

# Documentação do Projeto TesteDefensoria

## Requisitos

- *Python 3*, de preferência 3.9 e seu gerenciador de pacotes *pip*
- Sistema Operacional Windows
- Banco de Dados PostgreSQL

## Configuração

- Assumindo que o banco de dados e o Python estão instalados no seu computador, crie um banco de dados no servidor do Postgres com o nome `db.people` (se usar outro nome basta alterar no arquivo `settings`)
- Agora, com o banco de dados criado vá na pasta `\core`, procure o arquivo `settings.py` e na **linha 84** coloque a senha do seu banco de dados.
- Para instalar as dependências do programa basta rodar o script `setup.bat` que está na raiz do projeto. Ele irá criar um ambiente virtual(pasta `\venv`), instalará as dependências do programa e o executará.

Caso preferir, ou se houver algum problema na execução do *script*, você pode abri-lo, copiar os comandos e executar manualmente.

- O site estará funcionando na url `http://localhost:8000`

## Descrição do Projeto

A estrutura do projeto foi gerada de acordo com o *template* do *Famework Django*.

- `core\` : Pasta onde estão os módulos de configuração do projeto
- `core\settings.py` : onde estão as principais configurações do projeto

- `core\urls.py` : Onde estão mapeadas as rotas do site.
- `core\asgi.py` : Onde estão as configuração para rodar no servidor em produção Assíncrono.
- `core\wsgi.py` : Onde estão as configuração para rodar no servidor em produção Síncrono.
- `peopleform\` : Pasta de aplicação criada pelo framework, onde o código principal de desenvolvimento do programa estará.
- `peopleform\models.py` : Módulo onde estão os modelos que serão mapeados na tabela do banco de dados pelo ORM do Django.
- `peopleform\urls.py` : Módulo que possui as rotas das paginas desta aplicação.
- `peopleform\views.py` : Módulo que contem as classes de visualização. Essas classes mapearão o modelo para um formulário *html*, facilitando a comunicação entre os dados e a páginas *html*.

Os outros módulos da pasta `peopleform\` não foram usados no desenvolvimento desta aplicação.

- `static\` : Pasta que possui os arquivos `.css` que geram os estilos para as páginas *html*.
- `template\` Pasta que contém os *templates* `.html` utilizados na aplicação.

Os principais códigos desenvolvidos estão nos módulos: `models.py`, `urls.py`, `views.py` e nos `html's` da pasta `templates` e estão todos comentados.

Autor : Janilson Serra