

UNifeob
| ESCOLA DE NEGÓCIOS



2024

PROJETO INTEGRADO



UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS

ESCOLA DE NEGÓCIOS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

CIÊNCIA DA COMPUTAÇÃO

PROJETO INTEGRADO

SISTEMA DE GESTÃO E INTELIGÊNCIA DE NEGÓCIOS
PARA ORGANIZAÇÕES SOCIAIS

Tech Solutions

SÃO JOÃO DA BOA VISTA, SP

NOVEMBRO 2024

UNIFEOB
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS
ESCOLA DE NEGÓCIOS
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
CIÊNCIA DA COMPUTAÇÃO

PROJETO INTEGRADO
SISTEMA DE GESTÃO E INTELIGÊNCIA DE NEGÓCIOS
PARA ORGANIZAÇÕES SOCIAIS

Tech Solutions

MÓDULO COMPUTAÇÃO EM NUVEM

Estrutura de Dados – Prof. Marcelo Ciacco Almeida

Linguagem e Técnicas de Programação – Prof. Nivaldo de Andrade

Tópicos Avançados de Banco de Dados – Prof. Max Streicher Vallim

Computação em Nuvem – Prof. Rodrigo Marudi de Oliveira

Projeto de Computação em Nuvem – Prof^a. Mariângela Martimbianco Santos

Estudantes:

André de Lima Miranda Borges, RA 23000737

Carlos Miguel de Souza Costa, RA 23000844

João Rafael Lange Domingos, RA 23000497

Leticia Stanguini da Silva, RA 23000843

Rafael Alves Monteiro, RA 24002048

Pedro Volpe Mariano, RA 23000815

SÃO JOÃO DA BOA VISTA, SP
NOVEMBRO 2024

1. INTRODUÇÃO	4
2. DESCRIÇÃO DA EMPRESA	5
3. PROJETO INTEGRADO	6
3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS	6
3.1.1 MODELO LÓGICO	6
3.1.2 MODELO FÍSICO	7
3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO	10
3.2.1 APPLICATION PROGRAMMING INTERFACE (API) - BACK-END.	10
3.2.2 FRONT-END	11
3.3 COMPUTAÇÃO EM NUVEM	12
3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING	12
3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO	12
3.3.3 VANTAGENS DA CLOUD COMPUTING	13
3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING	14
3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)	14
3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING	15
3.3.7 GOOGLE CLOUD ou AWS	16
3.4 ESTRUTURA DE DADOS	17
3.4.1 LEVANTAMENTO DE REQUISITOS	18
3.4.2 VALIDAÇÃO DOS REQUISITOS	19
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS	20
3.5.1 ENFRENTANDO ESTEREÓTIPOS	20
3.5.2 ESTUDANTES NA PRÁTICA	21
4. CONCLUSÃO	23
REFERÊNCIAS	24
ANEXOS	25

1. INTRODUÇÃO

Com a crescente importância das tecnologias digitais no apoio às causas sociais, apresentamos o protótipo do site de doações para animais de estimação da nossa ONG. Reconhecemos a necessidade de uma plataforma online eficaz para arrecadar fundos e apoiar os pets em situação de vulnerabilidade. Este trabalho representa não apenas a concretização de um projeto técnico, mas também um compromisso tangível com a nossa missão de proteger e cuidar dos nossos amigos de quatro patas. Iremos destacar a importância do nosso site de doações, suas principais características e funcionalidades, e o impacto positivo que esperamos alcançar na comunidade.

As ONGs de pets enfrentam desafios na gestão eficiente de doações, essenciais para a manutenção de suas atividades e cuidado dos animais. Este projeto visa desenvolver um sistema de gerenciamento de doações para otimizar a captação e administração.

O objetivo é proporcionar um controle preciso e transparente dos recursos recebidos, facilitando o planejamento estratégico e a comunicação com os doadores. O sistema incluirá funcionalidades como registro e acompanhamento de doações, geração de relatórios financeiros, gestão de campanhas de arrecadação e emissão de recibos personalizados.

Com uma abordagem centrada no usuário, o projeto busca atender às necessidades da ONG e dos doadores, fortalecendo a confiança e o engajamento da comunidade. A implementação deste gerenciador de doações permitirá à ONG maximizar seu impacto social e continuar a cuidar dos animais de forma eficaz.

2. DESCRIÇÃO DA EMPRESA

A Tech Solutions opera em Unidade São João da Boa Vista: Rua José Bonifácio, 123 - Centro - São João da Boa Vista - SP, 13860-000 (Campus da UNIFEOB) portadora do CNPJ: 52.084.215/0001-31, visa o mercado de tecnologia da informação e comunicação (TIC), com foco no desenvolvimento de softwares personalizados para empresas e órgãos públicos. Atua em diversos segmentos, como:

- Indústria: Automação de processos, controle de produção, gestão de estoque.
- Comércio: Sistemas de vendas, controle de estoque, e-commerce.
- Serviços: Gestão de clientes, controle de financeiro, gestão de projetos.
- Setor Público: Saúde, educação, segurança pública.

Para desenvolver um Plano de Implementação (PI) eficaz e verdadeiramente impactante para a ONG em questão, nossa equipe partiu de um princípio fundamental: a compreensão profunda das necessidades e desafios que a organização enfrenta. Essa busca por conhecimento não se limitou a pesquisas superficiais ou análises distantes, mas sim a um processo colaborativo e empático, onde a voz da ONG se tornou o centro de nossa investigação.

