Gerenciador de Clientes – Documentação Técnica

Descrição Geral:

Aplicação desenvolvida em Python para gerenciamento de dados de clientes, armazenados em banco de dados SQLite. A interface gráfica é implementada com Tkinter, possibilitando operações CRUD (Create, Read, Update, Delete) sobre os registros de clientes, que contêm os campos: nome, sobrenome, email e CPF.

Funcionalidades Principais

- Inserção de novos registros de clientes.
- Visualização completa dos clientes cadastrados em formato tabular.
- Busca de clientes por qualquer campo (nome, sobrenome, email, CPF).
- Atualização dos dados de clientes existentes.
- Exclusão de clientes do banco de dados.

Arquitetura do Projeto

Estrutura Modular:

Gui.py:

Implementa a interface gráfica com Tkinter, incluindo campos de entrada, botões para execução das operações e uma tabela para exibição dos clientes. Esta camada interage com o backend para manipulação dos dados.

Backend.py:

Responsável pela camada de acesso a dados, gerenciando a conexão SQLite, criação de tabelas, e execução das operações de CRUD via consultas SQL parametrizadas, garantindo segurança contra injeção de código.

application.py:

Ponto de entrada da aplicação, inicializa o banco de dados e a interface gráfica.

Requisitos para Execução

Python 3.x (versão 3.8 ou superior recomendada).

Tkinter (biblioteca gráfica padrão do Python).

SQLite (módulo sqlite3 incluso no Python).

PyInstaller (opcional, para empacotamento em executável).

Guia de Configuração e Execução

Organizar os arquivos Gui.py, Backend.py e application.py em um diretório dedicado.

Verificar instalação do Python via terminal (python --version).

Executar o sistema com o comando:

python application.py

A interface gráfica será exibida, permitindo interações com os dados.

Uso da Aplicação

Inserir Cliente: Preencher os campos obrigatórios e clicar em "Adicionar".

Visualizar Clientes: A tabela apresenta todos os registros carregados ao iniciar.

Buscar Cliente: Inserir critério em qualquer campo e clicar em "Buscar".

Atualizar Cliente: Selecionar um cliente na tabela, editar campos e clicar em "Atualizar".

Deletar Cliente: Selecionar um cliente e clicar em "Deletar".

Limpar Campos: Resetar os campos de entrada.

Empacotamento em Executável com PyInstaller

Instalar PyInstaller:

pip install pyinstaller

Gerar executável:

pyinstaller --onefile application.py

O executável fica disponível na pasta dist/, podendo ser distribuído sem necessidade de instalação do Python.

Estrutura do Banco de Dados

Banco SQLite armazenado no arquivo clientes.db.

Tabela clientes com campos:

id (INTEGER, chave primária, auto-increment)

nome (TEXT, NOT NULL)

sobrenome (TEXT, NOT NULL)

email (TEXT, NOT NULL)

cpf (TEXT, NOT NULL)

A criação da tabela é automática na inicialização da aplicação via método Backend.initDB().

Solução de Problemas Comuns

Erro “No module named tkinter”: Verificar instalação correta do Python e Tkinter.

Falha na execução: Confirmar se todos os arquivos estão na mesma pasta e o comando é executado no diretório correto.

Executável não abre: Verificar mensagens de erro ao executar pelo terminal.

Tabela não atualiza: Confirmar que o arquivo clientes.db está presente na pasta do executável.

Aprendizados Propostos

Estruturação de código em módulos e classes Python.

Manipulação segura de banco de dados SQLite via SQL parametrizado.

Desenvolvimento de interface gráfica com Tkinter.

Empacotamento e distribuição de aplicações Python com PyInstaller.

Boas práticas de programação, incluindo separação de responsabilidades e documentação clara.

Sugestões para Extensão do Projeto

Implementação de validação para formato do CPF.

Exportação dos dados para arquivos CSV.

Inclusão de funcionalidade para recarregar os dados exibidos.

Aperfeiçoamento visual da interface com elementos gráficos adicionais.