



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

LOGDEV

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

LOGDEV

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 – Profª. Mariângela Martimbianco Santos

Estudantes:

Derik dos Anjos Alves Leite, 23001076

José Vinícius Martins Perina , 23000555

João Victor Saloti Alves , 23000200

Victor de Souza Rodrigues, 23000819

Virgilio Pedro Batista Junior, 23000383

SÃO JOÃO DA BOA VISTA, SP
NOVEMBRO 2024

SUMÁRIO

1. INTRODUÇÃO	4
2. DESCRIÇÃO DA EMPRESA	6
3. PROJETO INTEGRADO	7
3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS	7
3.1.1 MODELO LÓGICO	7
3.1.2 MODELO FÍSICO	7
3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO	8
3.2.1 APPLICATION PROGRAMMING INTERFACE (API) - BACK-END.	8
3.2.2 FRONT-END	8
3.3 COMPUTAÇÃO EM NUVEM	9
3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING	9
3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO	9
3.3.3 VANTAGENS DA CLOUD COMPUTING	9
3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING	10
3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)	10
3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING	10
3.3.7 GOOGLE CLOUD ou AWS	11
3.4 ESTRUTURA DE DADOS	11
3.4.1 LEVANTAMENTO DE REQUISITOS	11
3.4.2 VALIDAÇÃO DOS REQUISITOS	11
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS	11
3.5.1 ENFRENTANDO ESTEREÓTIPOS	12
3.5.2 ESTUDANTES NA PRÁTICA	12
4. CONCLUSÃO	14
REFERÊNCIAS	15
ANEXOS	16

1. INTRODUÇÃO

O objetivo deste Projeto Integrado é prototipar um sistema de gestão personalizado voltado para uma ONG de proteção a animais. Este sistema será projetado para atender às necessidades específicas da organização, incluindo a gestão de adoções, arrecadação de doações e acompanhamento de atendimentos veterinários. Com um foco robusto em Business Intelligence (BI), buscamos fornecer ferramentas que suportem a tomada de decisões estratégicas, otimizando a operação e ampliando o impacto da ONG em sua missão.

A plataforma será desenvolvida sobre uma arquitetura de computação em nuvem, garantindo escalabilidade e disponibilidade dos dados. Utilizaremos técnicas avançadas de gerenciamento de banco de dados para assegurar que as informações estejam sempre acessíveis e atualizadas, permitindo um melhor acompanhamento das atividades da ONG.

Além disso, nosso projeto se alinha aos princípios dos Objetivos de Desenvolvimento Sustentável (ODS), refletindo nosso compromisso com uma sociedade mais sustentável e inclusiva. Ao integrar as melhores práticas em tecnologia e gestão, almejamos não apenas dominar conceitos essenciais, mas também aplicá-los de forma criativa e responsável para beneficiar a ONG e os animais que ela protege.

Ao final deste Projeto Integrado, estaremos não apenas fornecendo uma solução inovadora para o gerenciamento das atividades da ONG, mas também contribuindo para a promoção do bem-estar animal e o avanço em direção a um futuro mais sustentável e inclusivo, alinhando-nos às metas das ODS.

2. DESCRIÇÃO DA EMPRESA

A empresa que está realizando o projeto de Sistema de gestão e inteligência de negócios para organizações sociais, de razão social LOGDEV TECHNOLOGY, com CNPJ 00.000.000/0001-00, localizada em São João da Boa Vista - SP na avenida Dr. Octávio da Silva Bastos, 2439, Campus II - Mantiqueira, Bairro Nova São João. A LOGDEV COMPANY, prestou serviço de desenvolvimento de sistema, para gerenciamento da S.O.S APAN, com CNPJ 01.433.298/0001-96 localizada na cidade de Casa Branca - São Paulo.

3. PROJETO INTEGRADO

O principal propósito deste projeto é aperfeiçoar a gestão e administração das ONGs. Nosso objetivo é o desenvolvimento de software, design e sistemas para facilitar a administração e gerenciamento de doações, resgates e adoções, tudo voltado para a praticidade, a fim de evitar o acúmulo dessas tarefas e melhorar o foco no cuidado dos animais. Essa iniciativa demonstra o potencial transformador da tecnologia na administração das ONG, oferecendo vantagens palpáveis tanto para a instituição quanto para seus administradores. O projeto representa um passo significativo em direção à modernização dos processos administrativos e organizacionais e à garantia de uma gestão de presença mais precisa e eficiente.

3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS

Os tópicos avançados de banco de dados abrangem áreas mais complexas e especializadas do estudo e da prática no campo de bancos de dados, além dos conceitos básicos de modelagem e manipulação de dados. Esses tópicos são essenciais para lidar com grandes volumes de dados, garantir a integridade e a segurança das informações, e otimizar a performance dos sistemas de banco de dados em ambientes de produção complexos.

Um dos tópicos mais relevantes é a modelagem de dados avançada, que vai além dos modelos relacionais tradicionais. O uso de modelos de dados não relacionais, como bancos de dados orientados a grafos, colunares ou de documentos, tornou-se cada vez mais importante para lidar com dados semi-estruturados ou não estruturados, que não se encaixam bem nos modelos relacionais tradicionais. Bancos de dados como MongoDB e Neo4j exemplificam essas abordagens (Elmasri & Navathe, 2015).