O desenvolvimento de um Plano de Implementação (PI) eficaz para a Tech Solutions exigiu uma metodologia abrangente e colaborativa, ancorada em diversas fontes de pesquisa e ferramentas de análise. A combinação de reuniões presenciais, mapas de detalhamento e o modelo Figma nos permitiu reunir informações valiosas, identificar necessidades e desafios, e traçar um plano de ação estratégico para o sucesso da empresa.

3. PROJETO INTEGRADO

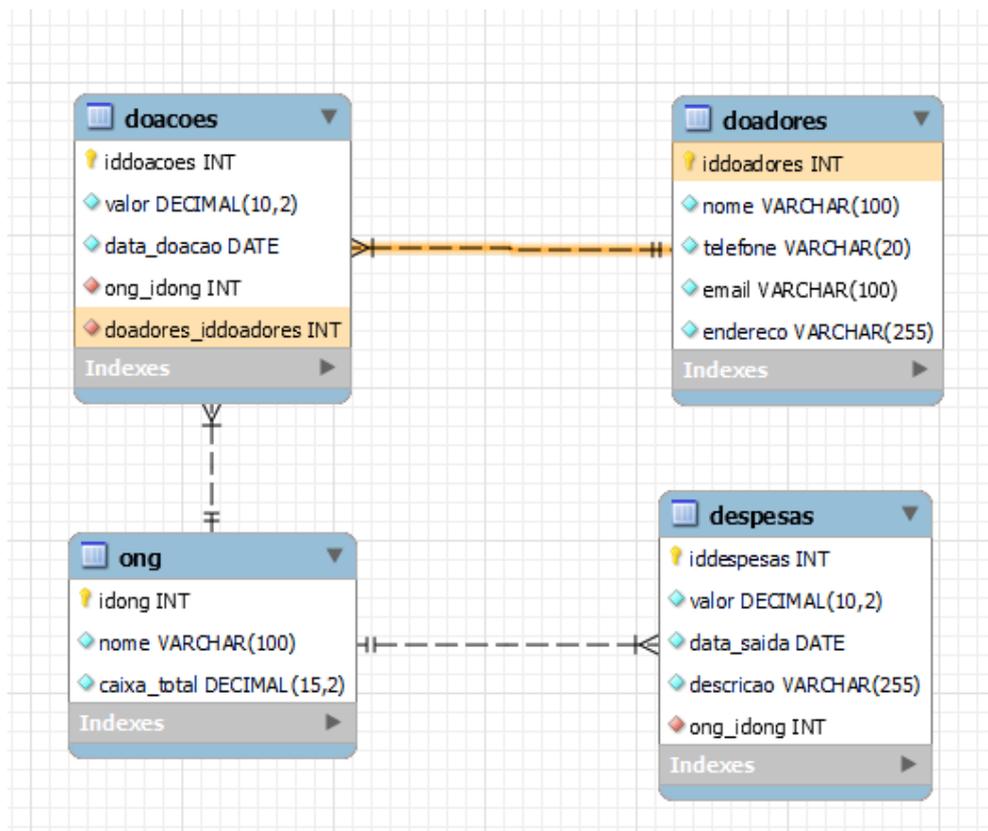
3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS

O Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido para o projeto foi o MySQL Workbench, devido à sua facilidade de uso, interface intuitiva e recursos que facilitam o gerenciamento e a administração do banco de dados. Além disso, a experiência do usuário proporcionada por essa ferramenta contribui para um desenvolvimento mais ágil e eficiente. O banco de dados será hospedado na nuvem, utilizando os servidores da AWS, com o serviço Amazon S3 para o armazenamento seguro e escalável dos dados.

3.1.1 MODELO LÓGICO

Conforme destacado por Leyendecker, “no nível lógico, a modelagem é mais detalhada: as tabelas são criadas e as colunas, tipos de dados, chaves primárias e estrangeiras e restrições são definidas.” Com isso em mente, iniciamos a criação do nosso modelo lógico, definindo as tabelas e estabelecendo as relações entre elas. Esse processo permite estruturar o banco de dados de forma precisa, garantindo a integridade e coerência dos dados através da definição de chaves e restrições adequadas para cada tabela.

Figura 1: Tabelas do modelo lógico



Fonte:Autores

Na elaboração das tabelas, foi necessário definir as relações entre elas para garantir um fluxo de dados consistente no banco de dados.

A tabela Doadores está ligada à tabela Doações em uma relação 1 para N, o que significa que um doador pode fazer várias doações, mas cada doação está associada a apenas um doador. Já a tabela Doações está conectada à tabela ONGs em uma relação N para 1, pois várias doações podem ser direcionadas a uma única ONG (neste sistema, cada doação está associada a apenas uma ONG). Por fim, a tabela ONGs está relacionada à tabela Despesas em uma relação 1 para N, uma vez que uma ONG pode ter várias despesas.

Essas relações permitem estruturar o fluxo de dados no sistema, garantindo que cada doador possa ter múltiplas doações associadas, que cada doação esteja vinculada a uma única ONG, e que cada ONG possa ter várias despesas associadas.

3.1.2 MODELO FÍSICO

De acordo com Nascimento, "o modelo físico traduz essa visão em termos concretos de implementação técnica, garantindo que os requisitos de negócios sejam atendidos de maneira eficiente." Com base nesse princípio, optamos pelo MySQL Workbench como Sistema de Gestão de Banco de Dados (SGBD), uma ferramenta que facilita a execução das tarefas de desenvolvimento e manutenção do banco de dados.

A seguir, realizamos a criação das tabelas essenciais para o sistema, incluindo as tabelas ONG, doadores, doações e despesas. Em cada uma dessas tabelas, definimos os campos e os tipos de dados apropriados, assegurando que todos os requisitos de dados fossem atendidos com precisão.

