

UNIVERSIDADE ESTÁCIO DE SÁ
CURSO ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
UNIDADE NOVA AMÉRICA

**TRABALHO DE GESTÃO DE CLIENTES
EM PYTHON (DESENVOLVIMENTO RÁPIDO DE
APLICAÇÕES EM PYTHON)**

Novembro / 2024

Trabalho de Gestão de Clientes em Python
(Desenvolvimento Rápido de Aplicações em Python)

Trabalho de Gestão de Clientes
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	5
2.1	XXXXXXXXXXXXXXXX	4
3	CONCLUSÃO	7
	REFERÊNCIAS.....	8

1 INTRODUÇÃO

O presente documento visa descrever o desenvolvimento de um sistema de gerenciamento de proponentes para a associação em que trabalho. O projeto foi iniciado com a necessidade de ter um controle eficiente sobre os dados de proponentes, ou seja, possíveis associados que já receberam uma proposta, seja por meio de um divulgador ou através de contato direto com a Associação. Esse sistema pretende facilitar o pré-cadastro dos proponentes e registrar informações fundamentais, como dados pessoais e o valor da contribuição associativa, a fim de acompanhar e gerenciar o processo de prospecção de novos membros para a Associação. A proposta principal do sistema é garantir uma ferramenta intuitiva que permita aos responsáveis pela prospecção terem acesso simplificado aos dados dos proponentes, tornando o processo de contato e fechamento de contrato mais organizado e eficiente.

1.1 DESCRIÇÃO DO PROBLEMA

A Associação enfrenta desafios na organização e acompanhamento dos dados de proponentes que demonstraram interesse em se associar. O processo manual de registro e prospecção não é eficiente e pode levar à perda de informações ou atrasos no contato com os proponentes, dificultando a conclusão do processo de adesão. É necessário um sistema que permita o cadastro, atualização e exclusão dos dados de forma ágil, oferecendo uma visão clara sobre o status de cada proponente e possibilitando o controle sobre o progresso das propostas realizadas.

1.2 OBJETIVOS

O objetivo principal deste projeto é desenvolver uma solução que atenda à necessidade de organizar e gerenciar os dados de proponentes de forma prática e acessível. O sistema permitirá:

- Cadastro e prospecção: Adicionar e manter registros completos de proponentes, incluindo informações de contato, dados pessoais e valor da contribuição associativa.
- Acompanhamento de status: Acompanhar em qual etapa do processo de prospecção cada proponente está, permitindo visualizar de forma clara os dados necessários para o próximo passo.
- Facilidade de atualização e exclusão: Facilitar o processo de atualização e

exclusão de registros, tornando o gerenciamento dos dados mais eficaz.

2 DESENVOLVIMENTO

O sistema de gerenciamento de proponentes foi desenvolvido utilizando Python e a biblioteca Streamlit, visando simplicidade na interface e rapidez no desenvolvimento. A estrutura do projeto foi pensada para garantir que o sistema fosse acessível a usuários com pouca experiência em tecnologia, oferecendo uma experiência intuitiva.

O projeto foi organizado em módulos principais que facilitam o desenvolvimento e a manutenção do código. Entre as funcionalidades implementadas, destacam-se:

- **Cadastro de Proponentes:** Interface para a inserção de novos proponentes no sistema, armazenando dados essenciais, como CPF, idade, órgão público associado e valor do contrato. Foi desenvolvido um método para formatação do CPF, garantindo que a entrada de dados esteja padronizada e facilitando o processo de validação.
- **Consulta e Edição de Proponentes:** Página que permite visualizar os proponentes cadastrados, com opções para editar dados existentes. A atualização é realizada de forma a refletir imediatamente no banco de dados e garantir que as informações estejam sempre atualizadas.
- **Exclusão de Proponentes:** Para facilitar a gestão, o sistema permite a exclusão de proponentes não desejados ou duplicados. A exclusão é confirmada com uma lista de todos os proponentes cadastrados, exibida em formato de tabela após a ação.
- **Banco de Dados:** O sistema utilizou o SQLite para armazenamento de dados nas fases iniciais do projeto, garantindo persistência das informações e permitindo consultas rápidas. A estrutura foi pensada para que o sistema fosse escalável e possibilitasse futura integração com outros bancos, e isso ocorreu na versão final, sendo utilizado o PostgreSQL.
- **Organização do Código e Manutenção:** Para manter a organização, as funcionalidades foram divididas em arquivos e módulos separados. Um exemplo é a separação entre o arquivo de controle de proponentes

(ProponenteController) e os modelos de dados, como o arquivo que define a classe Proponente. Com isso, o código está organizado e pronto para receber melhorias no futuro.

