

## Algoritmo Narrativo - Gerenciador de Clientes

Este é um guia passo a passo que explica como a aplicação "Gerenciador de Clientes" funciona. A aplicação permite gerenciar clientes (adicionar, visualizar, buscar, atualizar e deletar) usando uma interface gráfica Tkinter e um banco de dados SQLite.

### 1. Início da Aplicação

- O programa começa executando o arquivo `application.py`;
- O método `main()` é chamado, que:
  - Inicializa o banco de dados chamando `Backend.initDB()`;
  - Cria uma janela principal com Tkinter;
  - Instancia a classe `Gui` para exibir a interface gráfica;
  - Inicia o loop principal do Tkinter para manter a janela aberta.

### 2. Inicialização do Banco de Dados

- O método `Backend.initDB()`:
  - Cria uma conexão com o banco de dados SQLite (`clientes.db`);
  - Cria a tabela `clientes` (se ainda não existir) com as colunas:
    - `id`: Um número único gerado automaticamente (chave primária);
    - `nome`: Texto para o nome do cliente;
    - `sobrenome`: Texto para o sobrenome do cliente;
    - `email`: Texto para o e-mail do cliente;
    - `cpf`: Texto para o CPF do cliente;
  - Salva as alterações (`commit`) e fecha a conexão.

### 3. Exibição da Interface Gráfica

- A classe `Gui` cria uma janela com:
  - Um quadro com campos de texto para inserir Nome, Sobrenome, Email e CPF;

- Botões para as ações: Adicionar, Atualizar, Deletar, Buscar e Limpar.
- Uma tabela (Treeview) que mostra todos os clientes do banco de dados.
- Ao abrir, a tabela é preenchida automaticamente com os dados dos clientes, chamando Backend.view().

#### **4. Adicionar um Cliente**

- O usuário preenche os campos Nome, Sobrenome, Email e CPF;
- Clica no botão "Adicionar";
- O sistema verifica se todos os campos estão preenchidos;
  - Se sim, chama Backend.insert (nome, sobrenome, email, cpf) para inserir o cliente no banco;
- Salva as alterações (commit) e atualiza a tabela na interface;
- Mostra uma mensagem de sucesso;
- Limpa os campos de entrada;
- Se algum campo estiver vazio, mostra uma mensagem de erro.

#### **5. Visualizar Clientes**

- A tabela é preenchida automaticamente ao abrir a aplicação ou após ações como adicionar, atualizar ou deletar;
- O método Backend.view():
  - Conecta ao banco de dados;
  - Executa a consulta SQL: SELECT \* FROM clientes;
  - Retorna todos os registros para exibição na tabela.

#### **6. Buscar Clientes**

- O usuário insere um valor em qualquer campo (Nome, Sobrenome, Email ou CPF);
- Clica no botão "Buscar";
- O sistema chama Backend.search(nome, sobrenome, email, cpf):

- Executa uma consulta SQL: `SELECT * FROM clientes WHERE nome=? OR sobrenome=? OR email=? OR cpf=?`.

- Retorna os clientes que correspondem aos critérios;
- A tabela é atualizada para mostrar apenas os resultados da busca.

## **7. Atualizar um Cliente**

- O usuário seleciona um cliente na tabela (os campos de entrada são preenchidos automaticamente com os dados do cliente);
- Edita os campos desejados;
- Clica no botão "Atualizar";
- O sistema verifica se todos os campos estão preenchidos e se um cliente está selecionado;
- Se sim, chama `Backend.update(id, nome, sobrenome, email, cpf)` para atualizar o cliente no banco;
- Salva as alterações (`commit`) e atualiza a tabela;
- Mostra uma mensagem de sucesso;
- Limpa os campos de entrada;
- Se algum campo estiver vazio ou nenhum cliente estiver selecionado, mostra uma mensagem de erro.

## **8. Deletar um Cliente**

- O usuário seleciona um cliente na tabela;
- Clica no botão "Deletar";
- O sistema verifica se um cliente está selecionado;
- Se sim, chama `Backend.delete(id)` para remover o cliente do banco;
- Salva as alterações (`commit`) e atualiza a tabela;
- Mostra uma mensagem de sucesso;
- Limpa os campos de entrada;
- Se nenhum cliente estiver selecionado, mostra uma mensagem de erro.

## **9. Limpar Campos**

- O usuário clica no botão "Limpar";
- Todos os campos de entrada (Nome, Sobrenome, Email, CPF) são esvaziados.

## **10. Fechar a Aplicação**

- O usuário fecha a janela gráfica;
- O loop principal do Tkinter termina, encerrando o programa;
- O banco de dados (clientes.db) permanece salvo na pasta do projeto com todos os dados.

## **Fim do Algoritmo**

Este algoritmo mostra como a aplicação combina a interface gráfica (Tkinter) com o banco de dados (SQLite) para gerenciar clientes. Cada ação do usuário na interface desencadeia uma operação no banco de dados, mantendo os dados consistentes.