Um exemplo específico de implementação foi na tabela doações, onde utilizamos o tipo de dado ENUM para definir os diferentes métodos de pagamento. Isso garantiu uma forma clara e eficiente de armazenar os valores possíveis para essa categoria, como ilustrado na imagem abaixo.

Figura 2: Script da criação de tabelas

```
CREATE TABLE doacoes (  
  id_doacao INT AUTO_INCREMENT PRIMARY KEY,  
  id_doador INT,  
  valor DECIMAL(10, 2) NOT NULL,  
  data_doacao DATE NOT NULL,  
  metodo_pagamento ENUM('dinheiro', 'cartao', 'transferencia', 'outro'),  
  FOREIGN KEY (id_doador) REFERENCES doadores(id_doador)  
);
```

Fonte:Autores

Magalhães define que um “trigger ou gatilho, em sistemas de gerenciamento de banco de dados (SGBDs), é um procedimento automático executado ("disparado") em resposta a eventos específicos em uma tabela ou view ". Com base nesse conceito, implementamos um

gatilho em nosso sistema com o objetivo de automatizar uma tarefa importante: a atualização do caixa total da empresa sempre que uma nova doação é registrada.

Dessa forma, o gatilho garante que o valor do caixa esteja sempre atualizado, sem a necessidade de realizar cálculos manuais a cada doação. Essa automação proporciona maior eficiência e precisão na gestão financeira, mantendo o controle do caixa sempre em tempo real.

Figura 3: Script da criação da trigger

```
CREATE TRIGGER atualiza_caixa_apos_doacao
AFTER INSERT ON doacoes
FOR EACH ROW
BEGIN
    UPDATE ong SET caixa_total = caixa_total + NEW.valor WHERE id_ong = 1;
END$$
```

Fonte:Autores

Segundo Monteiro, Stored Procedure são "comandos T-SQL que suportam parâmetros de entrada e saída, e que podem ou não retornar resultados". Com base nesse conceito, utilizamos Stored Procedures em nosso projeto para gerenciar as doações realizadas em dinheiro. Como não é possível realizar uma doação em dinheiro de forma virtual, a Stored Procedure foi implementada para permitir que o usuário da ONG registre a doação em espécie, ao mesmo tempo em que atualiza automaticamente o caixa total da organização.

Essa abordagem assegura que tanto o registro da doação quanto a atualização do caixa sejam feitos de maneira eficiente e sem a necessidade de intervenções manuais, garantindo maior precisão na gestão financeira da ONG.

Figura 4: Script da criação da Procedure

```
CREATE PROCEDURE registrar_doacao(
    IN p_id_doador INT,
    IN p_valor DECIMAL(10, 2),
    IN p_data_doacao DATE,
    IN p_metodo_pagamento ENUM('dinheiro', 'cartao', 'transferencia', 'outro')
)
BEGIN
    -- Inserir a doação na tabela doacoes
    INSERT INTO doacoes (id_doador, valor, data_doacao, metodo_pagamento)
    VALUES (p_id_doador, p_valor, p_data_doacao, p_metodo_pagamento);

    -- Atualizar o caixa total da ONG
    UPDATE ong SET caixa_total = caixa_total + p_valor WHERE id_ong = 1;
END$$
```

Fonte:Autores

Além disso, foi desenvolvida uma Stored Procedure para a administração das saídas de dinheiro, uma vez que a ONG precisa registrar e controlar suas despesas, descontando os valores do caixa total. Essa Stored Procedure permite que o processo de contabilização das saídas seja realizado de forma automatizada e precisa, garantindo que o saldo do caixa seja ajustado sempre que uma despesa for registrada, sem a necessidade de intervenções manuais.

Figura 5: Exibição da procedure

```
CREATE PROCEDURE registrar_saida(  
    IN p_valor DECIMAL(10, 2),  
    IN p_data_saida DATE,  
    IN p_descricao VARCHAR(255)  
)  
BEGIN  
    -- Verificar se há saldo suficiente no caixa  
    DECLARE caixa_atual DECIMAL(15, 2);  
  
    -- Obter o valor atual do caixa  
    SELECT caixa_total INTO caixa_atual FROM ong WHERE id_ong = 1;  
  
    -- Verificar se o valor da saída não é maior que o caixa atual  
    IF caixa_atual >= p_valor THEN  
        -- Registrar a saída de dinheiro (despesa) em uma nova tabela de despesas  
        INSERT INTO despesas (valor, data_saida, descricao)  
        VALUES (p_valor, p_data_saida, p_descricao);  
  
        -- Subtrair o valor da saída do caixa total da ONG  
        UPDATE ong SET caixa_total = caixa_total - p_valor WHERE id_ong = 1;  
    ELSE  
        -- Se não houver saldo suficiente, lançar um erro  
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Saldo insuficiente no caixa.';  
    END IF;  
END$$
```

Fonte: Autores

Como mencionado anteriormente, a Stored Procedure solicita o valor da saída, a data e uma descrição detalhada sobre o destino do gasto. Em seguida, ela verifica se a ONG possui saldo suficiente no caixa para realizar o "saque". Caso o saldo seja insuficiente, a operação é cancelada e um erro é retornado à ONG, com a mensagem 'Saldo insuficiente no caixa'.

Por outro lado, quando há saldo suficiente, a Stored Procedure valida a transação, autoriza a saída do valor e registra automaticamente os dados da operação, mantendo o caixa total atualizado de maneira precisa.

