

DESENVOLVIMENTO DE UM SOFTWARE DE CONTROLE FINANCEIRO PARA UTILIZAÇÃO NA GESTÃO DE ORGANIZAÇÃO NÃO GOVERNAMENTAL

Adryan Luís Machado da Silva
Graduando em Curso do Aluno (a) pela Fatec Bauru
Adryan.silva@fatec.sp.gov.br

Marcos Junior Teixeira da Silva
Graduando em Curso do Aluno (a) pela Fatec Bauru
marcos.silva398@fatec.sp.gov.br

Orientador: Luís Alexandre da Silva
Mestre em Ciência da Computação e Docente na Fatec Bauru
luis.silva51@fatec.sp.gov.br

RESUMO

Os grupos voluntários desempenham um papel crucial na melhoria do bem-estar social, mobilizando ações beneficentes para beneficiar comunidades. No entanto, muitas organizações enfrentam desafios significativos na gestão financeira devido à utilização de métodos tradicionais e arcaicos de controle de caixa e contabilidade. Esta limitação pode resultar em perda de tempo e recursos, impedindo a eficácia e eficiência das atividades beneficentes realizadas. Para superar essas dificuldades, este trabalho tem como objetivo o desenvolvimento de um software de controle financeiro personalizado para grupos voluntários. Este software visa automatizar tarefas administrativas, como conferência de caixa e somatória de valores arrecadados, proporcionando uma ferramenta adaptada às necessidades específicas dessas organizações. Ao implementar essa solução tecnológica, espera-se fortalecer as organizações voluntárias, aumentando seu impacto positivo na comunidade.

Palavras-chave: Grupos voluntários; Software financeiro personalizado; Gestão eficiente; Impacto comunitário; Banco de Dados;

1 INTRODUÇÃO

A atuação de grupos voluntários é essencial para promover o bem-estar e o desenvolvimento das comunidades, através de iniciativas beneficentes e ações sociais. Uma estimativa levantada pela CNN em 2022 mostra que existe 57 milhões de voluntários ativos em 2021, cerca de 26,47% da população brasileira.

No entanto, muitas dessas organizações enfrentam desafios significativos na gestão financeira, devido à utilização de métodos tradicionais e arcaicos de controle de caixa e contabilidade. Essas limitações podem comprometer a eficácia e eficiência das atividades beneficentes realizadas, resultando em perda de tempo e recursos valiosos. Nesse contexto, a implementação de soluções tecnológicas adaptadas às necessidades específicas dos grupos voluntários torna-se fundamental para otimizar a gestão financeira e operacional dessas organizações.

O objetivo deste trabalho é propor o desenvolvimento de um software financeiro personalizado, visando automatizar tarefas administrativas e proporcionar uma ferramenta eficiente e adaptada às demandas das organizações voluntárias, reforçando o compromisso com a transparência, prestação de contas e otimização de recursos.

2 REFERENCIAL TEÓRICO

2.1 Tecnologia de banco de dados e desenvolvimento

Segundo a Amazon (2023), um Banco de Dados é uma coleção de dados que são armazenados de forma eletrônica, que pode conter qualquer tipo de dados sejam eles palavra, números, imagens, vídeos e arquivos. E a Oracle (2020) define que um banco de dados é geralmente controlado por um sistema gerenciador de banco de dados (DBMS) e com ele você pode manipular os dados como inserir, atualizar ou deletar um dado.

Segundo a IBM (2024), PostgreSQL é um banco de dados de código aberto que possui uma forte reputação por sua confiabilidade, flexibilidade e suporte a padrões técnicos abertos. Ao contrário de outros Relational Database Management Systems (RDMBS), o PostgreSQL suporta tipos de dados não relacionais e relacionais. Isso o torna um dos bancos de dados relacionais mais compatíveis, estáveis e maduros disponíveis atualmente.

A Figura 1 demonstra como funciona o banco de dados relacional que utilizaremos no nosso projeto. No exemplo são descritas 3 tabelas que possuem chaves que são únicas de cada tabela, e através delas é possível conectar dados de uma tabela para a outra, nota-se que a tabela “clientes” está relacionada com a tabela “reservas”, através de sua chave-primária “CodCliente”, que ao ser inserida na tabela “reservas”, torna-se uma chave estrangeira que será responsável pela ligação.