2.1 FUNCIONALIDADES DO SISTEMA

Cadastro e Atualização de Proponentes

O sistema permite que cada proponente faça seu pré-cadastro, registrando nome, idade, CPF, órgão público ao qual está associado e o valor proposto para o contrato. As informações podem ser atualizadas a qualquer momento, facilitando o processo de acompanhamento e eventual edição de dados.

Prospecção e Fechamento de Contrato

Proponentes são cadastrados com o objetivo de prospecção e acompanhamento até o fechamento do contrato associativo. O sistema permite visualização e consulta dos dados a qualquer momento, organizando o fluxo para que os responsáveis pela prospecção possam gerenciar as etapas e garantir que nenhum contato seja perdido.

Visualização e Exclusão de Proponentes

Com o intuito de facilitar a exclusão de registros desnecessários, como proponentes que perderam interesse, o sistema exibe uma tabela de dados logo após o botão de exclusão, permitindo uma visão clara e atualizada da base de dados. Essa tabela é atualizada a cada operação de exclusão.

Banco de Dados e Persistência

Utilizando PostgreSQL o sistema consegue manter as informações de forma permanente, o que é ideal para o tipo de aplicação proposta.

3 CONCLUSÃO

Desenvolver este sistema de gerenciamento de proponentes foi uma experiência muito enriquecedora e trouxe uma solução prática para um problema real dentro da Associação. Agora, temos uma ferramenta simples e direta para registrar, acompanhar e gerenciar os dados de pessoas interessadas em se associar. Com isso, o processo de prospecção se torna mais eficiente e organizado, o que facilita tanto o acompanhamento quanto a tomada de decisões.

A escolha de usar o PostgreSQL em vez do SQLite foi motivada pela necessidade de um banco de dados mais robusto, já que isso nos permite gerenciar melhor os dados e ter mais flexibilidade para escalas maiores no futuro. Isso também abre portas para integrações mais complexas, se precisarmos expandir o sistema.

Em resumo, o projeto trouxe uma melhora considerável para nossa equipe, simplificando o trabalho de cadastro e monitoramento dos proponentes e garantindo que essas informações fiquem centralizadas e acessíveis.

REFERÊNCIAS

BUG, C. C. Como criar um CRUD WEB em python | Streamlit #1. Disponível em: <<https://www.youtube.com/watch?v=9mnNSMCu3dl&list=PLhna1crYw0SMogdpgsahknQUOmlrr74hY&index=3>>. Acesso em: 6 nov. 2024.

BUG, C. C. Conectando ao banco de dados com python | Streamlit #2. Disponível em: <<https://www.youtube.com/watch?v=IOyCICREgy8&list=PLhna1crYw0SMogdpgsahknQUOmlrr74hY&index=4>>. Acesso em: 6 nov. 2024.

BUG, C. C. Consultando em Python os cadastros no banco de dados | Streamlit #3. Disponível em: <<https://www.youtube.com/watch?v=JOKxfZEPrDI&list=PLhna1crYw0SMogdpgsahknQUOmlrr74hY&index=5>>. Acesso em: 6 nov. 2024.

BUG, C. C. Como separar as páginas em arquivos Python com Streamlit #4. Disponível em: <https://www.youtube.com/watch?v=AqW_6hj1Vml&list=PLhna1crYw0SMogdpgsahknQUOmlrr74hY&index=6>. Acesso em: 6 nov. 2024.

BUG, C. C. Como excluir dados do banco com aplicativo WEB em Python #5. Disponível em: <https://www.youtube.com/watch?v=_lKn0rD0oFw&list=PLhna1crYw0SMogdpgsahknQUOmlrr74hY&index=8>. Acesso em: 6 nov. 2024.

BUG, C. C. Streamlit | Alterando dados no banco com Python #6. Disponível em: <<https://www.youtube.com/watch?v=GNN819yeOT4&list=PLhna1crYw0SMogdpgsahknQUOmlrr74hY&index=9>>. Acesso em: 6 nov. 2024.

BUG, C. C. Streamlit | Alterando dados no banco com Python #7. Disponível em: <<https://www.youtube.com/watch?v=6yJgF3LuHU8&list=PLhna1crYw0SMogdpgsahknQUOmlrr74hY&index=10>>. Acesso em: 6 nov. 2024.

BUG, C. C. Streamlit | Alterando dados no banco com Python #8. Disponível em: <<https://www.youtube.com/watch?v=zP2mRzjPl-A&list=PLhna1crYw0SMogdpgsahknQUOmlrr74hY&index=11>>. Acesso em: 6 nov. 2024.