3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO

3.2.1 APPLICATION PROGRAMMING INTERFACE (API) - BACK-END.

Para o desenvolvimento da API do "Gerenciador de Doações para ONG de Pets", utilizamos a arquitetura RESTful, que permite criar APIs escaláveis e de fácil manutenção. Isso facilita a organização dos dados de doadores, pets e histórico de doações, além de simplificar a criação de endpoints para adicionar, consultar, atualizar e remover informações. Um ponto crítico é a implementação de autenticação e autorização, para garantir que apenas usuários autorizados possam acessar e manipular dados sensíveis. O uso de tokens JWT (JSON Web Token) possibilita uma segurança adicional, controlando o acesso com base no nível de permissão de cada usuário.

Além disso, integramos um módulo de relatórios que fornece dados analíticos em tempo real sobre doações e necessidades dos pets, auxiliando a ONG na tomada de decisões estratégicas. Isso permite que a equipe visualize as principais métricas de desempenho e identifique rapidamente as áreas que precisam de mais recursos. Para assegurar a evolução contínua do sistema, adotamos boas práticas, como a implementação de testes automatizados para os principais endpoints da API e o uso de versionamento, permitindo atualizações graduais sem afetar os usuários que ainda utilizam versões anteriores. Como Richardson (2018, p. 102) afirma, “a lógica de programação no backend é o núcleo que processa e estrutura as informações, garantindo que cada dado seja tratado corretamente para suportar a experiência do usuário e a integridade do sistema”, princípio que seguimos ao desenvolver as funcionalidades de nossa API.

3.2.2 FRONT-END

No front-end do sistema, o desenvolvimento começa com a criação de um design no Figma, permitindo que o layout seja validado antes de ser implementado em Angular, que facilita o desenvolvimento de uma interface dinâmica e de alto desempenho integrada ao back-end em Node.js. Aplicamos princípios de design responsivo, garantindo que o sistema seja acessível em computadores, oferecendo uma experiência de usuário (UX) intuitiva e uma interface (UI) atraente e fácil de navegar. Isso é especialmente importante para facilitar o acesso dos doadores e voluntários que desejam acompanhar as doações e visualizar as necessidades dos pets.

A interface inclui dashboards interativos que mostram o status atual das doações e necessidades específicas dos pets em tempo real, dando suporte para que a ONG possa compartilhar informações com os doadores e incentivar novas contribuições de maneira clara e visualmente atrativa. De acordo com Pilgrim (2011, p. 58), “desenvolver frontend não é apenas sobre aparência visual; é também sobre a estrutura e a organização do código para que seja sustentável e escalável”, o que reforça a importância de um código bem planejado e organizado.

Por fim, realizamos testes de usabilidade e acessibilidade para garantir que o sistema atenda às necessidades de todos os usuários, assegurando que ele funcione de forma correta e inclusiva.

3.3 COMPUTAÇÃO EM NUVEM

3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING

Os provedores de nuvem investem pesadamente em segurança, oferecendo soluções e serviços especializados para proteger seus dados. A automação de tarefas de segurança, como a aplicação de patches e a detecção de ameaças, pode reduzir o risco de erros humanos. Com isto em mente o projeto foi pensado nos detalhes mínimos para que possamos passar uma maior segurança aos doadores conforme o uso da plataforma. Além que também que ele está sendo hospedado em São Paulo para impedir qualquer "demora" na confirmação ou até mesmo no próprio pagamento.

Com o objetivo de fazer um controle de doações para a ONG e facilitar de uma maneira visual com a interface para onde o dinheiro está sendo aplicado, assim passando uma transparência tornando uma plataforma limpa, segura, organizada e verdadeira com os usuários e a própria ONG.

3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO

A computação em nuvem oferece uma série de possibilidades que podem mudar a forma como o projeto pode ocorrer. Ao compreender os objetivos do projeto e as questões que funcionam durante o projeto, como a proteção da privacidade e o tamanho do banco de dados, juntamente com os picos de demanda que podem ocorrer, é necessário entender que "as coisas não mudam; nós mudamos" (THOREAU, 2010, p. 115).

Os provedores de nuvem investem pesadamente em segurança, possuindo uma diversidade de soluções e serviços especializados para aplicações e proteção de dados. Há alguns recursos adicionais como criptografia avançada, autenticação multifatorial e monitoramento contra possíveis ataques e ameaças. Os recursos citados são operados de forma "autônoma", ou seja, não há necessidade de deixar alguém designado para a tarefa pois a mesma se torna auto suficiente.

A escalabilidade oferecida pela nuvem também permite um gerenciamento mais eficiente do armazenamento e do processamento de dados, conseguindo se adaptar de forma automática a demanda, o que evita problemas de lentidão e potencializa o projeto. Estas vantagens tornam a computação em nuvem um ótimo recurso para o projeto de uma forma que trás menos "dores de cabeça" aos desenvolvedores e uma tranquilidade a quem usa, seja o doador ou funcionário.

3.3.3 VANTAGENS DA CLOUD COMPUTING

A computação em nuvem chegou como uma novidade no mundo tecnológico, com isso empresas e indivíduos mudaram a forma de armazenamento, gerenciamento, segurança e acesso de dados. Temos alguns protagonistas no quesito de vantagem em computação em nuvem sendo eles:

- **Redução de custos:**

A não necessidade de investimento de hardware, software em estruturas físicas se torna um dos principais pontos que não gera custos a quem usufrui. Além disso, também há que você apenas paga o que usa, então é igual a demanda necessária e assim consegue ter um controle maior sobre o gasto gerado.

- **Escalabilidade:**

O seu rápido modo de ajustar a questão de demandas sobre o negócio seja em momentos de pico ou baixa é um grande alívio para qualquer time de TI juntamente com o próprio servidor.

