

UNIVERSIDADE ESTÁCIO DE SÁ
CURSO: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
UNIDADE: NOVA AMERICA

TRABALHO DE RAD
EM DESENVOLVIMENTO RÁPIDO DE APLICAÇÕES EM
PYTHON

Rio de Janeiro - RJ
Junho / 2025

202403805731 – Rafael Avoglio Alves Oliveira Filho

**projeto de RAD
em Desenvolvimento Rápido de Aplicações em Python**

Trabalho de Rad apresentado a
Universidade Estácio de Sá,
como exigência para avaliação
na disciplina Desenvolvimento
Rápido de Aplicações em Python

Orientador:
Prof. Ronaldo Candido dos Santos

SUMÁRIO

1 INTRODUÇÃO 3

1.1 DESCRIÇÃO DO PROBLEMA..... 3

1.2 OBJETIVOS 3

2 DESENVOLVIMENTO 4

2.1 XXXXXXXXXXXXXXX 4

3 CONCLUSÃO 6

REFERÊNCIAS..... 7

1 INTRODUÇÃO

Este trabalho apresenta o desenvolvimento de um sistema de cadastro de pessoas com interface gráfica em Python. A motivação surgiu da necessidade de organizar dados pessoais de forma prática e automatizada, utilizando a biblioteca Tkinter e o banco de dados SQLite.

1.1 DESCRIÇÃO DO PROBLEMA

O cadastro manual de pessoas é sujeito a erros e dificulta a busca e a integridade dos dados. Por isso, propõe-se um sistema simples que permita gerenciar essas informações com eficiência.

1.2 OBJETIVOS

O principal objetivo deste trabalho é criar uma interface gráfica utilizando Python, com o auxílio da biblioteca Tkinter. O sistema desenvolvido deve permitir armazenar dados em um banco de dados SQLite, viabilizando o cadastro, a busca, a edição e a exclusão de registros de forma prática. Além disso, foram implementados filtros de busca para facilitar a localização de informações específicas, como por nome, CPF ou ID. Também foram incluídas validações nos campos, especialmente para dados sensíveis como CPF e telefone, garantindo maior integridade e confiabilidade ao sistema.

2 DESENVOLVIMENTO

Este trabalho teve como foco o desenvolvimento de um sistema de cadastro de pessoas, implementado com a linguagem Python, utilizando a biblioteca Tkinter para a criação da interface gráfica e SQLite para a gestão dos dados.

O objetivo foi construir uma aplicação funcional que possibilitasse a inserção, consulta, atualização e exclusão de registros em um banco de dados local. Para isso, adotou-se uma abordagem prática baseada na metodologia de desenvolvimento incremental, testando e validando cada parte da aplicação conforme sua implementação.

A estrutura do sistema segue o modelo CRUD (Create, Read, Update, Delete), muito comum em aplicações de banco de dados. Foram implementadas validações específicas para campos sensíveis como CPF (que deve conter exatamente 11 dígitos numéricos) e celular (somente números), além da verificação de unicidade no banco de dados para esses campos.

A interface foi projetada para ser intuitiva e responsiva. Os campos de entrada de dados foram centralizados para facilitar o uso, enquanto os botões de ação ficaram organizados verticalmente à direita da tela. O botão de busca foi posicionado no canto esquerdo, logo acima da lista de cadastros, permitindo ao usuário filtrar os dados por nome, CPF ou ID. Foram incluídas checkboxes para facilitar a seleção do tipo de filtro, além de um botão para limpar os critérios de busca.

Essas implementações refletem práticas modernas de desenvolvimento de software voltadas à experiência do usuário (UX), garantindo não apenas funcionalidade, mas também acessibilidade e organização visual.

2.1 O sistema foi dividido em três principais componentes:

- **Interface Gráfica com Tkinter**

A biblioteca Tkinter, nativa do Python, foi utilizada para a construção da interface gráfica. Essa escolha se deve à sua simplicidade e integração com a linguagem, permitindo criar janelas, botões, campos de entrada e caixas de listagem com facilidade.

O layout foi ajustado com o uso do gerenciador de layout `grid`, permitindo que os elementos da interface se adaptassem dinamicamente ao tamanho da janela, garantindo responsividade e uma melhor organização dos dados.

- **Banco de Dados com SQLite**

Para o armazenamento das informações, utilizou-se o banco de dados SQLite, que é leve, embutido e não requer instalação separada. Isso tornou o projeto mais portátil e fácil de executar em diferentes máquinas. A tabela foi estruturada com os seguintes campos: `id`, `nome`, `idade`, `cpf`, `gênero`, `email` e `celular`.

- **Funcionalidades Implementadas**

As principais funcionalidades do sistema incluem:

- **Cadastro de Pessoas:** Com validações de campos obrigatórios e formato dos dados.
- **Busca com Filtros:** Busca personalizada por nome, CPF ou ID, com checkboxes para selecionar os critérios.
- **Atualização e Exclusão:** Edição de dados existentes e remoção de registros diretamente pela

lista.

– **Interface Responsiva:** Adaptação ao redimensionamento da janela, com elementos centralizados.

O desenvolvimento do projeto permitiu aplicar conceitos fundamentais de programação, banco de dados e interfaces gráficas, além de promover reflexões sobre usabilidade e organização de dados, aspectos essenciais no desenvolvimento de sistemas voltados ao usuário final.

3 CONCLUSÃO

O sistema desenvolvido atinge o objetivo proposto de facilitar o cadastro e a gestão de pessoas por meio de uma interface gráfica intuitiva e um banco de dados leve e funcional. A implementação com Tkinter e SQLite mostrou-se eficaz para aplicações locais, dispensando servidores externos e oferecendo boa performance para usos simples. Além disso, a experiência com validação de dados, manipulação de banco de dados e construção de janelas adicionais com filtros reforça conhecimentos fundamentais em programação e desenvolvimento de software.

REFERÊNCIAS

(Elemento obrigatório constituído por uma lista ordenada dos documentos efetivamente citados no texto. Não devem ser referenciadas fontes bibliográficas que não foram citadas no texto. Indicar todos os artigos, livros, sites consultados e utilizados para o desenvolvimento deste trabalho. Exemplos:)

ASCENCIO, A. F. G.; ARAÚJO, G. S. de. Estrutura de Dados: Algoritmos, Análise da Complexidade e implementações em Java e C/C++. São Paulo: Pearson Prentice Hall, 2010. Capítulo 2: Algoritmos de ordenação e busca. Páginas 21-102.

BACKES, A. Vídeo [ED] Aula 52 - Ordenação - QuickSort. Disponível em: <https://www.youtube.com/watch?v=RZbg5oT5Fgw>. Acesso em: 22 mai. 2024.

BLOODSHED DEV C++: Download do software. Disponível em: <https://sourceforge.net/projects/orwelldevcpp/>. Acesso em: 28 mai. 2024.

CORMEN, T. Desmistificando algoritmos. Rio de Janeiro: Elsevier, 2014. Capítulo 3: Algoritmos para ordenar e buscar. Páginas 20-49.

KOFFMAN, E. B.; WOLFGANG, P. A. T. Abstração, Estruturas de Dados e Projeto Usando C++. Rio de Janeiro: LTC, 2008. Capítulo 10: Ordenação.

NORMAS ABNT. Normas para elaboração de trabalhos acadêmicos. Disponível em: <https://www.normasabnt.org/>. Acesso em: 28 mai. 2024.