

Fluxo Operacional - Sistema de Gestão de Clientes

Este documento descreve o funcionamento da aplicação "Sistema de Gestão de Clientes". Trata-se de uma aplicação com interface gráfica (Tkinter) e banco de dados SQLite, permitindo operações de cadastro, visualização, busca, edição e exclusão de clientes.

1. Inicialização do Sistema

- A execução inicia pelo arquivo `application.py`.
 - A função `main()` é chamada, realizando:
 - A preparação do banco de dados via `Backend.initDB()`.
 - A criação da janela principal com o Tkinter.
 - A inicialização da interface com a classe `Gui`.
 - O início do loop principal da interface gráfica.
-

2. Configuração do Banco de Dados

- O método `initDB()` no módulo `Backend`:
 - Conecta ao arquivo de banco de dados SQLite (`clientes.db`).
 - Cria a tabela `clientes`, caso ainda não exista, com os seguintes campos:
 - `id` (chave primária, autoincremento)
 - `nome` (texto)
 - `sobrenome` (texto)
 - `email` (texto)
 - `cpf` (texto)
 - As alterações são salvas e a conexão encerrada.
-

3. Interface Gráfica (GUI)

- A classe `Gui` monta a tela principal com:
 - Campos para entrada de dados: Nome, Sobrenome, Email e CPF.
 - Botões com ações: Adicionar, Editar, Remover, Pesquisar e Limpar.
 - Uma tabela (Treeview) para exibir os registros existentes.
 - Ao iniciar, a tabela é automaticamente preenchida usando `Backend.view()`.
-

4. Cadastro de Novo Cliente

- O usuário preenche os campos e clica em "Adicionar".
- O sistema valida se todos os dados foram informados:
 - Se validado, chama `Backend.insert(...)` para inserir os dados.

- Após a inserção:
 - A tabela é atualizada.
 - Uma mensagem de sucesso é exibida.
 - Os campos são limpos.
 - Se faltar algum campo, uma mensagem de erro é mostrada.
-

5. Listagem de Clientes

- A tabela é atualizada sempre que a aplicação inicia ou quando ocorre alguma operação.
 - O método `Backend.view()`:
 - Recupera todos os registros da tabela `clientes`.
 - Os dados são exibidos na interface.
-

6. Pesquisa de Clientes

- O usuário preenche qualquer um dos campos e clica em "Buscar".
 - O sistema chama `Backend.search(...)`, que:
 - Executa uma consulta SQL com os critérios informados.
 - Retorna os clientes correspondentes.
 - A tabela exibe apenas os resultados da pesquisa.
-

7. Atualização de Dados

- O cliente é selecionado na tabela, preenchendo os campos automaticamente.
 - O usuário edita os dados e clica em "Atualizar".
 - O sistema verifica se há um cliente selecionado e se os campos estão completos:
 - Se sim, executa `Backend.update(...)` para salvar as alterações.
 - A tabela é atualizada e os campos são limpos.
 - Uma confirmação é exibida.
 - Caso contrário, uma mensagem de erro é apresentada.
-

8. Remoção de Cliente

- O usuário seleciona um cliente e clica em "Deletar".
- O sistema confirma a seleção e executa:
 - `Backend.delete(id)` para remover o cliente.
 - Atualização da tabela e limpeza dos campos.
 - Exibição de mensagem de confirmação.
 - Caso não haja seleção, é exibido um aviso.

9. Limpeza dos Campos

- Ao clicar em "Limpar", todos os campos de entrada são esvaziados para novo uso.

10. Encerramento da Aplicação

- O programa é finalizado ao fechar a janela principal.
- O banco de dados permanece armazenado com os registros salvos.

Resumo

A aplicação oferece uma solução prática para o gerenciamento de clientes, integrando interface amigável com banco de dados local. Todas as interações do usuário refletem operações diretas e seguras na base de dados.