Backup e Recuperação de Dados Automatizados: A nuvem possibilita backups automáticos e recuperação rápida de dados, o que minimiza riscos de perda de informações sensíveis e assegura a continuidade das operações do aplicativo em caso de falhas.

- **Proteção de dados:** Os provedores de nuvem investem em segurança para proteger seus dados contra ameaças cibernéticas incluindo criptografia de ponta a ponta, autenticação multifatorial (duas etapas) e detecção de ataques e fraudes.
- **Transparência:** A nuvem permite armazenar e disponibilizar dados de forma centralizada e acessível, facilitando a exibição de relatórios financeiros e o uso dos recursos da ONG em tempo real para os doadores.
- **Redução de custos:** A utilização da nuvem elimina a necessidade de investir em servidores físicos, reduzindo custos com manutenção e infraestrutura, o que permite que mais recursos possam ser direcionados para a própria ONG.
- **Conformidade:** A nuvem pode ajudar a cumprir requisitos de conformidade com diversas normas e regulamentações.
- **Flexibilidade e Acessibilidade:** O acesso ao aplicativo de doações pode ser feito a partir de diferentes dispositivos e locais, permitindo que doadores acompanhem suas doações e a aplicação dos recursos de maneira fácil e transparente, a qualquer momento.

3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING

Um sistema de doação baseado em SaaS (Software as a Service) oferece uma série de vantagens que podem otimizar a gestão de doações e impulsionar o sucesso de campanhas de arrecadação de fundos, pois há alguns pontos interessantes.

Facilidade de uso pois são projetados para ser uma interfaces simples e intuitivas, assim permitindo que uma pessoa com pouca experiência com computação possa ter uma noção bem mais lógica e simples. Além disso, possui uma rápida configuração para o uso. Qualquer hora e qualquer lugar pode ser acessado por doadores ou administradores, assim

tornando possível atualizações a todos os momentos sejam por doações ou pela facilidade de o "ter na mão".

A integração é outra vantagem o SaaS que foi escolhido torna a conectividade mais fácil com outros sistemas que podem ser utilizados, no caso é mais fácil atribuir funções com aplicações "externas" como envio de e-mail, marketing e outras plataformas de pagamentos.

Explorar como esses elementos se relacionam entre si e como contribuem para o funcionamento da infraestrutura em nuvem, considerando as necessidades e requisitos específicos da empresa.

3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)

O serviço de AWS é a melhor escolha pela sua diversidade de serviços em comparação às suas concorrentes. O que é um diferencial é a soluções de armazenamento com S3 com o EC2, isso permite que as empresas personalizem a sua infraestrutura de acordo com as necessidades específicas. Além disto também apresenta suas ofertas em banco de dados, machine learning e Internet da Coisas (IoT).

Uma vantagem da AWS é a sua cobertura global (em termos de regiões e zonas). Garantindo uma infraestrutura com alta confiabilidade, onde permite que os serviços sempre fiquem disponíveis diante a uma possível falha em uma região. Tendo em vista que o projeto não pode possuir tempo de inatividade esta é a melhor opção além que a sua abrangência global é essencial.

Com o AWS fica fácil pela flexibilidade, escalabilidade e segurança, além de possuir a maior confiabilidade de infraestrutura de nuvem do mundo. São características ideias para fazer o projeto rodar sem grandes problemas.

3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING

O principal modelo de Cloud Computing utilizado no projeto é uma combinação entre PaaS e IaaS. O PaaS sendo o principal para gerenciar e executar as aplicações que são necessárias durante a operação do sistema, além que a sua principal função é automatizar o provisionamento de servidores e o balanceamento de cargas. Enquanto isso, o EC2 permite a criação de máquinas virtuais escaláveis conforme a demanda apareça.

O Amazon EC2, o serviço de IaaS possibilita a criação de máquinas virtuais escaláveis sob a demanda exigida. É o principal recurso para atender os picos de acesso e ainda garantir que os recursos implementados acompanhem as necessidades do sistema. O uso de IaaS permite o hardware (virtualizado em uma infraestrutura) de cloud pública onde a escalabilidade e elasticidade são garantidas, conforme explicado por Armbrust et al. (2010): "a elasticidade e a escalabilidade da computação em nuvem oferecem recursos sob demanda, permitindo que as empresas ajustem seus recursos de acordo com as necessidades momentâneas do sistema."

O Balanceamento de carga é essencial para que o tráfego de rede (resposta) em outros servidores e serviços seja o mais rápido possível, evitando o famoso "Lag". Um pouco sobre esta anatomia é que ela mais parece uma "teia de aranha" em palavras brucas, são pontos que estão em diferentes lugares mas que no final deixam tudo ligado e fazendo a "rede". De certa forma, possui uma sensibilidade alta no quesito segurança e até mesmo melhora a comunicação entre zonas de disponibilidade.

Além disso, a virtualização e a automação são princípios-chave que sustentam a infraestrutura de nuvem. A segurança de dados sensíveis é fortalecida pelos serviços de segurança da AWS, enquanto a automação permite que tarefas como backups e provisionamento sejam realizadas de forma coordenada e eficiente, reduzindo falhas e simplificando a gestão em momentos críticos.

3.3.7 GOOGLE CLOUD ou AWS

Com o EC2, o serviço de computação em nuvem oferece uma capacidade escalável e flexível, com isso foi possível configurar uma única máquina virtual (com umas instâncias escaláveis) para atender as necessidades que poderão vir na demanda do projeto. Após analisar os principais provedores de nuvem (AWS, Google Cloud, Microsoft Azure), foi optado pela AWS pela sua qualificação em segurança e ampla escalabilidade, além de possuir uma cobertura global facilitando o suporte.

