

Algoritmo Narrativo - Gerenciador de Clientes

Este documento descreve como a aplicação "Gerenciador de Clientes" funciona. A aplicação permite realizar operações de gerenciamento de clientes, como adicionar, visualizar, buscar, atualizar e deletar, utilizando a interface gráfica (Tkinter) e um banco de dados SQLite.

1. Início da Aplicação

- O programa é iniciado ao executar o arquivo `application.py`.
 - O método `main()` realiza as seguintes ações:
 - Inicializa o banco de dados através de `Backend.initDB()`.
 - Cria a janela principal com Tkinter.
 - Instancia a classe `Gui` para configurar a interface gráfica.
 - Inicia o loop principal do Tkinter para manter a janela aberta e interativa.
-

2. Inicialização do Banco de Dados

- O método `Backend.initDB()` executa os seguintes passos:
 - Conecta ao banco de dados SQLite (`clientes.db`).
 - Cria a tabela `clientes` (se não existir), com as colunas:
 - **id**: Identificador único (chave primária, auto-incrementável).
 - **nome**: Nome do cliente.
 - **sobrenome**: Sobrenome do cliente.
 - **email**: Endereço de email do cliente.
 - **cpf**: CPF do cliente.
 - Salva as alterações com o comando **commit** e fecha a conexão.
-

3. Exibição da Interface Gráfica

- A classe `Gui` configura a janela contendo:
 - Campos de entrada para os dados: Nome, Sobrenome, Email e CPF.
 - Botões para as funções: **Adicionar**, **Atualizar**, **Deletar**, **Buscar** e **Limpar**.
 - Uma tabela (Treeview) para exibir os registros dos clientes.
 - A tabela é preenchida automaticamente ao iniciar a aplicação, utilizando o método `Backend.view()`.
-

4. Adicionar um Cliente

- O usuário preenche os campos: Nome, Sobrenome, Email e CPF.

- Ao clicar em **Adicionar**, o sistema:
 - Verifica se todos os campos estão preenchidos.
 - Se estiverem completos, chama `Backend.insert(nome, sobrenome, email, cpf)` para inserir o cliente no banco de dados.
 - Realiza um **commit** para salvar os dados e atualiza a tabela na interface gráfica.
 - Exibe uma mensagem de sucesso e limpa os campos de entrada.
 - Se algum campo estiver vazio, exibe uma mensagem de erro.
-

5. Visualizar Clientes

- A tabela exibe automaticamente todos os clientes ao abrir a aplicação ou após ações como adicionar, atualizar ou deletar.
 - O método `Backend.view()` conecta-se ao banco de dados, executa a consulta SQL `SELECT * FROM clientes` e retorna todos os registros para exibição na tabela.
-

6. Buscar Clientes

- O usuário insere um valor em qualquer campo (Nome, Sobrenome, Email ou CPF) e clica em **Buscar**.
 - O sistema chama `Backend.search(nome, sobrenome, email, cpf)`, que executa a consulta SQL `SELECT * FROM clientes WHERE nome=? OR sobrenome=? OR email=? OR cpf=?`.
 - A tabela é atualizada para mostrar apenas os resultados da busca, filtrados pelos critérios inseridos.
-

7. Atualizar um Cliente

- O usuário seleciona um cliente na tabela, e os dados são carregados automaticamente nos campos de entrada.
 - O usuário edita os dados e clica em **Atualizar**.
 - O sistema verifica se todos os campos estão preenchidos e se um cliente foi selecionado.
 - Se todos os campos estiverem preenchidos, o sistema chama `Backend.update(id, nome, sobrenome, email, cpf)` para atualizar o cliente no banco de dados.
 - Realiza um **commit**, atualiza a tabela e exibe uma mensagem de sucesso.
 - Se algum campo estiver vazio ou nenhum cliente for selecionado, exibe uma mensagem de erro.
-

8. Deletar um Cliente

- O usuário seleciona um cliente na tabela e clica em **Deletar**.
 - O sistema verifica se um cliente foi selecionado.
 - Se sim, chama `Backend.delete(id)` para excluir o cliente do banco de dados.
 - Realiza um **commit**, atualiza a tabela e exibe uma mensagem de sucesso.
 - Se nenhum cliente for selecionado, exibe uma mensagem de erro.
-

9. Limpar Campos

- O usuário clica em **Limpar** para esvaziar todos os campos de entrada (Nome, Sobrenome, Email, CPF).
-

10. Fechar a Aplicação

- O usuário fecha a janela gráfica.
 - O loop principal do Tkinter é encerrado, finalizando o programa.
 - O banco de dados (`clientes.db`) permanece salvo na pasta do projeto, mantendo todos os dados.
-

Fim do Algoritmo

Este algoritmo descreve a interação entre a interface gráfica (Tkinter) e o banco de dados SQLite para realizar operações de gerenciamento de clientes. Cada ação do usuário na interface aciona operações correspondentes no banco de dados, mantendo os dados sempre consistentes e atualizados.