Outro tema crucial é a gestão de grandes volumes de dados ou Big Data. A habilidade de armazenar, processar e analisar grandes conjuntos de dados em tempo real exige o uso de tecnologias específicas, como Hadoop, Spark e NoSQL. A computação distribuída e o uso de arquiteturas de armazenamento distribuído são fundamentais para garantir escalabilidade e desempenho em ambientes de Big Data (Chen et al., 2012).

A consistência e recuperação de dados também são tópicos avançados vitais. Eles tratam da manutenção da integridade dos dados em sistemas distribuídos e do desenvolvimento de protocolos para recuperação de falhas. Técnicas como o uso de transações distribuídas, protocolos de commit de duas fases e redes de recuperação garantem que, em caso de falhas, o sistema de banco de dados possa recuperar ou preservar a integridade dos dados sem perda de informações. A replicação de dados e os algoritmos de consistência em sistemas distribuídos são essenciais para garantir a disponibilidade e a consistência dos dados em diferentes servidores e locais (Gray & Reuter, 1993).

A segurança de dados é outro aspecto avançado, onde são aplicadas técnicas de criptografia, controle de acesso e autenticação para proteger dados sensíveis. A proteção contra SQL injection e o gerenciamento de permissões de usuários em ambientes compartilhados são aspectos cruciais para prevenir ataques e garantir a integridade e a confidencialidade dos dados (Schneier, 2015).

Por fim, a otimização de consultas envolve o uso de algoritmos complexos para melhorar a eficiência das operações em bancos de dados. Técnicas de indexação avançada, planos de execução de consultas e rearranjo de dados em disco são importantes para reduzir o tempo de resposta e maximizar a performance de sistemas em larga escala, especialmente em consultas envolvendo grandes volumes de dados e operações complexas (Garcia-Molina et al., 2009).

Para o nosso sistema da ONG de animais, o MySQL com conexão em nuvem com a AWS é uma ótima opção. Isso porque oferece gerenciamento de dados seguro e escalável, permitindo armazenar e gerenciar informações sobre animais, doadores, voluntários e eventos. A segurança é garantida pela criptografia e autenticação da AWS, protegendo dados sensíveis.

Além disso, a conexão em nuvem com a AWS permite ajustar automaticamente o tamanho do banco de dados conforme necessário, garantindo acesso remoto para voluntários e funcionários. O pagamento baseado no uso reduz custos de infraestrutura.

O sistema pode incluir funcionalidades como cadastro de animais, gerenciamento de doações, voluntariado, eventos e relatórios. É possível integrar com gateways de pagamento para doações, redes sociais para compartilhar informações, email marketing para enviar newsletters ou atualizações, e Google Maps para localizar animais e eventos.

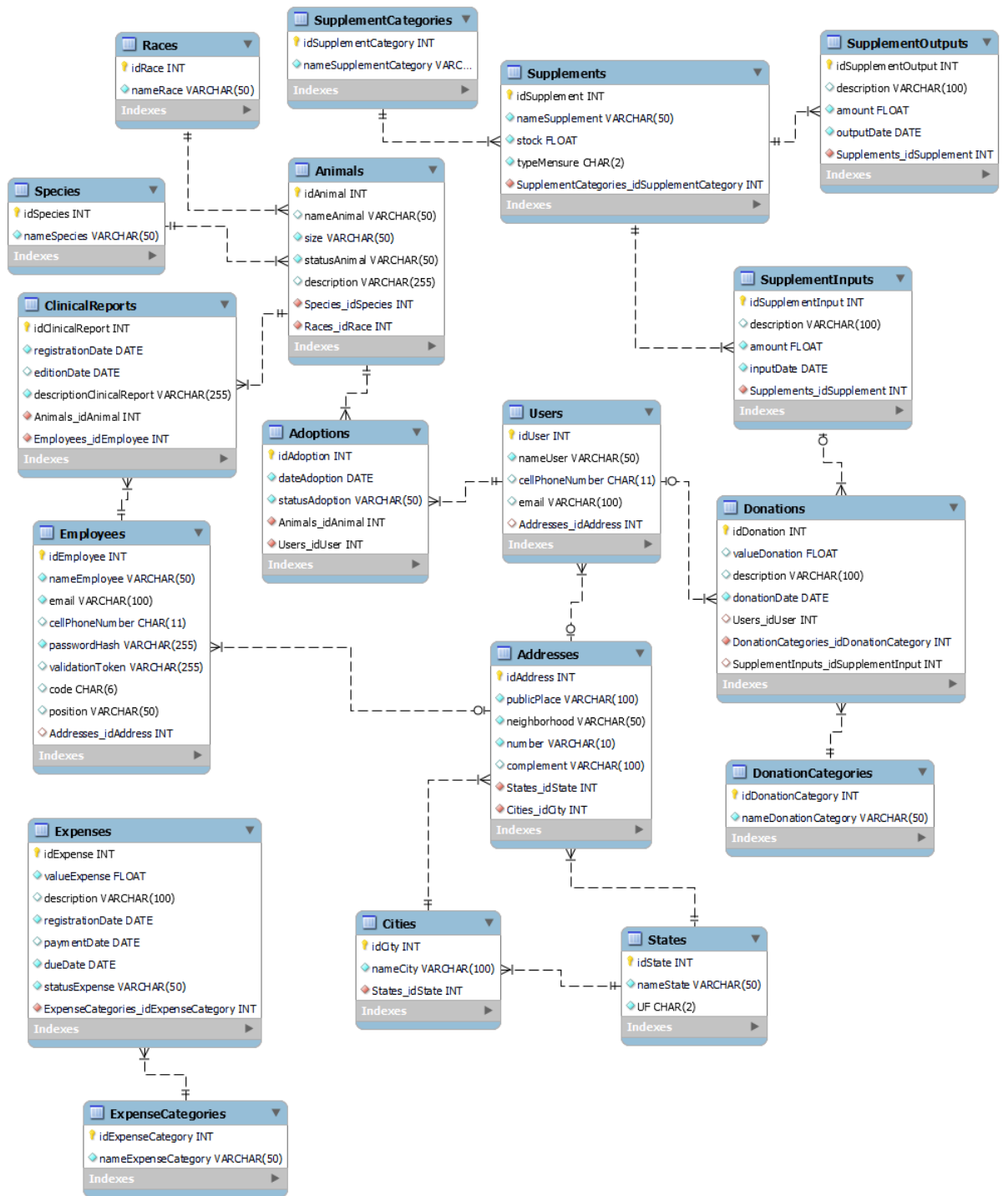
Para desenvolver o sistema, são necessários conhecimentos em linguagem de programação (PHP, Python, Java, etc.), framework (Laravel, Django, Spring, etc.), banco de dados (MySQL) e conexão em nuvem (AWS RDS ou Aurora). Com esses recursos, é possível criar um sistema eficiente e seguro para gerenciar a ONG de animais, garantindo uma experiência positiva para doadores, voluntários e animais.