Esta decisão foi tomada com base em muitos fatores sobre serviços, porém o que mais pesou foi no custo inicial de 6,45 USD, sendo o mais viável no momento que a ONG se encontra. A escolha da região de São Paulo (América do Sul) garantiu o melhor custo benefício pelo preço por uma baixa latência devido a proximidade geográfica. O banco de dados (MySQL) já está configurado com backups automáticas e políticas de seguranças protegidas dentro da VPC.

Estes serviços foram o'que tornaram a AWS ser o mais viável, pois os mesmos se tornaram a melhor opção comprada aos demais, o Google Cloud foi descartado por dois motivos principais, primeiro foi a quantidade de serviços que viriam junto com a assinatura e não seria usado por falta de conhecimento dos usuários que mais utilizaram o sistema e não bateu o preço da AWS, ele ficou um pouco acima de R\$85. Enquanto a Microsoft Azure foi de fato a maior concorrente na decisão, com a diferença de preço de R\$20,26. Havia recursos muitos impressionantes pelo preço como Ofertas de Inteligência Artificial e Machine Learning e Simplicidade em Ambientes Híbridos. Entretanto, um dos pontos que levou a escolha da AWS foi a maturidade e experiência em operações na nuvem que a plataforma oferece ao projeto.

3.4 ESTRUTURA DE DADOS

As estruturas de dados são componentes essenciais para o desenvolvimento de sistemas eficientes e bem estruturados, sendo responsáveis por como os dados são organizados, armazenados e manipulados. Cada tipo de estrutura de dados é projetado para atender às necessidades específicas, oferecendo diversas opções de acesso e modificação de dados. Algumas das estruturas mais comuns incluem listas, pilhas, filas, árvores e tabelas de dispersão. A escolha da estrutura adequada depende de como os dados serão usados na aplicação e de quais operações precisam ser realizadas com mais frequência. O desempenho de um sistema está diretamente relacionado à eficiência com que as operações de inserção, remoção e busca são realizadas, o que torna crucial o entendimento do funcionamento de cada estrutura de dados.

No contexto da aplicação que foi desenvolvida, focada em usuários se cadastrarem para poder fazer doações e assim ajudar uma ONG que resgata animais domésticos

desabrigados, a escolha de uma tabela de dispersão se alinha perfeitamente com a necessidade de garantir uma busca rápida e eficiente de informações associadas a cada usuário. já que ao estar cadastrado no sistema, o usuário poderá ver todas as suas doações já feitas até o momento e as suas informações, o acesso para esses dados será feito através da chave que será o token de autenticação da pessoa cadastrada no site.

Como a aplicação é escrita em JavaScript, a tabela de dispersão também é uma escolha natural, já que JavaScript é amplamente utilizado para construir sistemas dinâmicos e interativos na web. Além disso, bibliotecas e frameworks JavaScript oferecem suporte nativo a estruturas como objetos e mapas, que funcionam de maneira semelhante a tabelas de dispersão, facilitando a implementação e a integração dessa estrutura na aplicação. Em JavaScript, o uso de tabelas de dispersão também garante que a aplicação tenha o desempenho necessário para lidar com grandes volumes de dados e múltiplos usuários simultâneos, sem comprometer a experiência do usuário. Como observado em "Estruturas de Dados e Algoritmos em JavaScript" de Michael McMillan, "Uma das grandes vantagens das tabelas de dispersão é sua habilidade de realizar buscas extremamente rápidas em grandes volumes de dados, o que é ideal para sistemas que desativam alto desempenho em tempo de execução."

3.4.1 LEVANTAMENTO DE REQUISITOS

A estrutura de dados que será utilizada na aplicação será uma tabela de dispersão, ela é associada a chaves de pesquisa a valores. Seu objetivo é, a partir de uma chave simples, fazer uma busca rápida e obter o valor desejado, com base na API que foi desenvolvida, essa chave será o token de autenticação do usuário que irá acessar suas informações e todas as suas doações já feitas no sistema, o tratamento com os dados dentro da aplicação começa com o login, onde o usuário insere suas credenciais ou faz seu cadastro no sistema. Após a validação das informações, o sistema gera um token de autenticação, que é essencial para garantir que apenas usuários autorizados possam acessar o sistema.

No site o usuário poderá fazer uma nova doação ou visualizar suas doações já feitas, quando fizer uma nova doação e ela for enviada. Nesse momento, o sistema captura os dados e faz uma requisição HTTP do tipo POST para a API na rota responsável pelo registrador de ações, incluindo o ID do usuário e o valor da doação. Para garantir a segurança, um token de autenticação pode ser enviado no cabeçalho da requisição. Depois

que isso acontece no servidor, o controlador `DonationsController` recebe a requisição e chama o método `create`, que verifica se todos os dados necessários estão presentes. Se tudo estiver correto, o controlador aciona o modelo `Donation` para registrar a doação no banco de dados. Após a inserção, uma confirmação é enviada de volta ao usuário, informando se a doação foi realizada com sucesso.

O usuário também poderá visualizar todas as suas doações já feitas no sistema, isso acontece pois o site realiza uma requisição HTTP do tipo GET para a API, especificamente para a rota que permite buscar doações pelo usuário, como `/donations/user/:id`. O ID do usuário é extraído do token de autenticação para garantir que a requisição seja feita pelo próprio usuário. No servidor, o driver de doações, `DonationsController`, recebe essa requisição e invoca o método `findByUserId`. Este método extrai o ID do usuário dos parâmetros da requisição e em seguida, utiliza o modelo de doações para acessar o banco de dados e buscar todas as doações associadas a esse ID. Essa consulta é realizada de forma eficiente, garantindo que as informações retornadas sejam atualizadas e corretas. Assim que a consulta for concluída, os dados das ações são retornados ao driver, que formata a resposta para que seja facilmente interpretada pelo sistema.

