem aos critérios fornecidos. Além disso, a API implementa a lógica de negócio necessária para filtrar e ordenar as recomendações de maneira eficiente e personalizada, garantindo que as sugestões sejam precisas e relevantes.

Finalmente, é o banco de dados que garante a continuidade e a personalização da experiência do usuário. Nele, são armazenadas informações importantes, como o histórico de preferências ou as

interações anteriores dos usuários com a plataforma. Essa é uma forma de o "Decide Ai" aprender com o comportamento do usuário ao longo do tempo para uma melhoria contínua na qualidade das recomendações oferecidas.

Em resumo, o "Decide Ai" combina uma interface de usuário intuitiva, uma API Rest robusta e um banco de dados eficiente para fornecer recomendações de filmes personalizadas. Essa estrutura integrada não apenas facilita a escolha de filmes, mas também enriquece a experiência do usuário ao proporcionar sugestões que realmente correspondem aos seus gostos e preferências individuais.

## Integrantes

\* Pedro Gonçalves dos Santos - RA: 794042

\* Gabriel Orlando - RA: 790728

## Tecnologias utilizadas

### Frontend

A interface frontend é desenvolvida utilizando o React, uma biblioteca JavaScript popular e poderosa criada pelo Facebook. O React é conhecido por sua eficiência e flexibilidade, o que o torna uma escolha ideal para construir interfaces de usuário dinâmicas e responsivas. Ao utilizar o React podemos obter: Componentização, Facilidade da manutenção e um desenvolvimento mais ágil

<div align="center">



</div>

### Backend

No backend , utilizamos o **Node.js**, uma plataforma construída sobre o motor JavaScript V8 do Google Chrome. Node.js é uma escolha excelente para o desenvolvimento de aplicações web modernas devido à sua eficiência, escalabilidade e ampla adoção na comunidade de desenvolvimento. Além disso, é possível observar alguns pontos principais da utilização do Node JS: **Desempenho e eficiência e Escalabilidade**

<div align="center">



</div>

### **Banco de dados**

Para o armazenamento de dados no "Decide Ai", optamos por utilizar o PostgreSQL, um sistema de gerenciamento de banco de dados relacional poderoso e avançado. O PostgreSQL é muito utilizado por sua robustez, conformidade com padrões e vasta gama de funcionalidades, tornando-o uma escolha ideal para nossas necessidades.

<div align="center">



</div>

### **Manual de instalação**

Para a utilização da aplicação são necessários alguns requisitos:

- Docker
- Docker compose
- Kubernetes/Minikube

Guia de instalação:

- <https://docs.docker.com/compose/install/>
- <https://docs.docker.com/get-docker/>

- <https://minikube.sigs.k8s.io/docs/>

Para executar a aplicação com kubernetes, é necessário apenas executar o arquivo `minikube-up.sh`, e caso ele não esteja como executável é necessário executar previamente o seguinte comando

```
sudo chmod +x minikube-up.sh
```

Esse script já faz toda a configuração para montar um container com minikube, buildar as imagens dos sistemas e construir os serviços a partir dos arquivos .yaml

Após a execução do script, basta acessar a aplicação via kubernetes no endereço `localhost:8080`. Esse endereço foi disponibilizado através do comando `kubectl port-forward`, no final do script.

A configuração via ingress foi feita e é executada no script, entretanto estava inacessível na máquina da dupla, portanto, após alinhamento com o professor, foi disponibilizado também o acesso usando o `kubectl port-forward`. Caso queira testar o acesso via ingress, basta abrir o arquivo `/etc/hosts` e configurar com o ip fornecido pelo comando `minikube ip` com o endereço host `decideai.k8s.local` e depois acessar esse mesmo endereço host em algum navegador. Exemplo:

```
192.168.49.2 decideai.k8s.local
```

Utilize as seguintes credenciais de teste para login:

- Email: `pedro@gmail.com`
- Senha: `123456`

Para ativar as funcionalidades completas do sistema, é necessário usar a API de OMDb, basta criar uma API key no <https://www.omdbapi.com/> e alterar a Url no arquivo `/src/pages/SugerirFilmes/index.js`

Para a matéria de DevOps, basta apenas o login estar funcionando, demonstrando a conexão entre os containers da aplicação pelo kubernetes.

É possível também utilizar a aplicação somente via docker, rodando o comando `docker compose up -d --build` na raiz do projeto.