3.1.1 MODELO LÓGICO

O Modelo Lógico de banco de dados é uma representação detalhada da estrutura de dados, que visa traduzir o modelo conceitual (com uma visão mais ampla e abstrata) para um formato mais próximo do que será implementado no sistema de gerenciamento de banco de dados (SGBD). Ele define as entidades, seus atributos e os relacionamentos entre elas de forma estruturada e já considera as regras e restrições específicas do SGBD que será usado.

O modelo lógico tem como objetivo eliminar redundâncias (normalização) e otimizar a estrutura para a performance e integridade dos dados. Ele atua como um guia detalhado para a criação do Modelo Físico, que será a implementação final do banco de dados no SGBD.

Figura 1



3.1.2 MODELO FÍSICO

O Modelo Físico de banco de dados é a etapa final da modelagem, onde o design lógico do banco é traduzido em instruções específicas para o sistema de gerenciamento de banco de dados (SGBD) escolhido, como MySQL, PostgreSQL, ou SQL Server. Esse modelo define como os dados serão armazenados, acessados e organizados fisicamente dentro do sistema.

No modelo físico, cada aspecto é detalhado para que o banco seja criado de forma funcional, com desempenho otimizado e integridade garantida. O modelo físico é fundamental para transformar o design do banco em uma estrutura operacional que funcione no SGBD escolhido, sendo o último estágio antes da implementação do banco de dados real.

Definição de TRIGGERS

No contexto de banco de dados, triggers são procedimentos ou eventos automáticos que são configurados para serem executados em resposta a ações específicas dentro de uma tabela ou de um banco de dados. Essas ações geralmente envolvem operações como insert, update ou delete.

Figura 1

```
DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `supplementinputs_AFTER_INSERT` AFTER INSERT ON
`supplementinputs` FOR EACH ROW BEGIN
    declare now_stock INT;
    declare new_stock INT;

    SELECT stock INTO now_stock FROM supplements WHERE idSupplement =
NEW.Supplements_idSupplement;

    SET new_stock = now_stock + NEW.amount;

    UPDATE supplements SET stock = new_stock WHERE idSupplement =
NEW.Supplements_idSupplement;
END$$

DELIMITER;
```

Definição de PROCEDURES

No contexto de banco de dados, procedures (ou procedimentos armazenados) são conjuntos de instruções SQL armazenados no banco, que podem ser executados para realizar tarefas específicas. Elas funcionam como "funções" pré-compiladas que permitem a automação de processos no banco de dados.

Figura 2

```
DELIMITER ;;
CREATE DEFINER=`root`@`localhost` PROCEDURE `create_user`(
  IN par_nameUser VARCHAR(50),
  IN par_cellPhoneNumber CHAR(11),
  IN par_email VARCHAR(100),

  IN par_publicPlace VARCHAR(100),
  IN par_neighborhood VARCHAR(50),
  IN par_number VARCHAR(10),
  IN par_complement VARCHAR(100),
  IN par_states_idState INT,
  IN par_cities_idCity INT
)
BEGIN
  DECLARE last_address_id INT;

  INSERT INTO addresses (publicPlace, neighborhood, number
, complement, States_idState, Cities_idCity)
  VALUES
  (par_publicPlace, par_neighborhood, par_number, par_complement, par_states_idState, par_cities_idCity);

  SET last_address_id = LAST_INSERT_ID();

  INSERT INTO users (nameUser, cellPhoneNumber, email, Addresses_idAddress)
  VALUES (par_nameUser, par_cellPhoneNumber, par_email, last_address_id);
END ;;
DELIMITER ;
```

Definição de VIEWS.

