

# DOCUMENTAÇÃO DO PROCESSO – TESTE TÉCNICO

Gabrielle Lorryne Fernandes Nascimento

Objetivo: registrar o passo a passo do desenvolvimento, execução e testes da API, com evidências (prints).

## 1. Descrição do Projeto

Foi desenvolvida uma API REST para controle de clientes e suas respectivas placas de veículos, conforme solicitado no teste técnico.

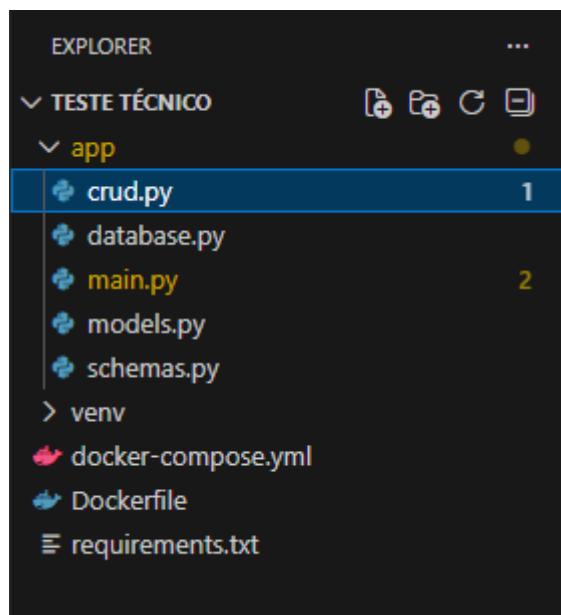
A API permite:

- Cadastro de clientes
- Consulta de clientes
- Atualização de dados
- Remoção de clientes
- Consulta por final da placa do carro

## 2. Tecnologias Utilizadas

- Python 3.11
- FastAPI
- SQLAlchemy
- PostgreSQL
- Docker e Docker Compose
- Swagger (OpenAPI) para documentação e testes
- Git e GitHub para versionamento

## 3. Estrutura do Projeto



#### 4. Configuração do Ambiente com Docker

Foi utilizado Docker Compose para subir:

- Um container da API (FastAPI)
- Um container do banco de dados (PostgreSQL)

Comando utilizado:

**docker compose up --build**

	Name	Container ID	Image	Port(s)	CPU (%)	Actions
	testetcnico	-	-	-	0.1%	
	db-1	daa46b67774b	postgres:1	-	0%	
	api-1	fbb1ea0228b2	testetcnico	8000:8000	0.1%	

#### 5. Endpoints Implementados

default	
<code>GET</code>	/clientes/ Listar Clientes
<code>POST</code>	/clientes/ Criar Cliente
<code>GET</code>	/clientes/{cliente_id} Obter Cliente
<code>DELETE</code>	/clientes/{cliente_id} Remover Cliente
<code>PUT</code>	/clientes/{cliente_id} Atualizar Cliente
<code>GET</code>	/consulta/final-placa/{numero} Consultar Por Final Placa

#### 7. Testes com Dados Reais

Todos os endpoints foram testados manualmente via Swagger.

Exemplo de cadastro de cliente

```
{  
  "nome": "Gabrielle Fernandes",  
  "telefone": "31997415117",  
  "cpf": "12345678900",  
  "placa_carro": "ABCD123"  
}
```

**POST** /clientes/ Criar Cliente

Parameters

No parameters

Request body required

application/json

```
{ "nome": "Gabrielle Fernandes", "telefone": "31997415117", "cpf": "12345678900", "placa_carro": "ABC1023" }
```

Execute Clear

Responses

Curl

```
curl -X 'POST' \
'http://localhost:8000/clientes/' \
-H 'accept: application/json' \
-H 'Content-Type: application/json' \
-d '{
  "nome": "Gabrielle Fernandes",
  "telefone": "31997415117",
  "cpf": "12345678900",
  "placa_carro": "ABC1023"
}'
```

## 7. Teste do Filtro por Final da Placa

Endpoint:

GET /consulta/final-placa/{numero}

Exemplo:

/consulta/final-placa/1234

Retorna apenas clientes cuja placa termina com o número informado.

**GET** /consulta/final-placa/{numero} Consultar Por Final Placa

Parameters

Name Description

numero required string (path) 1234

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'http://localhost:8000/consulta/final-placa/1234' \
-H 'accept: application/json'
```

Request URL

```
http://localhost:8000/consulta/final-placa/1234
```

## 8. Testes de Atualização e Remoção

- PUT para atualização dos dados
- DELETE para remoção do cliente
- Teste posterior confirmando remoção

## 9. Versionamento

O código foi versionado no GitHub, com commits separados por funcionalidade.

Name	Last commit message	Last commit date
app	feat: API REST Controle Carros	37 minutes ago
Dockerfile	feat: API REST Controle Carros	37 minutes ago
docker-compose.yml	feat: API REST Controle Carros	37 minutes ago
requirements.txt	feat: API REST Controle Carros	37 minutes ago