Figura 1 – Exemplo de Banco de Dados Relacional



Fonte: Os autores, 2024. Adaptado de:

https://www.rmfaiss.com/rmfaiss/artigos/table.php?_codigo=6

O PostgreSQL suporta transações de atomicidade, consistência, isolamento, durabilidade (ACID), garantindo que as operações de banco de dados sejam executadas de forma confiável e segura. A Amazon (2024) afirma que o PostgreSQL garante que as transações sejam atômicas, consistentes, isoladas e duráveis (ACID). Ela promove altos níveis de consistência de dados. Como é um sistema de gerenciamento de banco de dados relacional, o PostgreSQL pode garantir que as transações sigam cada propriedade do ACID.

Além dos tipos de dados padrão, o PostgreSQL oferece suporte a tipos de dados complexos, como JSON, XML, arrays e tipos geométricos, permitindo a modelagem flexível de dados. A Amazon (2024) destaca que trabalhar com o PostgreSQL é mais fácil e intuitivo para os desenvolvedores de banco de dados. Ele também oferece suporte a outros tipos de dados adicionais, como matrizes, XML, JSON entre outros tipos.

Segundo a Microsoft (2024), o C# é uma linguagem de programação moderna, orientada a objeto e fortemente tipada, permitindo que os desenvolvedores criem vários tipos de aplicativos seguros e robustos que são executados no NET. O C# fornece construções de linguagem para dar suporte diretamente a esses conceitos, tornando C# uma linguagem natural para criação e uso de componentes de software.

A Figura 2 mostra o índice TIOBE, que classifica as linguagens de programação mais populares com base na demanda dos desenvolvedores, o C# é a quinta linguagem de programação mais usada no mundo, tendo 6,49% de utilização (TIOBE, 2024). Isso pode ser atribuído em parte à sua forte integração com o ecossistema da Microsoft, incluindo o framework .NET, que oferece uma ampla gama de bibliotecas e ferramentas para o desenvolvimento de software.

Figura 2 – Ranking de Linguagens

Maio 2024	Maio 2023	Alteração	Linguagem de Programação	Porcentagem de Uso	Alteração
1	1		Python	16.33%	+2.88%
2	2		C	9.98%	-3.37%
3	4	^	C++	9.53%	-2.43%
4	3	v	Java	8.69%	-3.53%
5	5		C#	6.49%	-0.94%
6	7	^	JavaScript	3.01%	+0.57%
7	6	v	Visual Basic	2.01%	-1.83%
8	12	^	Go	1.60%	+0.61%
9	9		SQL	1.44%	-0.03%
10	19	^	Fortran	1.24%	+0.46%

Fonte: Os autores, 2024. Adaptado de: <https://www.tiobe.com/tiobe-index/>

É uma linguagem orientada a objetos, o que significa que ela modela o mundo real através de objetos que têm propriedades e comportamentos. Segundo o site DevMedia (2012), A orientação a objetos é uma ponte entre o mundo real e virtual, é possível transcrever a forma como enxergamos os elementos do mundo real em código fonte como por exemplo pessoas, animais e carros, a fim de nos possibilitar a construção de sistemas complexos baseados em objetos.

É uma linguagem de tipagem estática, onde os tipos de dados das variáveis são verificados em tempo de compilação. Segundo a Microsoft (2024), C# é uma linguagem fortemente tipada. Todas as variáveis e constantes têm um tipo, assim como cada expressão que é avaliada como um valor.

2.2 Grupos voluntários e como controlar suas finanças

A contribuição dos grupos voluntários para a sociedade é multifacetada e significativa. De acordo com Smith (2010), o voluntariado desempenha um papel crucial na promoção da coesão social, no fortalecimento do capital social e na construção de comunidades mais resilientes e inclusivas.

Apesar de sua importância, os grupos voluntários enfrentam uma série de desafios que podem limitar sua eficácia e sustentabilidade. Segundo a ONU (2022) À medida que os países e regiões enfrentam enormes desafios, uma coisa é clara nenhuma entidade ou setor pode enfrentar estes desafios sozinhos. Agora, mais do que nunca, as parcerias são de vital importância.