No contexto de banco de dados, views (ou visões) são objetos que atuam como tabelas virtuais, criadas a partir de uma consulta SQL. Elas não armazenam dados fisicamente, mas exibem dados com base em uma ou mais tabelas subjacentes.

Figura 3

```
DELIMITER $$

CREATE DEFINER=`root`@`localhost` TRIGGER `supplementinputs_AFTER_INSERT` AFTER INSERT ON
`supplementinputs` FOR EACH ROW BEGIN
    declare now_stock INT;
    declare new_stock INT;

    SELECT stock INTO now_stock FROM supplements WHERE idSupplement =
NEW.Supplements_idSupplement;

    SET new_stock = now_stock + NEW.amount;

    UPDATE supplements SET stock = new_stock WHERE idSupplement =
NEW.Supplements_idSupplement;
END$$

DELIMITER;CREATE VIEW ClinicalReportsView AS
SELECT
    cr.idClinicalReport,
    cr.registrationDate,
    cr.editionDate,
    cr.descriptionClinicalReport,
    a.nameAnimal,
    e.nameEmployee AS EmployeeName,
    s.nameSpecies AS SpeciesName,
    r.nameRace AS RacesName
FROM
    clinicalreports cr
JOIN
    animals a ON cr.Animals_idAnimal = a.idAnimal
JOIN
    employees e ON cr.Employees_idEmployee = e.idEmployee
JOIN
    species s ON a.Species_idSpecies = s.idSpecies
JOIN
    races r ON a.Races_idRace = r.idRace;
```

3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO

A linguagem de programação é um conjunto estruturado de regras e sintaxes que permite aos desenvolvedores escreverem instruções que um computador pode entender e executar. Essas instruções são essenciais para a criação de softwares, sites, aplicativos e sistemas operacionais, dentre outros. As linguagens de programação se dividem em dois grandes grupos: linguagens de baixo nível, mais próximas do código de máquina, como o Assembly, e linguagens de alto nível, mais próximas da linguagem humana, como Python, Java, C++ e JavaScript. A escolha da linguagem de programação depende do tipo de projeto, das necessidades específicas de desempenho, das preferências do programador e do ecossistema de ferramentas e bibliotecas disponíveis para cada linguagem (Sebesta, 2019).

Além da linguagem, as técnicas de programação se referem aos métodos e práticas que os desenvolvedores utilizam para resolver problemas de forma eficaz e eficiente. Entre as técnicas mais comuns estão a programação orientada a objetos (POO), que organiza o código em "objetos" que representam entidades do mundo real; a programação funcional, que trata funções como cidadãos de primeira classe, promovendo a imutabilidade e a ausência de efeitos colaterais; e a programação procedural, que organiza o código em funções ou procedimentos, focando na sequência de instruções. Cada técnica tem suas vantagens e é escolhida conforme o tipo de aplicação e os requisitos do sistema a ser desenvolvido. Técnicas de otimização de código, como o uso de algoritmos eficientes e a análise de complexidade computacional, também são fundamentais para melhorar o desempenho dos sistemas (Knuth, 1997).

A combinação de uma linguagem de programação adequada e a aplicação das técnicas corretas são cruciais para o sucesso no desenvolvimento de software, influenciando diretamente a qualidade, a manutenção e a escalabilidade do sistema.

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

No desenvolvimento do backend, optei por utilizar o Node.js com JavaScript, uma combinação que oferece grande performance e flexibilidade para a construção de aplicativos escaláveis e eficientes. O Node.js, por ser uma plataforma baseada no motor V8 do Google Chrome, permite que o código JavaScript seja executado no lado do servidor, oferecendo excelente performance para tarefas de I/O intensivas, como leitura e escrita em banco de dados. Além disso, sua natureza assíncrona e orientada a eventos (event-driven) facilita a criação de aplicações altamente responsivas, capazes de lidar com múltiplas requisições simultâneas sem comprometer a performance (Tilkov & Vinoski, 2010).

Para a integração com o banco de dados, utilizei o Knex.js, que é um construtor de consultas SQL para Node.js. O Knex oferece uma interface poderosa para interagir com bancos de dados relacionais, como PostgreSQL, MySQL e SQLite. Ele facilita a criação de consultas SQL complexas de forma programática e segura, evitando a escrita manual de SQL e ajudando a prevenir problemas como injeção de SQL (Soni, 2016). Além disso, o Knex.js permite a utilização de migrações, garantindo que o banco de dados se mantenha consistente ao longo do desenvolvimento, controlando e versionando as alterações na estrutura do banco (Soni, 2016). A funcionalidade de transações do Knex também foi fundamental para garantir a integridade dos dados, aspecto essencial em sistemas que exigem alta confiabilidade e consistência nas operações (Hughes, 2019).

Em conjunto, o Node.js e o Knex.js proporcionaram uma base sólida e eficiente para o desenvolvimento do backend, permitindo a criação de uma aplicação robusta, com uma comunicação eficiente com o banco de dados e uma arquitetura escalável e de fácil manutenção. Essa combinação de tecnologias é amplamente adotada em aplicações modernas devido à sua escalabilidade, desempenho e flexibilidade, o que facilita a construção de soluções de alta performance com baixo custo de operação (Tilkov & Vinoski, 2010; Soni, 2016).

