

Banco de Doações e Voluntariado

Informações da Equipe

Nome do Integrante	Matrícula	Curso	E-mail institucional
Ana Beatriz M. Santiago	570493	Ciência da Computação	anasantiago@alu.ufc.br
Carlos Daniel	566429	Ciência da Computação	carlos.rodrigueszx81@alu.ufc.br
Murilo Fragoso	570701	Ciência da Computação	murilofragoso@alu.ufc.br
Paulo Josué	567608	Ciência da Computação	paulojosuebarros@alu.ufc.br

1. Introdução

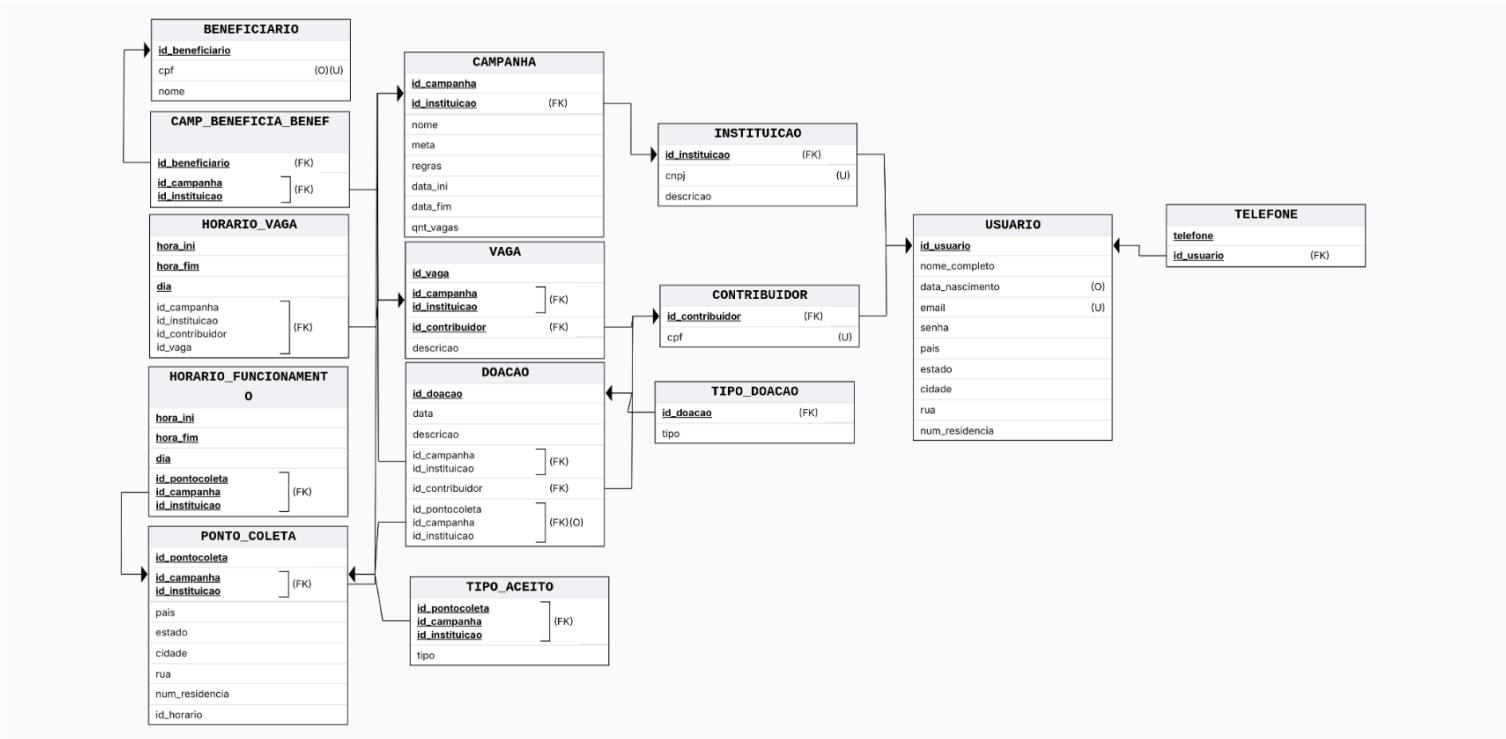
A aplicação ajuda instituições sociais a centralizarem campanhas de arrecadação e mobilização de voluntários. As instituições registram-se, criam campanhas com metas, prazos e opcionalmente registram pontos de coleta físicos para entrega de itens; também podem acompanhar doações recebidas e voluntários inscritos.