3.4.2 VALIDAÇÃO DOS REQUISITOS

A escolha pela tabela de dispersão é estratégica para o cenário em nuvem, pois oferece suporte direto a operações de leitura e escrita de alta demanda, essenciais para sistemas que operam em ambientes distribuídos. Na nuvem, essa estrutura de dados se beneficia da flexibilidade, podendo escalar horizontalmente para gerenciar o aumento no volume de requisições conforme a base de usuários e a quantidade de doações crescentes. A integração em um ambiente de nuvem permite que a aplicação se adapte às cargas variáveis, ampliando a capacidade de resposta e otimizando a utilização de recursos.

A integração com a nuvem potencializa essa estrutura para fornecer escalabilidade e integração, elementos fundamentais para suportar cargas de trabalho e responder a picos de demanda. No armazenamento, o sistema de partição de dados e a distribuição entre diversos nós na nuvem, de forma que cada instância da tabela de dispersão armazena um subconjunto específico, aliviando a carga de processamento e retardando a latência nas operações. O balanceamento de carga entre as instâncias garante que as operações de leitura e escrita

sejam distribuídas, garantindo resposta rápida aos usuários, mesmo em momentos de alto tráfego.

Além de escalável, uma tabela de dispersão é adequada para lidar com grandes volumes de dados devido à sua capacidade de mapear diretamente chaves a valores, sem varreduras complexas. O uso de um sistema de particionamento baseado em hash divide automaticamente os dados entre diferentes instâncias, reduzindo o risco de sobrecarga e garantindo que o tempo de resposta para o usuário seja eficiente. Em termos de integridade e disponibilidade, o sistema em nuvem replica os dados em diversas zonas de disponibilidade, garantindo resiliência a falhas e alta acessibilidade. A sincronização entre réplicas mantém a consistência dos dados, permitindo que, em caso de falha, o sistema redireciona as requisições para uma réplica ativa, garantindo continuidade sem interrupção de serviço. Como observado em Introdução aos Algoritmos de Thomas H. Cormen, Charles Eric Leiserson, Ronald Rivest, Clifford Stein, “Hashing é uma das técnicas mais poderosas para armazenar e acessar dados, pois permite operações de tempo constante em média. Sua aplicabilidade se estende a sistemas distribuídos, onde a eficiência e a escalabilidade são requisitos fundamentais.”.

Nesse cenário também pode se levar em conta que a tabela de dispersão permite que o sistema manipule operações de leitura e escrita de alta demanda com eficiência, uma vez que a busca por uma chave específica, como o token de autenticação, seja rápida e direta. Pois a nuvem adiciona um componente de elasticidade à aplicação, adaptando automaticamente a infraestrutura conforme o volume de usuários e requisições variadas, o que é fundamental para suportar picos de uso sem degradação

Essa estrutura em nuvem permite que o sistema realize operações de alta demanda com eficiência, pois a busca por chaves específicas, como tokens de autenticação, é rápida e direta. A flexibilidade da nuvem adapta a infraestrutura conforme o volume de usuários e requisições variáveis, permitindo ao sistema suportar picos de uso sem perda de desempenho. A combinação da tabela de dispersão com a infraestrutura da nuvem permite que a aplicação se adapte ao crescimento e as novas demandas sem comprometer a integridade, a disponibilidade e a experiência do usuário.

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS

A Formação para a Vida na UNIFEOB se configura como um pilar fundamental do Projeto Pedagógico de Formação por Competências da instituição. Mais do que um conjunto de disciplinas, essa iniciativa representa um compromisso com o desenvolvimento integral dos alunos, preparando-os para serem cidadãos conscientes, críticos e engajados na transformação da sociedade.

3.5.1 ENFRENTANDO ESTEREÓTIPOS

A internet transformou-se num novo espaço de busca de reconhecimento e sucesso. Meios de comunicação como Instagram, Facebook e TikTok oferecem uma versão idealizada da nossa vida, fomentando uma cultura de comparação, na qual a felicidade e a autoestima tornam-se aferidas por medidas externas, em seus aspectos superficiais, tais como gostos e seguidores.

Quando comparamos a nossa versão imperfeita de vida com a versão idealizada dos outros, como a construção sempre perfeita que encontramos nas redes sociais, estamos vulneráveis a uma derrocada da nossa imagem. Comparando, afinal, a nossa vida imperfeita com a vida perfeita da vida dos outros. Segurança da imagem pode resultar numa crise de autoestima, de ansiedade, e da tentativa de corrigir a ideia de que você nunca é bom o suficiente.

Outro aspecto da cultura digital é a exigência da validação externa para ser validado e avaliado. O desejo de reconciliação e aceitação pode ser a busca por curtidas e comentários positivos, que podem ter um caráter de obsessão, resultando, conseqüentemente, numa pressão com a qual leva à formação da ansiedade e do desejo de validação para os outros. O desejo de levar a multidão pelas suas experiências e formas pode resultar em perda da originalidade e da captura da infância das opiniões alheias.

As redes sociais permitem construir as histórias sobre a própria vida e selecionar quais são os momentos e as experiências que você gostaria de incluir. Essa seleção rigorosa pode causar uma representação distorcida da realidade, tanto para o criador de conteúdo quanto para o consumidor de conteúdo. Mostrar somente as melhores experiências, sem dúvida alguma, pode dar a impressão de felicidade e realização, e pode também causar o sentimento de inadequação nas pessoas ao seu redor.