3.2.2 FRONT-END

No desenvolvimento do front-end, optei por utilizar o AngularJS em conjunto com o Tailwind CSS. O AngularJS é um framework JavaScript de código aberto, desenvolvido pelo Google, amplamente utilizado para a construção de aplicações web dinâmicas de página única (SPA, Single Page Application). Com o Angular, foi possível criar uma aplicação altamente interativa e modular, utilizando a arquitetura MVC (Model-View-Controller) que facilita a organização do código e a manutenção da aplicação ao longo do tempo (Brad Green & Shyam Seshadri, 2018). O Angular oferece recursos robustos como data binding, injeção de dependência e diretivas personalizadas, que tornam o desenvolvimento mais ágil e a interação do usuário mais fluida (Abrams, 2017).

Além disso, a linguagem utilizada com o Angular é o TypeScript, que é um superconjunto do JavaScript que adiciona tipagem estática ao código. O uso do TypeScript no Angular melhora a legibilidade e a manutenção do código, tornando o processo de desenvolvimento mais seguro e eficiente, já que ele permite detectar erros em tempo de compilação e oferece suporte avançado para a criação de interfaces e tipos, o que ajuda a evitar problemas comuns durante o desenvolvimento (Basarat, 2016). A tipagem estática também facilita a refatoração de código e melhora a colaboração em equipes de desenvolvimento.

Para a parte de estilização da aplicação, utilizei o Tailwind CSS, um framework de utilitários que permite a criação de interfaces responsivas e customizáveis de forma rápida e eficiente. Diferente de outros frameworks tradicionais como o Bootstrap, o Tailwind não impõe um estilo específico, mas oferece classes de utilitários que permitem aplicar estilos diretamente no HTML, o que proporciona maior controle sobre o design e evita a sobrecarga de código desnecessário (Adam Wathan, 2017). A combinação do Angular com o Tailwind CSS facilitou o desenvolvimento de uma interface moderna, otimizada e altamente responsiva, sem comprometer a performance da aplicação.

Em resumo, o uso do AngularJS, com a programação em TypeScript e a estilização com Tailwind CSS, proporcionou uma estrutura sólida e escalável para o desenvolvimento do front-end, permitindo a criação de uma aplicação web de alta performance, com um código bem organizado e fácil de manter. A escolha dessas tecnologias refletiu a necessidade de criar

uma experiência de usuário fluida, com uma interface moderna e eficiente, ao mesmo tempo em que garantiu maior produtividade e qualidade no desenvolvimento.

3.3 COMPUTAÇÃO EM NUVEM

3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING

Os objetivos específicos do projeto em relação à computação em nuvem para uma ONG de animais incluem a melhoria da eficiência operacional, a redução de custos, o aumento da escalabilidade e a melhoria na segurança e confiabilidade. Em termos de eficiência operacional, a ONG pode implementar um sistema de gerenciamento de dados para acompanhar o status de adoções, doações e voluntários, facilitando o acesso e a atualização das informações. Além disso, o uso de ferramentas de colaboração na nuvem pode melhorar a comunicação entre a equipe e os voluntários.

A redução de custos é outro objetivo fundamental. A ONG pode minimizar os gastos com servidores físicos e manutenção, utilizando soluções de nuvem que não exigem investimentos altos em hardware. A criação de plataformas seguras para doações online também facilita o processo de contribuição, aumentando a arrecadação sem custos operacionais elevados.

A escalabilidade é essencial para a ONG, permitindo a expansão dos serviços, como o aumento de campanhas de adoção ou resgate de animais, sem a necessidade de reestruturação física. A realização de eventos e campanhas online também ajuda a alcançar um público maior, aumentando a conscientização e as doações.

A segurança e confiabilidade são prioridades para garantir a proteção dos dados dos adotantes e das doações. A computação em nuvem oferece soluções com criptografia e backups automáticos, além de um plano de recuperação de desastres que assegura a continuidade das operações em caso de problemas técnicos.

A nuvem facilita o uso de sistemas de gerenciamento de relacionamento com o cliente (CRM), ajudando a ONG a acompanhar interações com adotantes e doadores. Isso promove uma organização melhor e acesso fácil às informações. Além disso, a adoção de soluções de nuvem baseadas em assinatura permite à ONG evitar grandes investimentos iniciais, com manutenção e atualizações realizadas pelo provedor, liberando recursos para outras áreas.

Por fim, a nuvem possibilita a utilização de ferramentas de marketing digital, como plataformas de e-mail marketing e redes sociais, que ajudam a engajar mais pessoas em campanhas de adoção e conscientização. Com isso, a implementação da computação em nuvem pode transformar as operações de uma ONG de animais, aumentando sua eficiência, reduzindo custos e possibilitando a escalabilidade de suas atividades. Essa tecnologia, alinhada às necessidades específicas da organização, pode gerar um impacto significativo na proteção e bem-estar dos animais.

3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO

A computação em nuvem pode ser aplicada de forma concreta em uma ONG de animais de várias maneiras, alinhando-se diretamente aos objetivos do projeto e proporcionando benefícios significativos. Uma das aplicações é a criação de uma plataforma de adoção online, onde a ONG pode desenvolver um site que utiliza serviços de nuvem para hospedar e gerenciar perfis de animais disponíveis para adoção. Isso inclui a integração de formulários online que facilitam a inscrição de adotantes e permitem que os interessados façam perguntas e agendem visitas. Com um banco de dados em nuvem, a ONG pode atualizar facilmente informações sobre os animais, como status de adoção e histórico médico (Marston et al., 2011).

