

# Documentação da Aplicação: Gerenciador de Clientes

## Descrição Geral

O Gerenciador de Clientes é uma aplicação simples desenvolvida em Python. Seu objetivo é permitir o controle de informações de clientes, como nome, sobrenome, e-mail e CPF, utilizando um banco de dados local chamado SQLite. A interface gráfica é feita com a biblioteca Tkinter, tornando o uso fácil e visual.

Essa aplicação é excelente para quem quer aprender como integrar uma interface gráfica com um banco de dados e realizar operações como adicionar, visualizar, buscar, atualizar e excluir dados.

## O que a aplicação faz

Com o Gerenciador de Clientes, você pode:

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

Tudo isso é feito por meio de uma janela com campos de preenchimento e botões de ação. Os dados são salvos em um arquivo chamado clientes.db.

## Como o projeto está organizado

O projeto é dividido em três arquivos principais, todos escritos em Python. Isso ajuda a manter o código organizado e facilita o entendimento:

1. Gui.py
  - Responsável pela criação da interface gráfica com Tkinter.
  - Possui os campos para digitar os dados dos clientes, botões para executar as ações e uma tabela para exibir os dados.
  - Se comunica com o código do arquivo Backend.py para acessar o banco de dados.
2. Backend.py
  - Contém os comandos que lidam diretamente com o banco de dados SQLite.
  - Aqui estão os métodos para criar a tabela, inserir dados, buscar clientes, atualizar e deletar registros.
  - Utiliza comandos SQL seguros, protegendo contra falhas ou ataques.

### 3. application.py

- É o arquivo principal da aplicação.
- Inicia o banco de dados e abre a interface gráfica.

O que você precisa para rodar o projeto

Antes de começar, verifique se você tem os seguintes itens:

- Python 3 (recomendado: versão 3.8 ou superior). Pode ser baixado do site oficial: [python.org](https://python.org).
- Tkinter já vem instalado com o Python na maioria dos casos.
- SQLite também está incluído com o Python (módulo sqlite3).
- PyInstaller (opcional): só é necessário se quiser transformar o programa em um executável (.exe).

Como configurar e executar o programa

Siga este passo a passo:

#### 1. Organize os arquivos

- Crie uma pasta para o projeto (exemplo: gerenciador\_clientes).
- Salve dentro dessa pasta os três arquivos: Gui.py, Backend.py e application.py.

#### 2. Verifique se o Python está instalado

- Abra o terminal ou prompt de comando.
- Digite: `python --version`
- Se o Python estiver instalado corretamente, ele mostrará a versão. Se não, instale o Python.

#### 3. Execute a aplicação

- No terminal, vá até a pasta onde estão os arquivos:
- Exemplo: `cd caminho/para/gerenciador_clientes`
- Execute o programa com:
- `python application.py`
- A janela do Gerenciador de Clientes será aberta.

## Como usar a aplicação

Ao abrir a janela, você verá:

- Campos para preenchimento: Nome, Sobrenome, E-mail e CPF.
- Botões de ação:
  - Adicionar: Salva um novo cliente.
  - Atualizar: Altera os dados do cliente selecionado.
  - Deletar: Remove o cliente selecionado.
  - Buscar: Filtra os dados com base nos valores digitados.
  - Limpar: Limpa os campos de entrada.
- Tabela: Exibe os clientes registrados no banco.

Passos principais para usar:

1. Adicionar um cliente
  - Preencha os campos e clique em "Adicionar".
  - O cliente será salvo e a tabela será atualizada automaticamente.
2. Visualizar clientes
  - Todos os clientes aparecem automaticamente ao abrir o programa.
3. Buscar um cliente
  - Preencha um ou mais campos (como nome ou CPF) e clique em "Buscar".
  - A tabela mostrará apenas os clientes que combinam com os dados.
4. Atualizar dados
  - Clique em um cliente na tabela para preencher os campos.
  - Edite os dados desejados e clique em "Atualizar".
  - A tabela será atualizada.
5. Excluir um cliente
  - Selecione um cliente na tabela e clique em "Deletar".
  - O cliente será removido do banco de dados.

## Criando um executável com PyInstaller

Caso você queira rodar a aplicação sem precisar abrir o Python ou o terminal, é possível gerar um arquivo executável (.exe).

Passo a passo:

1. Instale o PyInstaller
  - No terminal, digite: `pip install pyinstaller`

## 2. Crie o executável

- Vá até a pasta do projeto no terminal.
- Execute: `pyinstaller --onefile application.py`
- Um executável será gerado dentro da pasta `dist`.

## 3. Executando o programa

- Vá até a pasta `dist` e execute o arquivo `application.exe`.
- O programa abrirá normalmente.

## 4. Observações:

- O executável pode ser grande (entre 50 e 100 MB) pois inclui tudo que o programa precisa.
- Para compartilhar com outra pessoa, envie o executável e, se necessário, o arquivo `clientes.db`.

## Sobre o banco de dados

Os dados são armazenados em um arquivo chamado `clientes.db`, que é criado automaticamente. A tabela dentro do banco tem os seguintes campos:

- `id`: Identificador único do cliente (gerado automaticamente).
- `nome`
- `sobrenome`
- `email`
- `cpf`

A função `initDB()` no código cria essa tabela na primeira vez que o programa é executado.

## Resolução de problemas comuns

- Erro: "No module named tkinter"  
Verifique se o Python está corretamente instalado. Se necessário, use o comando `pip install tk`.
- Erro ao rodar o programa  
Verifique se todos os arquivos estão na mesma pasta e se você está executando o comando dentro da pasta correta.
- O executável não abre  
Tente abrir pelo terminal para ver mensagens de erro. Execute o comando:  
`./dist/application.exe`
- Tabela não mostra os dados  
Confirme se o arquivo `clientes.db` está na mesma pasta do programa ou do executável.

## O que você pode aprender com este projeto

- Como usar Python para organizar códigos em arquivos e classes.
- Como trabalhar com SQL para inserir, consultar, atualizar e apagar dados.
- Como criar interfaces gráficas com Tkinter.
- Como aplicar boas práticas, como separar a lógica da interface e usar comandos SQL seguros.
- Como transformar um programa Python em um executável usando o PyInstaller.

## Ideias para melhorias no projeto

Você pode tentar adicionar novos recursos, como:

- Validar o formato do CPF.
- Exportar a lista de clientes para um arquivo CSV.
- Adicionar um botão "Recarregar" para atualizar os dados após uma busca.
- Melhorar o visual da interface com cores, ícones ou estilos.