Usuários — contribuidores e instituições — cadastram-se para criar/localizar campanhas por categoria, status ou proximidade, ofertar doações e inscrever-se em ações presenciais ou remotas. O sistema prioriza facilidade de uso, registro transparente de transações e comunicação entre instituição e usuário, permitindo relatórios para prestação de contas e análise de impacto.

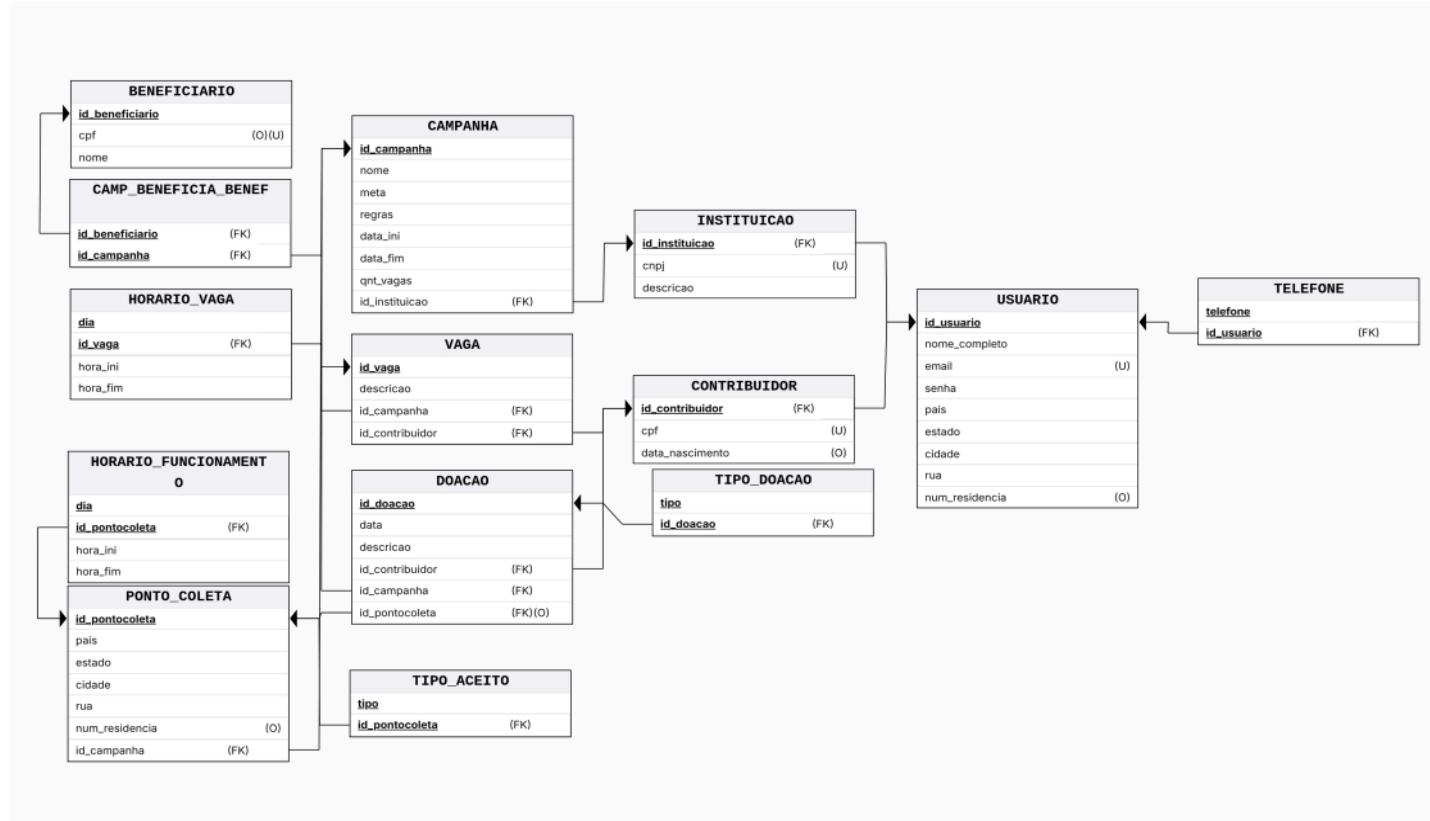
Objetivo principal: aumentar eficiência e transparência no atendimento às necessidades da comunidade.