Outra aplicação relevante é o gerenciamento de doações. A ONG pode implementar uma plataforma de arrecadação de fundos na nuvem que permita a aceitação de doações online de forma segura. Ferramentas como PayPal ou Stripe podem ser integradas ao sistema da ONG, garantindo que o processo de doação seja simples e acessível, aumentando a arrecadação e a transparência nas finanças (Kumar et al., 2017).

A nuvem também pode ser utilizada para melhorar a comunicação e a colaboração entre a equipe. Com ferramentas como Google Workspace ou Microsoft 365, a equipe da ONG pode colaborar em tempo real, compartilhar documentos e gerenciar tarefas, independentemente da localização. Isso é especialmente importante em campanhas de resgate ou eventos, onde a coordenação é essencial (Rao, 2016).

Além disso, a computação em nuvem facilita o marketing e a conscientização. Ferramentas de automação de marketing, hospedadas na nuvem, permitem que a ONG alcance um público maior com campanhas direcionadas, aumentando a conscientização sobre a adoção de animais e as necessidades de financiamento (Chaffey, 2019). Também é possível criar um sistema em nuvem para gerenciar o cadastro de voluntários, agendar turnos e acompanhar atividades, facilitando a comunicação entre a ONG e os voluntários.

Os benefícios específicos da computação em nuvem para a ONG de animais são numerosos. A maior flexibilidade é um dos principais aspectos, pois permite que a ONG ajuste rapidamente sua infraestrutura de TI conforme suas necessidades mudam. Se houver um aumento no número de animais resgatados, a ONG pode escalar os recursos facilmente para gerenciar os dados e as operações associadas (Zhang et al., 2010). A agilidade no desenvolvimento de soluções é outra vantagem, pois os serviços em nuvem possibilitam a implementação rápida de novas funcionalidades ou modificações em sistemas existentes, acelerando o tempo de resposta a novas demandas, como campanhas de emergência para arrecadar fundos após um desastre (Armbrust et al., 2010).

A redução de custos operacionais também é significativa. A ONG pode evitar grandes investimentos iniciais em hardware, optando por um modelo de pagamento por uso. A manutenção e atualizações são gerenciadas pelo provedor de nuvem, liberando recursos financeiros que podem ser direcionados a programas em prol dos animais (Marston et al., 2011). Além disso, a melhoria na segurança dos dados é um benefício crucial, já que as soluções em nuvem oferecem medidas avançadas de segurança, como criptografia e autenticação multifator, garantindo que informações sensíveis de adotantes e doadores estejam protegidas (Rao, 2016).

Por fim, a capacidade de realizar backups automáticos e a recuperação de desastres são essenciais para garantir a continuidade das operações da ONG, minimizando o risco de perda de dados críticos e permitindo uma recuperação rápida de eventuais problemas (Zhang et al., 2010). Em resumo, a computação em nuvem pode transformar as operações de uma

ONG de animais, proporcionando soluções eficientes e adaptáveis que atendem às necessidades atuais e preparam a organização para um futuro em crescimento e mudança.

3.3.3 VANTAGENS DA CLOUD COMPUTING

A computação em nuvem oferece diversas vantagens específicas para uma ONG de animais, considerando suas necessidades atuais. A elasticidade dos recursos permite à ONG escalar rapidamente a capacidade de armazenamento e processamento durante picos de atividade, como campanhas de adoção (Armbrust et al., 2010). O modelo de pagamento conforme o uso elimina a necessidade de grandes investimentos iniciais em infraestrutura, possibilitando uma alocação financeira mais eficiente para programas de proteção animal (Marston et al., 2011).

Outro benefício é a alta disponibilidade, garantida pelos provedores de nuvem, o que é crucial para evitar interrupções em plataformas online de doações e adoções (Zhang et al., 2010). A segurança aprimorada é igualmente importante, com tecnologias como criptografia e autenticação multifator protegendo dados sensíveis de adotantes e doadores (Rao, 2016). Além disso, a nuvem facilita a colaboração e o acesso remoto, permitindo que a equipe e os voluntários trabalhem juntos de qualquer lugar (Chaffey, 2019).

Em termos econômicos, a migração para a nuvem pode resultar em significativa redução de custos operacionais, eliminando despesas com manutenção de hardware e software (Kumar et al., 2017). A otimização de recursos permite à ONG pagar apenas pelo que realmente utiliza, evitando desperdícios (Marston et al., 2011). A economia de capital é outro fator, pois a ONG pode redirecionar recursos que seriam usados em infraestrutura para atividades essenciais (Zhang et al., 2010). A eficiência nos processos também melhora com a automação, resultando em economia de custos com pessoal (Rao, 2016).

Por fim, se houver estudos específicos sobre a análise econômica do projeto, é importante que os estudantes abordem os benefícios financeiros da computação em nuvem, como a comparação de custos antes e depois da migração e o impacto na arrecadação de fundos. Em resumo, a computação em nuvem proporciona à ONG não apenas vantagens operacionais e de segurança, mas também uma economia financeira significativa, tornando-a uma solução ideal para suas necessidades.

Segue as principais vantagens do cloud computing.

