

Gerenciador de Clientes - Documentação

Bem-vindo ao **Gerenciador de Clientes**, uma aplicação simples em Python para gerenciar informações de clientes (nome, sobrenome, e-mail e CPF) com um banco de dados SQLite. A aplicação foi desenvolvida com **Python**, **SQL** e **SQLite**, e possui uma interface gráfica criada com **Tkinter**. Este projeto é ideal para aprender a criar uma aplicação com interface gráfica, conectando-a a um banco de dados e realizando operações CRUD (adicionar, visualizar, buscar, atualizar e deletar registros).

Funcionalidades da Aplicação

- **Adicionar novos clientes** ao banco de dados.
- **Visualizar todos os clientes** em uma tabela.
- **Buscar clientes** por nome, sobrenome, e-mail ou CPF.
- **Atualizar os dados** de um cliente existente.
- **Deletar clientes** do banco de dados.

A interface gráfica é intuitiva, com campos para inserir dados e botões para executar as ações. O banco de dados SQLite armazena as informações no arquivo `clientes.db`.

Estrutura do Projeto

O código é organizado em três arquivos Python:

1. **Gui.py:**
 - Cria a interface gráfica usando Tkinter.
 - Contém campos de entrada e botões para as ações (Adicionar, Atualizar, Deletar, Buscar, Limpar).
 - Conecta-se à classe **Backend** para realizar as operações no banco de dados.
2. **Backend.py:**
 - Gerencia todas as operações do banco de dados SQLite.
 - Inclui métodos para conectar ao banco, criar a tabela, inserir, visualizar, buscar, atualizar e deletar clientes.
 - Utiliza consultas SQL seguras para evitar vulnerabilidades.
3. **application.py:**
 - Arquivo principal que inicializa o banco de dados e abre a interface gráfica.

Pré-requisitos

Antes de rodar a aplicação, é necessário ter:

- **Python 3.x** (Recomendado: Python 3.8 ou superior) – [Download Python](#).
- **Tkinter** (Já incluído no Python, sem necessidade de instalação adicional).
- **SQLite** (Módulo `sqlite3`, já incluso no Python).
- **PyInstaller** (Opcional) – Para criar um arquivo executável. Veja "Criando um Executável" abaixo.

Como Configurar e Executar

1. Baixar ou criar os arquivos:

- Crie uma pasta para o projeto (exemplo: gerenciador_clientes).
- Salve os três arquivos (Gui.py, Backend.py, application.py) nesta pasta.

2. Verificar se o Python está instalado:

- Abra o terminal (Prompt de Comando no Windows, Terminal no Linux/Mac) e digite:

```
bash
CopiarEditar
python --version
```

- Se o Python estiver instalado, a versão será exibida. Caso contrário, baixe e instale o Python.

3. Executar a aplicação:

- No terminal, navegue até a pasta do projeto:

```
bash
CopiarEditar
cd caminho/para/gerenciador_clientes
```

- Execute o arquivo principal:

```
bash
CopiarEditar
python application.py
```

- A janela gráfica será aberta, mostrando a interface do Gerenciador de Clientes.

Como Usar a Aplicação

Ao executar `application.py`, a janela gráfica será exibida com:

- **Campos de entrada:** Para Nome, Sobrenome, E-mail e CPF.
- **Botões:**
 - **Adicionar:** Insere um novo cliente.
 - **Atualizar:** Atualiza o cliente selecionado.
 - **Deletar:** Remove o cliente selecionado.
 - **Buscar:** Filtra os clientes com base nos dados inseridos.
 - **Limpar:** Limpa os campos de entrada.
- **Tabela (Treeview):** Exibe todos os clientes do banco de dados.

Passo a Passo:

1. Adicionar um cliente:

- Preencha os campos (Nome, Sobrenome, E-mail, CPF).
- Clique em "Adicionar".