2. Alterações do Diagrama Relacional

Versão Antiga



Versão Nova



Sobre as alterações

Na nova versão do modelo relacional, reduzimos o uso de chaves estrangeiras como chave primária, substituindo-as por chaves substitutas (*surrogate keys*). Como foi o caso das tabelas **VAGA**, **CAMPANHA**, **PONTO_COLETA**, **TIPO_ACEITO**, **CAMP_BENEFICIA_BENEF**.

Na tabela **HORARIO_FUNCIONAMENTO** a chave primária era somente o 'id_pontocoleta'. Mas, como o ponto de coleta pode ter vários horários de funcionamento (ao longo da semana, por exemplo), acrescentamos o atributo *dia* na chave primária. O mesmo caso ocorreu na tabela **HORARIO_VAGA**, então atualmente a chave é composta por **id_vaga** e **dia**.

Na tabela **TIPO_DOACAO** a chave primária era só o atributo **id_doacao**, mas a doação pode ter mais de um tipo, então foi acrescentado **tipo** na chave primária. Mesmo caso na tabela **TIPO_ACEITO**, onde a chave era apenas **id_pontocoleta**, colocamos o **tipo** como parte da chave primária.

O atributo **data_nascimento** foi removido da Tabela **USUARIO** e passou a ser atributo apenas da subclasse Contribuidor. Na Tabela **USUARIO**, o atributo **num_residencia** na tabela **USUARIO** agora é opcional, pois há casas sem número. O mesmo acontece na tabela **PONTO_COLETA**, que além disso teve seu atributo **id_horario** removido, que não estava em uso.

3. Descrição das telas desenvolvidas

1. Estrutura Geral das Telas

O sistema utiliza um padrão visual consistente para todas as telas de gerenciamento:

- **Layout de Duas Colunas:** À esquerda contém o formulário de entrada de dados e botões de ação; a direita exibe uma tabela interativa (**Tabulator**) com os dados do banco.
- **Navegação:** Cada tela possui um botão "Voltar para Home" para retornar ao menu principal.

- **Notificações:** O sistema utiliza `pn.state.notifications` para informar ao usuário se as operações no banco foram bem-sucedidas ou se ocorreram erros.

2. Detalhamento do CRUD de Usuário

Gerencia a tabela **USUARIO**, utilizando o **e-mail** como identificador principal para buscas e alterações.

- **Consulta (`on_consultar_user`):** Filtra usuários pelo e-mail fornecido. Se o campo estiver vazio, o sistema obriga o preenchimento para realizar a busca específica.
- **Inserção (`on_inserir_user`):** Recebe nome, e-mail, senha e endereço. O e-mail é obrigatório para evitar registros nulos.
- **Alteração (`on_atualizar_user`):** Atualiza todos os dados (exceto o ID) baseando-se no e-mail fornecido.
- **Remoção (`on_excluir_user`):** É a função mais complexa. Ela busca o `id_usuario` via e-mail e remove em cascata as referências nas tabelas `instituicao` e `contribuidor` antes de excluir o registro da tabela pai (`usuario`), garantindo a integridade referencial.

3. Detalhamento do CRUD de Campanha

Gerencia a tabela **CAMPANHA**, focando na organização de metas e prazos.

- **Consulta (`on_consultar_campaign`):** Realiza uma busca textual pelo nome da campanha utilizando o operador `LIKE` e formatando a entrada com `title()` para padronização.
- **Inserção (`on_inserir_campaign`):** Registra nome, meta, regras, datas (início/fim) e o ID da instituição responsável. Nome e ID da instituição são campos obrigatórios.
- **Alteração (`on_atualizar_campaign`):** Atualiza os detalhes da campanha localizando o registro pelo `id_campanha`.
- **Remoção (`on_excluir_campaign`):** Remove o registro baseado no ID. O código inclui um comentário técnico alertando que doações dependentes podem impedir a exclusão devido a restrições de chave estrangeira (FK).