- Menor **custo inicial**: A cloud computing permite a redução significativa de investimentos iniciais em infraestrutura, uma vez que os recursos são oferecidos como serviço e pagos conforme o uso.
- Redução **do custo operacional**: O gerenciamento, manutenção e suporte dos sistemas são responsabilidade do provedor, o que reduz os custos para os assinantes.
- Pagamento **por uso (pay-per-use)**: Os usuários pagam somente pelo tempo e recursos efetivamente utilizados, aumentando a eficiência de custos.
- Escalabilidade e **elasticidade**: A cloud computing oferece poder de computação e armazenamento praticamente ilimitados, permitindo o ajuste dinâmico conforme a demanda.
- Alta **confiabilidade e disponibilidade**: A infraestrutura da nuvem é projetada para alta disponibilidade (com uptime superior a 99,9%) e resiliência, proporcionando segurança e recuperação de desastres.
- Acesso **remoto**: A nuvem pode ser acessada de qualquer local com internet, facilitando o uso independentemente da localização.
- Atualizações **automáticas e licenciamento simplificado**: Atualizações de software ocorrem automaticamente, e os usuários não precisam gerenciar licenças.
- Sustentabilidade **ambiental**: A nuvem reduz a necessidade de novos equipamentos, contribuindo para a diminuição do lixo eletrônico e da pegada de carbono.

3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING

Para garantir o alto desempenho e a disponibilidade do sistema, nosso projeto na AWS utilizará uma combinação estratégica de serviços. O Amazon RDS e o Amazon Aurora fornecerão a base de dados robusta e escalável necessária para armazenar e gerenciar os dados do sistema. As instâncias EC2 oferecerão a infraestrutura de computação flexível para executar nossos aplicativos. Para garantir a saúde e o desempenho do sistema, utilizaremos o Amazon CloudWatch para monitorar continuamente a métrica chave. Além disso, o Amazon SNS nos permitirá enviar notificações e alertas em tempo real para a equipe, garantindo uma

resposta rápida a qualquer incidente. Utilizaremos os serviços da AWS como servidor, e com isso iremos hospedar a aplicação e gerenciar os dados com IAAS.

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

Iremos utilizar a AWS, pois, ela é extremamente reconhecida como a melhor escolha para um ambiente de nuvem pública, principalmente devido a três pilares essenciais: escalabilidade, segurança e custo-benefício. A escalabilidade da AWS permite que as empresas cresçam de forma ágil e eficiente, ajustando sua capacidade conforme a demanda. Com isso, é possível evitar desperdício de recursos, uma vez que a infraestrutura se adapta automaticamente a picos e quedas de utilização. A infraestrutura global da AWS, com datacenters distribuídos em várias regiões e zonas de disponibilidade, fornece uma baixa latência e alta disponibilidade, garantindo uma experiência consistente e rápida para usuários em diferentes localidades.

Em relação aos custos, a AWS utiliza um modelo de pagamento por uso (pay-as-you-go), onde os clientes pagam apenas pelos recursos que realmente consomem. Esse modelo elimina a necessidade de grandes investimentos iniciais em infraestrutura e facilita o controle financeiro, especialmente para empresas que lidam com as variações da demanda. Além disso, a AWS oferece diferentes opções de compra de instâncias, como instâncias sob demanda, reservadas e spot, que ajustam os gastos de acordo com as necessidades de cada empresa. A escala global da AWS também possibilita preços competitivos, além de uma redução frequente no custo de muitos serviços.

3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING

No projeto foi utilizado a **Infrastructure as a Service (IaaS)**, pois, a ONG precisa de maior controle sobre a infraestrutura, o IaaS pode ser uma boa opção. Nesse modelo, é possível configurar e escalar servidores virtuais, armazenamento e redes, ideal para gerenciar um site de alto tráfego ou um sistema próprio de gestão de voluntários e doações. A Amazon EC2 é um exemplo, permitindo controle e flexibilidade sobre o ambiente de hospedagem.

O **balanceamento de carga** é essencial em cloud computing, pois garante que o tráfego seja distribuído eficientemente entre múltiplos servidores ou recursos, evitando sobrecarga em um único servidor e melhorando o desempenho, a disponibilidade e a resiliência do sistema. Ele é particularmente importante para aplicações que atendem muitos usuários, como sites e plataformas de alto tráfego, onde é fundamental manter a experiência do usuário estável e rápida, além de minimizar o risco de quedas de serviço.

O projeto irá utilizar a ALB:

Application Load Balancer (ALB):

- Focado em aplicações HTTP e HTTPS e trabalha na camada 7 do modelo OSI (camada de aplicação).
- Ideal para aplicações baseadas em micro serviços e APIs, pois permite roteamento de requisições por regras, como URL, cabeçalho e informações de cookies.
- Suporta balanceamento de carga para contêineres e permite que múltiplos serviços compartilhem o mesmo ALB.

(ALB) da AWS oferece várias vantagens específicas para aplicações que utilizam HTTP e HTTPS, especialmente em arquiteturas modernas, como micro serviços e APIs. As principais vantagens do ALB incluem: **Balanceamento na Camada 7 (Camada de Aplicação), Roteamento Inteligente e Direcionamento Baseado em Regras , Suporte para Contêineres e Orquestração de Serviços**, Escalabilidade Automática, **Segurança Avançada**, Monitoramento e Integração com Ferramentas AWS, Redução de Custos e Complexidade de Infraestrutura e essas vantagens tornam o ALB ideal para aplicações modernas, como micro serviços e APIs, que requerem roteamento inteligente e configurável, suporte a contêineres e alta escalabilidade e segurança.