Questões como a escassez de recursos financeiros e humanos, a falta de reconhecimento institucional e os desafios de governança podem representar obstáculos significativos para essas organizações (Handy, Cnaan, & Brudney; 2000). No entanto, esses desafios também oferecem oportunidades para a inovação, a colaboração e o fortalecimento do setor voluntário.

Segundo o Sebrae (2022), os controles financeiros são de grande importância para as decisões empresariais. Fazer o acompanhamento das finanças é fundamental para o dia a dia da empresa. É possível dizer que as informações

geradas com os controles representam o primeiro estágio para a gestão do capital de giro da empresa. Nas empresas de pequeno porte, quando se consegue administrar o capital de giro de maneira eficiente resolve-se basicamente a maioria dos problemas de natureza financeira (Sebrae, 2022).

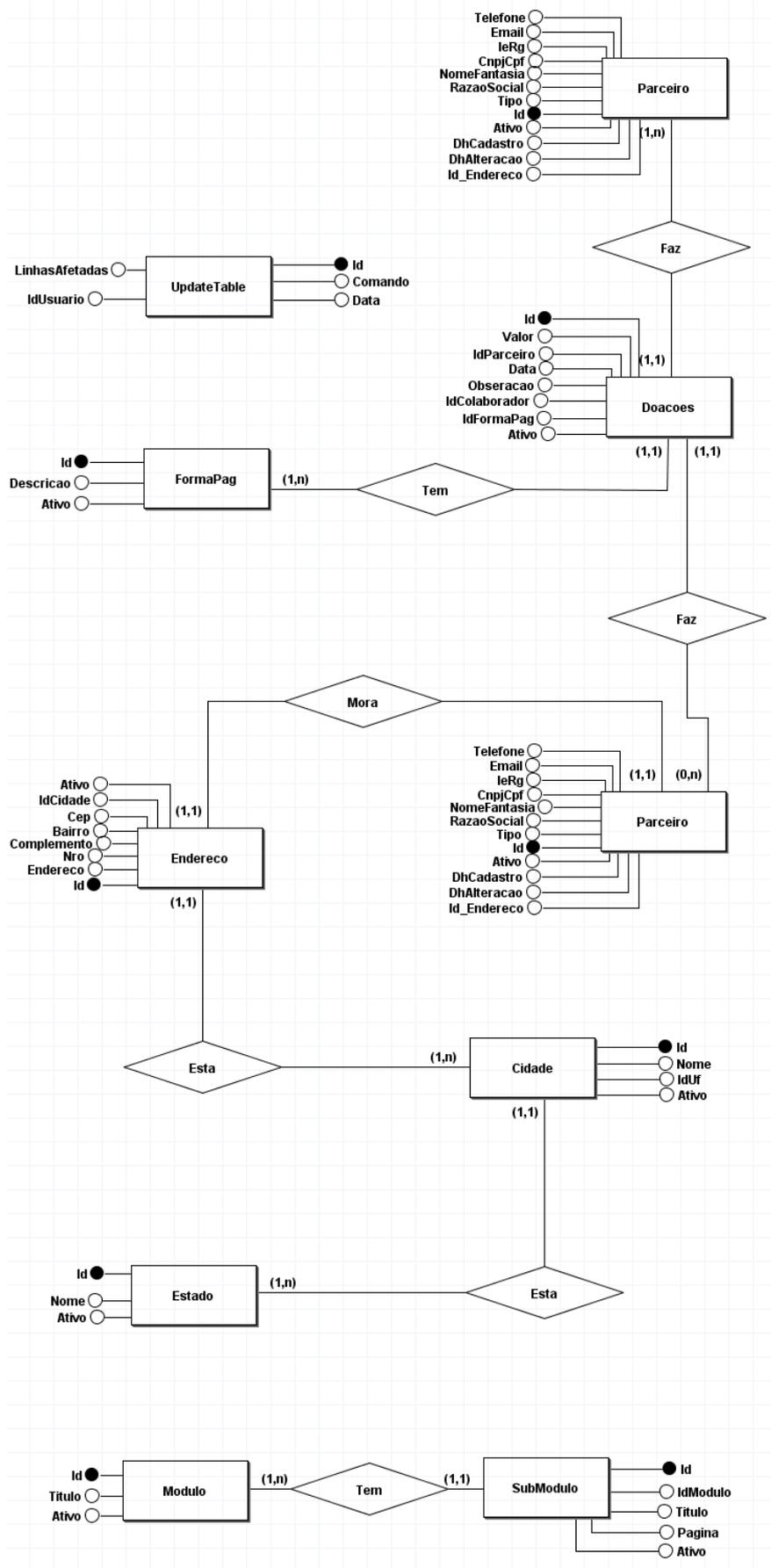
O controle de custos envolve o monitoramento e gerenciamento dos custos operacionais e despesas de uma entidade. Horngren, Sundem e Stratton (2005) afirmam que o controle de custos é essencial para garantir a eficiência operacional e a rentabilidade de uma organização, permitindo a identificação de áreas de desperdício e oportunidades de economia.

