

App de Caronas – Motoristas e Caroneiros

Estrutura do Projeto

A estrutura do projeto está organizada da seguinte forma:

```
/
├── database-structure.sql  # Estrutura do banco de dados
├── front                  # Aplicação frontend (React + Vite + TypeScript)
└── back                  # API backend (Python)
```

- O arquivo `database-structure.sql` contém os scripts SQL necessários para a criação da estrutura do banco de dados.
- A pasta `front` contém o código da aplicação frontend.
- A pasta `back` contém o código da API backend.

O frontend e o backend são projetos separados, mas interdependentes, ou seja, ambos devem estar em execução para o funcionamento completo do sistema.

Frontend – React + Vite + TypeScript

Localizado na pasta `/front`, o frontend é desenvolvido com as seguintes tecnologias:

- React
- Vite
- TypeScript
- shadcn/ui (biblioteca de componentes)
- pnpm (gerenciador de pacotes)

Funcionalidades principais:

- Registro e autenticação de usuários (motoristas e caroneiros);
- Seleção de origem e destino usando a API do Google Maps;
- Exibição de motoristas disponíveis com base na rota do caroneiro;
- Redirecionamento para um link de pagamento via Mercado Pago após a escolha da carona.

Backend – API em Python

Localizado na pasta `/back`, o backend é uma API construída em Python. Suas principais responsabilidades são:

- Gerenciar o cadastro e autenticação de usuários;
- Armazenar informações de rotas, pontos de origem/destino e valores;
- Persistir os dados necessários para as integrações com o Google Maps e o Mercado Pago.

Observação: o backend não se conecta diretamente com a API do Google Maps nem com o Mercado Pago. Ele apenas armazena os dados necessários para que o frontend realize essas integrações.

As dependências estão listadas no arquivo `requirements.txt` localizado na pasta `/back`.

Como Executar o Projeto

Backend (Python)

```
cd back
pip install -r requirements.txt
```

Não é necessário ativar um ambiente virtual com `venv\\Scripts\\activate`, mas recomenda-se o uso de ambientes isolados.

Frontend (React)

```
cd front  
pnpm install  
pnpm dev
```

APIs Utilizadas

- **Google Maps API:** usada no frontend para geolocalização, escolha de origem e destino, e cálculo de rotas.
- **Mercado Pago API:** usada no frontend para gerar links de pagamento após a seleção de uma rota.

Observações

- O frontend e o backend são separados, mas precisam estar ativos simultaneamente.
- O backend não consome APIs externas diretamente — apenas fornece os dados necessários.
- É recomendada a configuração de variáveis de ambiente usando arquivos `.env` para dados sensíveis como tokens e chaves.

Objetivo do Projeto

Este projeto tem como finalidade conectar motoristas e caroneiros que realizam trajetos semelhantes. A ideia é promover a economia e a praticidade por meio do compartilhamento de caronas, otimizando o uso de veículos e reduzindo custos.

Licença

Este projeto é livre para fins educacionais e pode ser utilizado e adaptado conforme necessário.