

Documentação referente ao Projeto solicitado como Desafio

Nivaldo Lucas

Visão Geral:

O projeto Gerenciador de Clientes é uma aplicação Laravel que gerencia clientes de forma eficiente. Ele oferece funcionalidades de CRUD (Create, Read, Update, Delete) para clientes, incluindo campos como NOME, DATA DE NASCIMENTO, CPF OU CNPJ, FOTO e NOME SOCIAL.

Requisitos:

PHP >= 7.3

Laravel >= 7

Banco de dados **MySQL**

Funcionalidades Principais:

Listagem de Clientes:

Página de listagem de todos os clientes cadastrados.

Cadastro de Clientes:

Formulário para cadastrar novos clientes.

Atualização de Clientes:

Página e formulário para editar informações de um cliente.

Visualização Individual de Clientes:

Página detalhada com as informações de um cliente específico.

Instalação:

Clone o repositório. (https://github.com/magrelinho4512/Gerenciador_de_Clientes)

(Deixei um arquivo chamado **clientespbsoft.sql** com as informações do banco de dados)

Execute **composer install** para instalar as dependências.

Copie **.env.example** para **.env** e configure as variáveis de ambiente.

Execute **php artisan key:generate** para gerar a chave de aplicativo.

Configure o banco de dados no arquivo **.env**

Execute **php artisan migrate** para criar as tabelas do banco de dados.

Configuração:

As configurações principais estão no arquivo **.env**

Configure o arquivo **config/database.php** para personalizar as conexões de banco de dados.

Estrutura do Projeto:

```
/app
  /Http
    /Controllers
      ClienteController.php
    /Models
      Cliente.php

/resources
  /views
    /clientes
      create.blade.php
      edit.blade.php
      index.blade.php
      show.blade.php
    /Layouts
      app.blade.php

/database
  /migrations
    /create_clientes_table.php

/routes
  /web.php

/public
  /css
    style.css
  /images

.env
```

Arquitetura e Design:

O projeto segue a arquitetura **MVC** padrão do Laravel. Os controllers estão localizados em **app/Http/Controllers**, e as views em **resources/views/clientes**. O código faz uso extensivo do **Eloquent ORM** para interação com o banco de dados.

Banco de Dados:

O modelo de dados inclui a tabela **clientes** com os campos **id**, **nome**, **data_nascimento**, **cpf_cnpj**, **foto** e **nome_social**.

Relacionamentos e chaves estrangeiras, se aplicáveis, são definidos no próprio **Eloquent**.