3 MATERIAIS E MÉTODOS

Para iniciar as atividades no banco primeiro foi preciso fazer a instalação do SGBD do postgres o PGAdmin, disponível gratuitamente na internet. O pgAdmin é uma ferramenta de administração e desenvolvimento de banco de dados para o PostgreSQL, um sistema de gerenciamento de banco de dados relacional de código aberto. O pgAdmin é projetado para fornecer uma interface gráfica intuitiva e abrangente para administrar bancos de dados PostgreSQL.

Na Figura 3 temos o modelo de Banco de Dados, mostrando como funciona as conexões entre as tabelas e todas as informações que serão guardadas, o banco de dados gerará um novo registro para cada tabela, cujas chaves são geradas automaticamente com o recurso de auto-increment. Além de que nenhuma informação será excluída do banco de dados apenas deixara de estar ativa, e teremos controle de todos os dados íntegros no banco. De tabelas temos a tabela de “Doações” que será uma das principais tabelas do nosso sistema ela é interligada com a tabela “FormaPag” nela é gravada os tipos de forma de pagamentos são recebidos pelo grupo voluntário e “Parceiros” que irá armazenar os dados de quem está fazendo a doação, e nela é interligada a tabela de endereço que armazena onde é a residência do doador. Temos também nossa tabela de “Modulo” e “SubModulo” elas são tabelas feitas para armazenar os dados de qual pagina irá ser exibida na tela do sistema. E por último temos a tabela de “UpdateTable” que nela será armazenado todos os comandos feitos no banco de dados.

Figura 3 – Exemplo modelo do Banco de Dados



Fonte: Os autores, 2024

Para a construção do sistema e seu design foi utilizado à linguagem de programação C#. Para criação do projeto foi utilizado o Visual Studio Community 2022

O Visual Studio possui uma extensão gratuita completa para desenvolvedores que inclui as melhores ferramentas que no caso foi utilizado para criação do projeto tanto na parte de design quanto na programação e conexão com o banco de dados.

A Figura 4 mostrar a tela de doação, ela é uma das telas que mais serão usadas no sistema, foi projetada para ter um design limpo, ser intuitiva e básica para os usuários. Assim a tela foi cuidadosamente concebida para simplificar o processo de contribuição garantido que os usuários não percam tanto tempo e posso forçar em seus outros compromissos, na figura 4 o usuário terá que informa o doador, a forma de pagamento que foi feita a doação, o valor feito, a data que foi feita além de alguma observação se assim necessário, com essas informações colocadas ao clicar em salvar o sistema ira gravar as informações no banco de dados através da *function* criada.

