

# Documentação Técnica - Church Members

(Aplicação de Gerenciamento de Membros da Igreja)

*Feito por: Maykon Heleno Máximo*

*Supervisão e Orientação: Prof Emeson Borges*

## **- Visão Geral**

Este projeto é uma aplicação web desenvolvida em Python utilizando o framework Streamlit, destinada ao gerenciamento de membros da igreja. Ela permite que os administradores consultem, adicionem, editem e excluam dados dos membros de forma segura. A aplicação inclui uma interface para autenticação de usuário, que restringe o acesso a páginas específicas, como a de consulta de dados dos membros.

## **- História**

O desenvolvimento da aplicação surgiu com a necessidade dos líderes da igreja terem as informações dos membros para realizar algumas atividades, como: envio de cartas, comemoração dos aniversariantes do mês, organização, controle de membros e ministérios (departamentos).

Partindo da necessidade da igreja e do pedido do professor Emeson Borges de criar uma aplicação web com CRUD (Create, Read, Update, Delete), surgiu a ideia da concepção do Church Members.

## **- Requisitos**

Para executar a aplicação, os seguintes requisitos são necessários:

- Python 3.7 ou superior
- Bibliotecas Python: Streamlit, Pandas, Requests
- PostgreSQL

## **- Instalação**

1. Entre no repositório da aplicação no Github pelo [link](#);

2. Clone o Repositório:

```
git clone <URL_DO_REPOSITORIO>  
cd <PASTA_DO_PROJETO>
```

3. Crie um Ambiente Virtual (opcional, apenas recomendação):\*\*

```
python -m venv venv  
source venv/bin/activate (Linux/Mac)  
venv\Scripts\activate (Windows)
```

4. Instale as Dependências:

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

5. Execute a Aplicação:

```
streamlit run main.py
```

Ou acesse a aplicação pelo navegador em <http://localhost:8501>

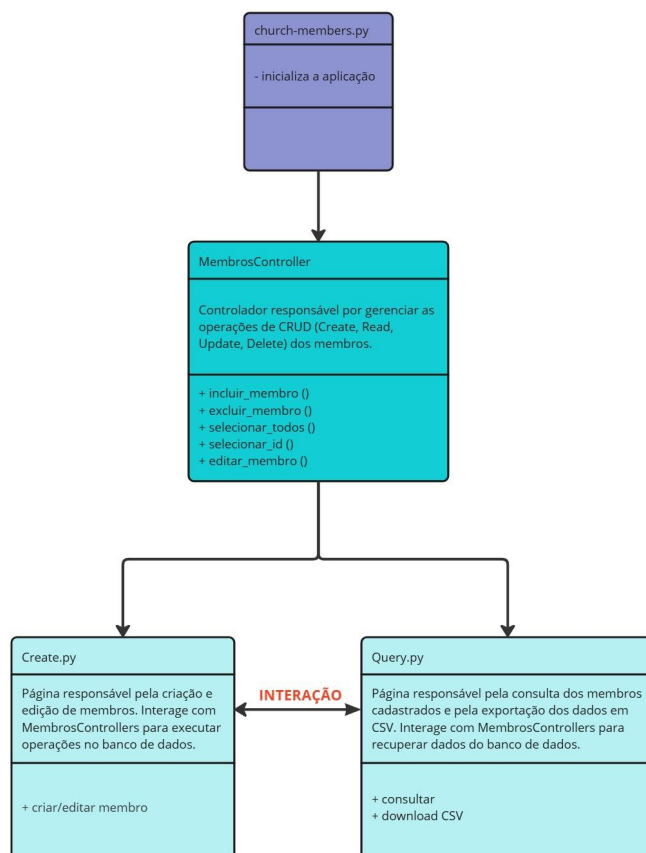
*(POSTERIOMENTE SERÁ FEITO DEPLOY EM SERVIDOR)*

## - Estrutura do Projeto

Abaixo está uma estrutura simplificada dos arquivos do projeto:

PROJETO/

- |
  - |— Controllers/
    - |— MembrosControllers.py (*Controlador de membros para comunicação com a camada de dados*)
  - |— Pages/
    - |— Clientes/
      - |— Create.py (*Página de criação/edição de membros*)
      - |— query.py (*Página de consulta dos membros*)
  - |— main.py (*Arquivo principal para execução da aplicação*)
  - |— requirements.txt (*Arquivo com dependências do projeto*)
  - |— README.md (*Documentação geral do projeto*)



## **- Descrição dos Módulos**

(main.py)

O arquivo principal que inicializa a aplicação e controla o roteamento entre diferentes páginas.

(Controllers/MembrosControllers.py)

Controlador responsável pelas operações CRUD (Criar, Ler, Atualizar, Excluir) relacionadas aos membros., incluindo funções como:

- `incluir_membro()`: Insere um novo membro no banco de dados.
- `selecionar_todos()`: Retorna uma lista de todos os membros.
- `selecionar_id (id)`: Retorna um membro específico pelo seu ID.
- `excluir_membro(id)`: Exclui um membro pelo seu ID.

(Pages/Clientes/Create.py)

Este módulo define a página de criação/edição de membros. Ele inclui um formulário para preencher os dados do membro e lida com as operações de validação e salvamento.

(Pages/query.py)

Este módulo contém a página de consulta dos membros. Ele exibe uma lista de todos os membros cadastrados e permite que o administrador visualize detalhes, edite e exclua membros específicos.

## **- Autenticação**

A autenticação é implementada no Pages/query.py para restringir o acesso às páginas de consulta. Para acessar as páginas restritas, o usuário precisa fornecer um nome de usuário e senha. As credenciais são definidas diretamente no código para simplificar a implementação, mas é recomendável usar variáveis de ambiente em produção.

### **- Exemplo de Autenticação:**

```
# Configuração de credenciais

USERNAME = "admin"

PASSWORD = "senha123" # Trocar pela senha desejada
```

## ***- Funcionalidades da Aplicação***

### **1. Cadastro de Membros**

Permite inserir novos membros no sistema com informações como nome, idade, data de nascimento, telefone, endereço, estado civil e ministério.

### **2. Consulta de Membros**

Exibe uma lista de todos os membros cadastrados, com opção para download dos dados em CSV. Os administradores podem:

- Visualizar detalhes individuais de cada membro.
- Editar informações de um membro específico.
- Excluir membros do banco de dados.

### **3. Autocompletar Endereço**

Ao inserir um CEP válido, o sistema faz uma consulta à API do ViaCEP para autocompletar os campos de endereço.

## ***- Segurança e Privacidade***

1. **Autenticação de Usuário:** Apenas usuários autenticados têm acesso à página de consulta.
2. **Proteção de Dados:** Recomenda-se que dados sensíveis, como credenciais de login, sejam armazenados em variáveis de ambiente em vez de diretamente no código.
3. **API do ViaCEP:** Os dados de CEP são consultados de forma segura pela API do ViaCEP. Não são mantidos registros das consultas feitas à API.

## ***- Futuro da Aplicação***

Como dito anteriormente, a aplicação foi feita por causa da necessidade dos líderes da igreja terem as informações dos membros para realizar algumas atividades e pelo pedido do professor Emeson Borges de criar uma aplicação web com CRUD. Porém, existe projetos de expansão da aplicação, como: Página exclusiva de cada ministério para gerenciamento de escalas, página para inscrições de eventos, página de anúncios, e entre outros.

## ***- Referências e Recursos Adicionais***

- Documentação do Streamlit
- Documentação do PostgreSQL