



Gerenciador de Clientes — Documentação

Bem-vindo ao Gerenciador de Clientes!

Esta é uma aplicação simples desenvolvida em Python que permite gerenciar informações de clientes — como nome, sobrenome, e-mail e CPF — utilizando um banco de dados SQLite e uma interface gráfica desenvolvida com Tkinter.

Ideal para iniciantes, este projeto ajuda a entender como integrar Python, SQL e interfaces gráficas em uma aplicação funcional, com as operações básicas de um CRUD (Create, Read, Update, Delete).



O que esta aplicação faz?

A aplicação permite:

- 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 é intuitiva, com campos de entrada para dados e botões para cada ação. As informações são armazenadas localmente em um arquivo SQLite chamado `clientes.db`.



Estrutura do Projeto

O projeto é dividido em três arquivos Python para manter o código modular e organizado:

1. `Gui.py`

- Contém a classe `Gui`, responsável por criar a interface gráfica com Tkinter.
- Inclui campos de texto, botões (Adicionar, Atualizar, Deletar, Buscar, Limpar) e uma tabela (`Treeview`) para exibir os clientes.
- Conecta-se ao `Backend` para executar operações no banco de dados.

2. `Backend.py`

- Contém a classe `Backend`, responsável pela lógica de acesso ao banco de dados SQLite.

- Inclui métodos para:
 - Criar e conectar ao banco de dados
 - Criar a tabela
 - Inserir, consultar, buscar, atualizar e deletar registros
- Utiliza consultas SQL parametrizadas, seguras contra injeção.

3. `application.py`

- Arquivo principal do projeto.
- Inicializa o banco de dados e executa a interface gráfica.

Pré-requisitos

Para executar este projeto, você precisa de:

- **Python 3.x** (recomendado: Python 3.8 ou superior)
[Download em python.org](https://www.python.org/downloads/)
- **Tkinter** — já incluído no Python padrão.
- **SQLite** — incluído no Python como o módulo `sqlite3`.
- **PyInstaller** (*opcional*) — para criar um executável `.exe`.

Como configurar e executar

1. Baixar ou criar os arquivos

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

2. Verificar se o Python está instalado

Abra o terminal ou prompt de comando e digite:

```
python --version
```

Se o Python estiver instalado corretamente, será exibida a versão.

3. Executar a aplicação

No terminal, navegue até a pasta do projeto:

```
cd caminho/para/gerenciador_clientes
python application.py
```

A interface gráfica será aberta.

Como usar a aplicação

A janela principal inclui:

- **Campos de entrada:** Nome, Sobrenome, Email, CPF
- **Botões:**
 - **Adicionar:** Insere novo cliente
 - **Atualizar:** Altera os dados do cliente selecionado
 - **Deletar:** Remove cliente selecionado
 - **Buscar:** Filtra clientes pelos campos preenchidos
 - **Limpar:** Limpa os campos de entrada
- **Tabela (Treeview):** Mostra todos os clientes do banco

Passo a passo:

Adicionar um cliente

1. Preencha os campos.
2. Clique em **Adicionar**.
3. O cliente será salvo e a tabela atualizada.

Buscar clientes

1. Digite um valor em qualquer campo.
2. Clique em **Buscar**.
3. A tabela mostrará apenas os resultados correspondentes.

Atualizar cliente

1. Clique em um cliente da tabela.
2. Edite os campos.
3. Clique em **Atualizar**.

Deletar cliente

1. Selecione um cliente na tabela.
2. Clique em **Deletar**.

Limpar campos

Clique em **Limpar** para esvaziar todos os campos.

Criando um executável com PyInstaller (opcional)

1. Instalar o PyInstaller


```
pip install pyinstaller
```

2. Gerar o executável

No terminal, na pasta do projeto:

```
pyinstaller --onefile application.py
```

O executável será criado na pasta `dist/`.

 **Dica:** O executável pode ficar grande (50–100 MB) porque inclui o interpretador Python e dependências.

Estrutura do banco de dados

O banco é salvo no arquivo `clientes.db` e contém uma única tabela: `clientes`.

Tabela: `clientes`

Campo	Tipo	Descrição
<code>id</code>	INTEGER	Chave primária, gerada automaticamente
<code>nome</code>	TEXT	Nome do cliente
<code>sobrenome</code>	TEXT	Sobrenome do cliente
<code>email</code>	TEXT	Endereço de e-mail
<code>cpf</code>	TEXT	CPF (armazenado como texto)

A tabela é criada automaticamente na primeira execução por `Backend.initDB()`.

Dicas para solução de problemas

- **Erro: No module named tkinter**
Verifique se o Python foi instalado corretamente. Caso necessário, instale o Tkinter com:

```
pip install tk
```

- **Erro ao executar o programa**

Certifique-se de que os arquivos `Gui.py`, `Backend.py` e `application.py` estão na mesma pasta.

- **O executável não abre**

Rode o `.exe` pelo terminal para ver mensagens de erro:

```
./dist/application.exe
```

- **Tabela não atualiza**

Confirme se o arquivo `clientes.db` está no mesmo diretório do executável.



Para alunos: o que você aprende com este projeto?

- **Python:** Estrutura de classes, modularização, boas práticas.
- **SQL / SQLite:** Criação de tabelas, comandos `INSERT`, `SELECT`, `UPDATE`, `DELETE`.
- **Tkinter:** Criação de interfaces gráficas simples e funcionais.
- **Segurança:** Uso de consultas SQL parametrizadas.
- **Distribuição:** Como empacotar seu projeto com PyInstaller.



Próximos passos (para prática e evolução)

Você pode melhorar esta aplicação com:

- Validação do CPF (formato e número).
- Exportar os dados para um arquivo `.csv`.
- Adicionar botão para recarregar todos os clientes.
- Melhorar a interface com cores, ícones ou menus.
- Adicionar campo de telefone ou data de nascimento.