Figura 4 – Tela de doação

A imagem mostra uma janela de software intitulada "Cadastro de Doações". O formulário contém os seguintes campos:

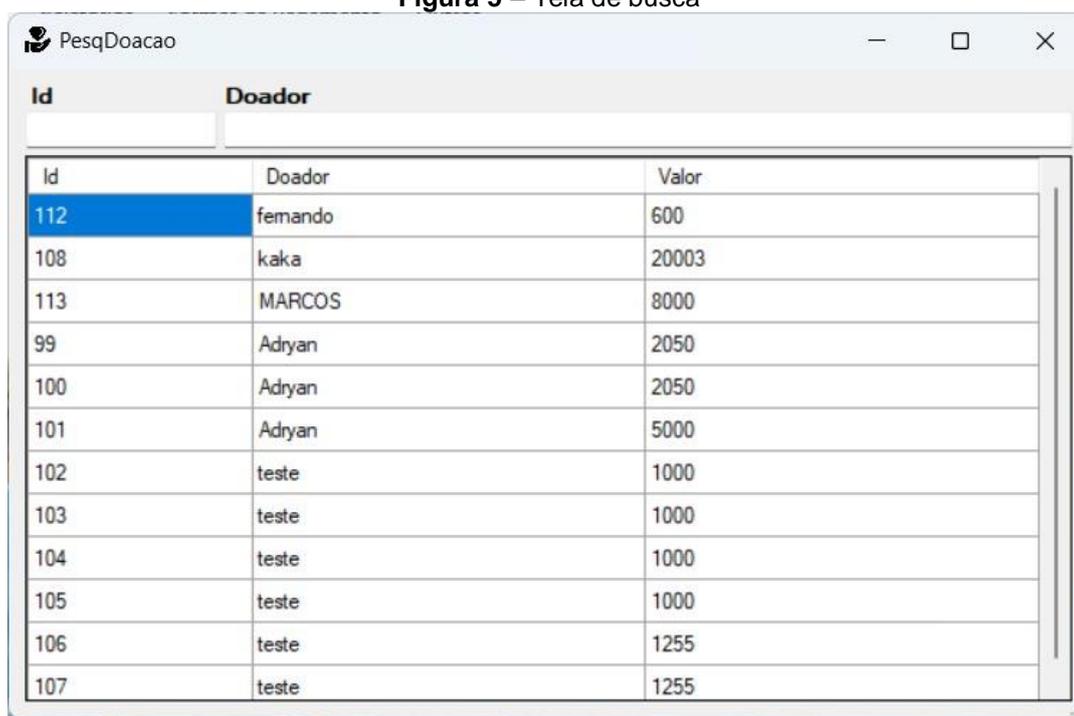
- Principal**:
 - Nome Doador**: Campo de texto para o nome do doador.
 - Forma de Pagamento**: Menu suspenso para selecionar o método de pagamento.
 - Data da Doação**: Campo de data com o valor "27/10/2024" e ícone de calendário.
 - Valor**: Campo de texto com o valor "0".
 - Observação**: Campo de texto para notas adicionais.

Na base da janela, há dois botões: "Cancelar" (com um ícone de X vermelho) e "Salvar" (com um ícone de checkmark verde).

Fonte: Os autores, 2024

A Figura 5 demonstra a tela de pesquisa de doador ela é usada para buscar um doador já cadastrado anteriormente no sistema para o cadastro na tela de doação. Para sua exibição é feita uma busca no em uma das views criadas no banco de dados, além de conter filtros para facilitar a pesquisa do usuário.

Figura 5 – Tela de busca



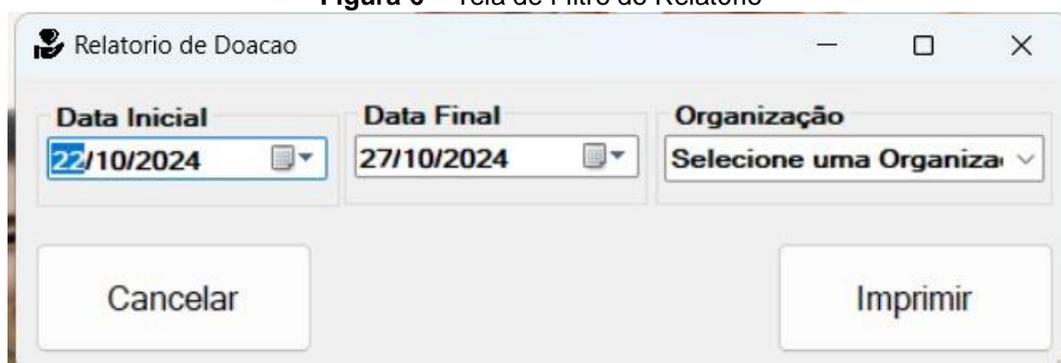
The screenshot shows a window titled "PesqDoacao" with a search bar and a table of results. The table has three columns: "Id", "Doador", and "Valor". The first row is highlighted in blue.