3.5.2 ESTUDANTES NA PRÁTICA

A expressão "tecnologia para contrabalançar estereótipos..." já nos leva a um debate complexo e intrigante. A internet transformou-se num novo espaço de busca de reconhecimento e sucesso. Meios de comunicação como Instagram, Facebook e TikTok oferecem uma versão idealizada da nossa vida, fomentando uma cultura de comparação, na qual a felicidade e a autoestima tornam-se aferidas por medidas externas, em seus aspectos superficiais, tais como gostos e seguidores.

Quando comparamos a nossa versão imperfeita de vida com a versão idealizada dos outros, como a construção sempre perfeita que encontramos nas redes sociais, estamos vulneráveis a uma derrocada da nossa imagem. Comparando, afinal, a nossa vida imperfeita com a vida perfeita da vida dos outros. Segurança da imagem pode resultar numa crise de autoestima, de ansiedade, e da tentativa de corrigir a ideia de que você nunca é bom o suficiente.

Outro aspecto da cultura digital é a exigência da validação externa para ser validado e avaliado. O desejo de reconciliação e aceitação pode ser a busca por curtidas e comentários positivos, que podem ter um caráter de obsessão, resultando, conseqüentemente, numa pressão com a qual leva à formação da ansiedade e do desejo de validação para os outros. O desejo de levar a multidão pelas suas experiências e formas pode resultar em perda da originalidade e da captura da infância das opiniões alheias.

As redes sociais permitem construir as histórias sobre a própria vida e selecionar quais são os momentos e as experiências que você gostaria de incluir. Essa seleção rigorosa pode causar uma representação distorcida da realidade, tanto para o criador de conteúdo quanto para o consumidor de conteúdo. Mostrar somente as melhores experiências, sem dúvida alguma, pode dar a impressão de felicidade e realização, e pode também causar o sentimento de inadequação nas pessoas ao seu redor.

4. CONCLUSÃO

O desenvolvimento do sistema "Gerenciador de Doações para ONG de Pets" foi fundamentado em tecnologias modernas e práticas de design robustas para atender às necessidades específicas da ONG, focando na gestão de doações e na transparência das informações para doadores e administradores. A implementação de uma API RESTful no back-end, com autenticação JWT e arquitetura escalável, garante segurança e facilidade na manipulação dos dados, possibilitando o acesso seguro e controlado às informações dos doadores, pets e histórico de doações.

Além disso, a adoção da computação em nuvem pela AWS potencializa o projeto, oferecendo escalabilidade, backup automatizado e segurança reforçada com autenticação multifatorial e criptografia. Com recursos como balanceamento de carga e escalabilidade automatizada, o sistema é capaz de lidar com altos volumes de acesso e armazenamento, assegurando a continuidade e a alta disponibilidade dos serviços. A escolha da tabela de

dispersão como estrutura de dados permite operações rápidas de leitura e escrita, essenciais para um sistema que deve responder em tempo real às consultas dos usuários, otimizando o desempenho e proporcionando uma experiência ágil e eficiente.

Em resumo, o "Gerenciador de Doações para ONG de Pets" se destaca pela sua capacidade de unir tecnologia e propósito social. Ele não só facilita a gestão de doações, como também promove transparência, acessibilidade e segurança, beneficiando tanto os doadores quanto a ONG. A infraestrutura projetada garante um sistema escalável e confiável, pronto para evoluir e acompanhar o crescimento da base de usuários e as necessidades da ONG.

REFERÊNCIAS

Computação em nuvem

TOFFLER, Alvin. *Future Shock*. New York: Random House, 1970.

ARMBRUST, Michael et al. *A view of cloud computing*. Communications of the ACM, v. 53, n. 4, p. 50-58, 2010.

MCMILLAN, Michael. *Estruturas de dados e algoritmos em JavaScript, 2014*.

CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald; STEIN, Clifford. *Introdução aos algoritmos*, 1990.

Banco de dados

Davi Leyendecker

<https://www.dio.me/articles/os-tres-niveis-da-modelagem-de-dados-conceitual-logico-e-fisico>

Igor Nascimento Alves

<https://www.alura.com.br/artigos/diferencas-modelo-logico-modelo-fisico#conceitos-e-objetivos-de-modelos-logico-e-fisico-de-banco-de-dados>

Beatriz Magalhães

https://www.alura.com.br/artigos/trigger-em-sql?srsltid=AfmBOor_7mZuWBN4hgEjCg8ghOWL6DuLKo0jfYxEuTV3hV3X66bgNtGK

Danielle Monteiro

<https://imasters.com.br/banco-de-dados/afinal-de-contas-o-que-e-uma-stored-procedure>

Lógica de programação

RICHARDSON, Chris. *Microservices Patterns: With Examples in Java*. Shelter Island, NY: Manning Publications, 2018.

PILGRIM, Mark. *Pro HTML5 with CSS, JavaScript, and Multimedia*. Berkeley: Apress, 2011.

ANEXOS

Figura 1: Banner

TECNOLOGIA

ENFRENTANDO ESTEREÓTIPOS..

A internet, por meio de redes sociais como Instagram, Facebook e TikTok, criou novas formas de medir aprovação e sucesso, usando curtidas e seguidores. Ao buscar validação, as pessoas mostram versões idealizadas de suas vidas, o que pode distorcer a realidade e afetar a autoestima dos outros.

Exemplos aplicáveis: Comparação social, busca pela validação externa, subversão da realidade.



Os exemplos mostram que redes sociais incentivam comparações, criando baixa autoestima, e a busca por validação externa, gerando ansiedade. Além disso, idealizar a vida online pode prejudicar relações e produtividade.

Fonte: Autores