4. Detalhamento do CRUD de Ponto de Coleta

Gerencia a tabela **PONTO_COLETA**, onde estão locais físicos de arrecadação vinculados às campanhas.

- **Consulta (`on_consultar_cpoint`)**: Realiza uma busca combinada obrigatória por **cidade e rua**, formatando ambas para letras iniciais maiúsculas.
- **Inserção (`on_inserir_cpoint`)**: Cadastra o endereço completo e o ID da campanha vinculada. Todos os campos geográficos e o ID da campanha são obrigatórios.
- **Alteração (`on_atualizar_cpoint`)**: Permite modificar o endereço e a campanha vinculada através do ID do ponto de coleta.
- **Remoção (`on_excluir_cpoint`)**: Exclui o ponto pelo ID, sujeito a validação de dependência no banco de dados.

5. Detalhamento do CRUD de Doação

Gerencia a tabela de **DOACAO**, que é o registro histórico de itens doados.

- **Consulta (`on_consultar_donation`)**: Diferente dos demais, busca um registro específico pelo `id_doacao`. Se o ID for zero, retorna a listagem completa.
- **Inserção (`on_inserir_donation`)**: Registra a data, descrição do item e vincula obrigatoriamente aos IDs do contribuidor e da campanha.
- **Alteração (`on_atualizar_donation`)**: Permite corrigir a descrição, data ou vínculos da doação através de um ID válido.
- **Remoção (`on_excluir_donation`)**: Remove o registro de doação pelo seu ID e remove em cascata as referências na tabela **tipo_doacao** e antes de excluir o registro da tabela pai (**doacao**), garantindo a integridade referencial.

6. Tela de Gráficos (Análise Estatística)

Esta tela não possui funções de edição, apenas de consulta agregada para gerar inteligência de dados:

- **Voluntários por Campanha**: Gráfico de barras que conta registros na tabela **VAGA** por campanha.

- **Distribuição por Tipo:** Gráfico de pizza mostrando a proporção de categorias de itens doados.
- **Quantidade de Doações:** Comparativo visual do volume de arrecadação entre campanhas.
- **Usuários por Estado:** Mapeamento geográfico da base de usuários.

4. Distribuição das Tarefas na Equipe

Nome do Integrante	Tarefa(s) realizada(s)	Descrição resumida das atividades
Ana Beatriz	Atualização no diagrama Relacional. Tela de Usuário. Gráfico de Usuários por estado.	Colaborou na atualização do esquema relacional. E ficou responsável pela tela Usuário.
Carlos Daniel	Atualização no diagrama Relacional. Tela de Doação. Gráfico de Doações por campanha.	Colaborou na atualização do esquema relacional. E ficou responsável pela tela Doação.
Murilo Fragoso	Atualização no diagrama Relacional. Tela de Ponto de Coleta. Gráfico de Doações por tipo.	Colaborou na atualização do esquema relacional. E ficou responsável pela tela Ponto de Coleta.
Paulo Josué	Atualização no diagrama Relacional. Tela de Campanha. Gráfico de Voluntários por campanha.	Colaborou na atualização do esquema relacional. E ficou responsável pela tela Campanha.

5. Conclusão

O desenvolvimento deste sistema demonstrou como uma boa modelagem de banco de dados é a base para qualquer aplicativo funcional. Ao conectar o script **SQL** à interface em **Python**, conseguimos transformar tabelas de dados em uma ferramenta prática para o dia a dia.

Os pontos principais desta etapa foram:

- **Segurança e Organização:** O banco de dados foi configurado para que nenhuma informação seja excluída por erro ou fique "solta", mantendo tudo conectado corretamente (usuários, campanhas e doações).
- **Visão do Todo:** Com os gráficos, o sistema deixa de apenas guardar nomes e passa a mostrar resultados reais, como quais campanhas estão funcionando melhor e onde estão nossos doadores.

6. Link referente ao repositório do projeto no GitHub

<https://github.com/carlosrodrigueszx/SistemaDoacoes>

7. Referências

SILBERSCHATZ, Abraham; KORTH, Henry; SUDARSHAN, S. Sistemas de Banco de Dados. 6^a ed. São Paulo: Pearson, 2011.