Id	Doador	Valor
112	femando	600
108	kaka	20003
113	MARCOS	8000
99	Adryan	2050
100	Adryan	2050
101	Adryan	5000
102	teste	1000
103	teste	1000
104	teste	1000
105	teste	1000
106	teste	1255
107	teste	1255

Fonte: Os autores, 2024

As Figuras 6 é a tela de filtro para o usuário gerar o relatório ele adiciona o período que o relatório irá ser exibido além de também pode filtrar as informações pelo usuário ao clicar em imprimir ele reune essas informações e adiciona ao filtro SQL para gerar o relatório.

Figura 6 – Tela de Filtro do Relatório



The screenshot shows a window titled "Relatorio de Doacao" with three input fields: "Data Inicial" (22/10/2024), "Data Final" (27/10/2024), and "Organização" (Selecione uma Organiza). There are "Cancelar" and "Imprimir" buttons at the bottom.

Fonte: Os autores, 2024

Na Figura 7 temos como exemplo um dos nossos relatórios que é feito em CSV ele traz informações como id da doação, data da doação, valor recebido, nome do parceiro e a forma de pagamento, que é gerado através de uma pesquisa feita no banco de dados em uma das views criadas por nós.

Figura 7 – Relatório em CSV

	A	B	C	D	E
1	Id	Data	Valor	Parceiro	Pagamento
2	124	01/11/2024 15:36	444,74	JOÃO DE BARROS	PIX
3	114	02/11/2024 15:31	4000	MARCOS TEIXEIRA	PIX
4	115	03/11/2024 15:31	2040	MARCOS SILVA	DINHEIRO
5	116	03/11/2024 15:32	2040,39	GABRIEL AUGUSTO	DINHEIRO
6	117	03/11/2024 15:33	2770,39	RODRIGO FERREIRA	DINHEIRO
7	118	03/11/2024 15:33	2770,39	AUGUSTO PINTO	DINHEIRO
8	119	03/11/2024 15:34	2770,39	LORENA NAVARRO	DINHEIRO
9	120	03/11/2024 15:34	2770,39	FREDERICO MACHADO	PIX
10	125	04/11/2024 15:37	456,74	LUIZA FIGUEIREDO	PIX
11	121	05/11/2024 15:35	620,79	CARLOS	PIX
12	128	05/11/2024 15:39	24714	JULIA LARISSA	PIX
13	123	06/11/2024 15:36	94,74	CATARINIA DA SILVA	DINHEIRO
14	126	07/11/2024 15:37	116,11	VANESSA OLIVEIRA	DINHEIRO
15	127	07/11/2024 15:37	198,88	SANDRA SILVA	DINHEIRO
16	122	13/11/2024 15:35	180,74	PAULO NEGRINE	PIX

Fonte: Os autores, 2024

A Figura 8 mostra a criação da tabela de doação umas das tabelas que terá o maior número de movimentações ela segue um padrão criação SQL com primeira letra sempre maiúscula além da criação da chave primaria sendo ela a coluna de nome id, e chave estrangeira que serão as colunas que liga essa tabela a outras tabelas. Ela também segue outro padrão criado por nós que é a adição da coluna ativo com isso não precisaremos deletar nenhuma informação do banco apenas deixa-la inativa. Contem também a criação de duas *foreign keys* que servem para fazer ligação entre a tabela demonstrada abaixo com outras 2 tabelas do sistema sendo elas “FormaPag” e “Parceiro”.