- Uma mensagem de sucesso será exibida e a tabela será atualizada.
- 2. **Visualizar clientes:**
 - A tabela mostra todos os clientes ao abrir a aplicação.
- 3. **Buscar clientes:**
 - Insira um valor em qualquer campo (ex: Nome ou CPF) e clique em "Buscar".
 - A tabela será filtrada com os resultados.
- 4. **Atualizar um cliente:**
 - Selecione um cliente na tabela (os campos serão preenchidos automaticamente).
 - Edite os campos desejados e clique em "Atualizar".
 - A tabela será atualizada com as novas informações.
- 5. **Deletar um cliente:**
 - Selecione um cliente na tabela e clique em "Deletar".
 - O cliente será removido e a tabela será atualizada.

Criando um Executável com PyInstaller

Para rodar a aplicação sem a necessidade do Python, você pode gerar um executável (.exe no Windows ou equivalente) com **PyInstaller**.

Passos para criar o executável:

1. Instalar o PyInstaller:

- No terminal, execute:

```
bash
CopiarEditar
pip install pyinstaller
```

2. Criar o executável:

- Navegue até a pasta do projeto:

```
bash
CopiarEditar
cd caminho/para/gerenciador_clientes
```

- Execute o comando para criar o executável:

```
bash
CopiarEditar
pyinstaller --onefile application.py
```

3. Explicação:

- `--onefile`: Cria um único arquivo executável.
- O PyInstaller cria uma pasta `dist` contendo o arquivo executável (ex: `application.exe`).

4. Executar o executável:

- Vá até a pasta `dist` e execute o arquivo `application.exe`.

O executável pode ser grande (50-100 MB), pois inclui o Python e todas as dependências.

5. Compartilhando o Executável:

- Para compartilhar, copie o arquivo `.exe` e o banco de dados (`clientes.db`) para outros computadores.

Estrutura do Banco de Dados

O banco de dados SQLite é armazenado em um arquivo chamado `clientes.db` na mesma pasta do projeto. A tabela `clientes` possui os seguintes campos:

- **id:** Identificador único (auto-incrementado).
- **nome:** Nome do cliente (texto).
- **sobrenome:** Sobrenome do cliente (texto).
- **email:** E-mail do cliente (texto).
- **cpf:** CPF do cliente (texto).

A tabela é criada automaticamente na primeira execução da aplicação.

Dicas para Solução de Problemas

- **Erro: "No module named tkinter":**
 - Certifique-se de que o Python está instalado corretamente. Tkinter vem com o Python, mas pode estar faltando em algumas instalações. Instale Tkinter com:

```
bash
CopiarEditar
pip install tk
```

- **Erro ao executar a aplicação:**
 - Verifique se os arquivos (`Gui.py`, `Backend.py`, `application.py`) estão na mesma pasta.
 - Confirme que está executando o comando na pasta correta.
- **O executável não abre:**
 - Verifique se o PyInstaller foi executado com sucesso.
 - No Windows, tente executar o `.exe` pelo terminal para ver mensagens de erro:

```
bash
CopiarEditar
./dist/application.exe
```

- **Tabela não atualiza:**
 - Verifique se o arquivo `clientes.db` está na mesma pasta do executável ou dos arquivos Python.

O que você pode aprender com este projeto

- **Python:** Organizar código em classes e módulos, além de trabalhar com métodos estáticos.
- **SQL e SQLite:** Criar e manipular tabelas e dados com SQL seguro.
- **Tkinter:** Criar interfaces gráficas com campos de texto, botões e tabelas.
- **Boas práticas:** Uso de consultas parametrizadas, separação de responsabilidades e comentários claros.
- **PyInstaller:** Transformar um programa Python em um executável para fácil distribuição.

Próximos Passos

- **Validação de CPF:** Adicionar validação para garantir o formato correto.
- **Exportação para CSV:** Permitir exportar a lista de clientes para um arquivo CSV.
- **Recarregar a tabela:** Adicionar um botão para recarregar a lista de clientes após uma busca.
- **Melhorar a interface:** Adicionar cores, ícones ou melhorias no layout.