Figura 8 – Criação de tabela

```
CREATE TABLE public."Doacoes"  
(  
    "Id" integer NOT NULL DEFAULT nextval('doacao_id_seq'::regclass),  
    "Valor" double precision NOT NULL,  
    "IdParceiro" integer,  
    "Data" time with time zone,  
    "Observacao" character varying NOT NULL,  
    "IdColaborador" integer,  
    "IdFormaPag" integer NOT NULL,  
    "Ativo" boolean DEFAULT true,  
    CONSTRAINT "Doacao_PK" PRIMARY KEY ("Id"),  
    CONSTRAINT "FormaPag_FK" FOREIGN KEY ("IdFormaPag")  
        REFERENCES public."Doacoes" ("Id") MATCH SIMPLE  
        ON UPDATE NO ACTION ON DELETE NO ACTION,  
    CONSTRAINT "Parceiro_FK" FOREIGN KEY ("IdParceiro")  
        REFERENCES public."Parceiro" ("Id") MATCH SIMPLE  
        ON UPDATE NO ACTION ON DELETE NO ACTION  
)
```

Fonte: Os autores, 2024

A Figura 9 mostra a criação da tabela “*UpdateTable*” outra tabela importante para nosso projeto pois ela irá armazenar todas as operações feitas dentro do nosso banco de dados, com isso ela será nossa tabela de segurança no bando que irá nos mostrar de maneira fácil e eficiente as modificações feitas nos mostrando o usuário que fez a alteração, a data de modificação, o que foi alterado além de quantas linhas foram afetadas.

Figura 9 – Tabela de controle

```
CREATE TABLE public."UpdateTable"
(
    "Id" integer NOT NULL DEFAULT nextval('uptables_id_seq'::regclass),
    "Comando" text,
    "Data" timestamp without time zone,
    "IdUsuario" integer,
    "LinhasAfetadas" integer,
    CONSTRAINT uptables_pkey PRIMARY KEY ("Id")
)
```

Fonte: Os autores, 2024

Na Figura 10 exibe a *function* que é usado dentro da *trigger* nela pegamos o número de linhas afetadas pelo comando, após essa ação é feita uma verificação para ver se o comando foi um insert ou um update, ao validar é criado um script que é feito com as novas informações fornecidas no sistema, depois dessa operação é feita a inserção na tabela “*UpdateTable*” o comando dado no banco de dados, o número de linhas afetadas e a data e hora no momento que foi ativada a trigger, além de demonstrar a criação da *trigger* “*trigger_log_update_doacoes*” usando a *function* criada.

Figura 10 – Trigger de Controle

```
CREATE OR REPLACE FUNCTION public.log_insert_update_doacoes()
RETURNS TRIGGER AS $$
DECLARE
    v_numero_linhas_afetadas INTEGER;
    v_comando TEXT;
BEGIN
    -- Obter o número de linhas afetadas pela operação
    GET DIAGNOSTICS v_numero_linhas_afetadas = ROW_COUNT;

    -- Verificar se é um INSERT ou UPDATE
    IF TG_OP = 'INSERT' THEN
        v_comando := 'INSERT INTO public."Doacoes"("Valor",
        "IdParceiro", "Data", "Observacao", "IdColaborador", "IdFormaPag",
        "Ativo") VALUES (' ||
        quote_literal(NEW."Valor") || ', ' ||
        quote_literal(NEW."IdParceiro") || ', ' ||
        quote_literal(NEW."Data") || ', ' ||
        quote_literal(NEW."Observacao") || ', ' ||
        quote_literal(NEW."IdColaborador") || ', ' ||
        quote_literal(NEW."IdFormaPag") || ', ' ||
        quote_literal(NEW."Ativo") || ');';
    ELSIF TG_OP = 'UPDATE' THEN
        v_comando := 'UPDATE public."Doacoes" SET ' ||
        '"Valor" = ' || quote_literal(NEW."Valor") || ', ' ||
        '"IdParceiro" = ' || quote_literal(NEW."IdParceiro") || ', ' ||
        '"Data" = ' || quote_literal(NEW."Data") || ', ' ||
```

```

        "Observacao" = ' || quote_literal(NEW."Observacao") || ', ' ||
        "IdColaborador" = ' || quote_literal(NEW."IdColaborador") ||
        ', ' ||
        "IdFormaPag" = ' || quote_literal(NEW."IdFormaPag") || ', ' ||
        "Ativo" = ' || quote_literal(NEW."Ativo") ||
        ' WHERE "Id" = ' || quote_literal(OLD."Id") || ';';
END IF;
-- Inserir os dados na tabela UpdateTable
INSERT INTO public."UpdateTable"( "Comando", "Data", "LinhasAfetadas
")
VALUES ( v_comando, -- O comando que foi executado (INSERT ou UPDATE)
        now(), -- Data e hora da operação
        v_numero_linhas_afetadas -- Número de linhas afetadas
);
-- Continuar com a operação (INSERT ou UPDATE)
RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trigger_log_update_doacoes
AFTER UPDATE ON public."Doacoes"
FOR EACH ROW
EXECUTE FUNCTION public.log_insert_update_doacoes();

```

Fonte: Os autores, 2024

4 CONSIDERAÇÕES FINAIS

O trabalho foi finalizado com um resultado satisfatório por parte dos autores do projeto, onde foi criado um modelo de banco de dados que atende à proposta do trabalho de se ter um controle financeiro. O sistema foi desenvolvido para otimizar o processo administrativo dessas entidades, facilitando o controle de doações e de despesas. A eficácia da aplicação foi comprovada pelas telas implementadas e demonstradas, evidenciando sua funcionalidade intuitiva e a capacidade de automatizar o fluxo financeiro de maneira organizada e acessível.

Como trabalhos futuros, poderia ser considerada a expansão do sistema para incluir novas funcionalidades avançadas, como análise de dados para relatórios financeiros mais detalhados, e uma interface de uso simplificada para dispositivos móveis, visando atender às necessidades crescentes das ONGs e melhorar ainda mais a transparência e a eficiência na gestão dos recursos.

REFERÊNCIAS

AMAZON. **O que é um banco de dados ?**. 2023. Disponível em: <https://aws.amazon.com/pt/what-is/database/>

AMAZON. **Qual a diferença entre o MongoDB e o PostgreSQL?**. 2024. Disponível em: <https://aws.amazon.com/pt/compare/the-difference-between-mongodb-and-postgresql/>

CNN. **Em 2021, 57 milhões de brasileiros fizeram trabalho voluntário, diz pesquisa**. 2022. Disponível em: <https://www.cnnbrasil.com.br/nacional/em-2021-57-milhoes-de-brasileiros-fizeram-trabalho-voluntario-diz-pesquisa/>

DEVMEDIA. **C# Orientado a Objetos: Introdução**. 2013. Disponível em: <https://www.devmedia.com.br/csharp-orientado-a-objetos-introducao/29539>

Handy, F., Cnaan, R. A., & Brudney, J. L. (2000). **Asymmetrical Influence: A Study of Voluntary Association Dependence on Government**. *Nonprofit and Voluntary Sector Quarterly*, 29(3), 378-398. Disponível em: <https://journals.sagepub.com/doi/abs/10.1177/089976400773746300>

IBM. **O que é PostgreSQL?**. 2024. Disponível em: <https://www.ibm.com/br-pt/topics/postgresql>

MICROSOFT. **Documentação do C#**. 2024. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/>

MICROSOFT. **O sistema do tipo C#**. 2024. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/csharp/fundamentals/types/>

ORACLE. **O que é um banco de dados?**. 2020. Disponível em: <https://www.oracle.com/br/database/what-is-database/>

SEBRAE. **Controle Financeiros são essenciais para a gestão do capital de giro**. 2022. Disponível em: <https://sebrae.com.br/sites/PortalSebrae/artigos/controles-financeiros-sao-essenciais-para-a-gestao-do-capital-de-giro,df395415e6433410VgnVCM1000003b74010aRCRD>

Smith, D. H. (2010). **Understanding Volunteerism: A Theoretical Framework**. *Nonprofit and Voluntary Sector Quarterly*, 39(2), 153-160. Disponível em: <https://journals.sagepub.com/doi/abs/10.1177/0899764009358305>

TIOBE. **TIOBE Index for 2024**. 2024. Disponível em: <https://www.tiobe.com/tiobe-index/>

IBM. **O que é PostgreSQL?**. 2024. Disponível em: <https://www.ibm.com/br-pt/topics/postgresql>

UN Volunteers. **Building Equal and Inclusive Societies**. 2022. Disponível em: <https://swvr2022.unv.org/>