3.3.7 GOOGLE CLOUD ou AWS

Etapa 1: Configuração do Ambiente na AWS

A Primeira etapa é o levantamento de custo dos recursos necessários para a rodar a aplicação.

Etapa 2: Configuração da Instância EC2 (para Hospedar a API e o Sistema Angular)

1. Configurar a Instância EC2:

- **Região:** us-east-1
- **Tipo de Instância:** t2.micro
- **Sistema Operacional:** Ubuntu
- **Configuração de Segurança:**
 - Configure as **regras de segurança** para permitir acesso via HTTP e HTTPS (porta 80 e 443) para o front-end Angular e a API.
- **Volume de Armazenamento:** EBS de 8GB
- **Configuração de Uso:** 750 horas mensais (uso contínuo durante um mês)

2. Instalar e Configurar o Servidor Angular e API:

- **Conectar-se à Instância:** Depois de iniciar a instância EC2, faça uma conexão via SSH.
- **Instalar o Node.js:** Baixe e instale o Node.js, que será necessário para o servidor Angular e a API.
- **Configurar o Servidor Angular e a API:** Após a instalação, configure o servidor Angular e a API para serem executados na instância EC2.

Etapa 3: Configuração do Banco de Dados (RDS com MySQL)

1. Criar uma Instância RDS:

- **Banco de Dados:** MySQL.
- **Tipo de Instância:** "db.t2.micro"
- **Armazenamento:** 32 GB de armazenamento EBS (SSD).
- **Segurança e Backup:** Configurar a segurança para que a instância de banco de dados seja acessível somente pela instância EC2.

2. Configurar as Regras de Acesso:

- Permita o acesso da instância EC2 à instância RDS configurando o **Security Group** para a rede privada da VPC.

Etapa 4: Configuração do Serviço de Notificações (SNS)

1. Criar um Tópico SNS:

- **Configuração de Mensagens:** Configurar as mensagens para recuperação de senha.

Etapa 5: Monitoramento e Ajustes de Custos

1. Configuração do CloudWatch:

- Configure alertas básicos para acompanhar o uso de CPU e memória, especialmente da instância EC2 e do RDS.

2. Ajustes com Base no Orçamento:

- Monitore o consumo na seção de **Custos e Cobrança** do console da AWS.
- Ajustar as configurações se o consumo estiver maior que o orçamento de US\$ 100.

3.4 ESTRUTURA DE DADOS

Estruturas de Dados são fundamentais na ciência da computação para organizar, gerenciar e armazenar informações de maneira eficiente, facilitando o acesso e a manipulação desses dados em diversos sistemas. Uma estrutura de dados pode ser entendida como uma maneira organizada de armazenar e arranjar os dados na memória de um computador, possibilitando operações de busca, inserção, remoção e modificação com diferentes graus de eficiência dependendo do tipo de estrutura escolhido. Essa organização eficiente é essencial, pois o desempenho de um sistema computacional muitas vezes depende diretamente da estrutura de dados utilizada, uma vez que operações de acesso e manipulação de dados devem ser executadas com rapidez e baixo consumo de memória (Aho et al., 1983).

As estruturas de dados são geralmente classificadas em lineares e não lineares. Entre as estruturas lineares, temos os arrays, ou vetores, que armazenam dados de um mesmo tipo em posições contíguas de memória e permitem acesso direto aos elementos por meio de índices, o que facilita operações de leitura e escrita em tempo constante. Outras estruturas lineares incluem as listas, que podem ser simplesmente encadeadas, duplamente encadeadas ou circulares, possibilitando uma maior flexibilidade na inserção e remoção de elementos (Cormen et al., 2009). Também fazem parte das estruturas lineares as filas, que seguem o princípio FIFO (First In, First Out), onde o primeiro elemento a entrar é o primeiro a sair, e as pilhas, que seguem o princípio LIFO (Last In, First Out), onde o último elemento a entrar é o primeiro a sair.

Já entre as estruturas não lineares, destacam-se as árvores e os grafos. As árvores representam uma estrutura hierárquica, onde cada nó pode ter um ou mais nós filhos, formando uma estrutura em formato de árvore. Um tipo específico é a árvore binária, na qual cada nó possui até dois filhos, sendo muito utilizada em algoritmos de busca e ordenação. Por outro lado, os grafos são compostos de vértices conectados por arestas, representando relações complexas entre elementos e podendo ser utilizados em problemas de roteamento e redes de comunicação (Sedgewick & Wayne, 2011).

Além disso, as tabelas hash são uma estrutura que permite o mapeamento de chaves para valores por meio de uma função hash, facilitando a busca rápida de informações com base em uma chave única. Essa estrutura é amplamente empregada em cenários que exigem

acesso eficiente, como bancos de dados e sistemas de gerenciamento de memória (Weiss, 2012).

A escolha da estrutura de dados ideal é essencial para garantir que um algoritmo ou aplicação tenha o melhor desempenho possível em termos de uso de memória e velocidade de execução. Cada tipo de estrutura possui vantagens e limitações, e a seleção da mais apropriada depende do tipo de operação que será realizada com maior frequência. Por exemplo, estruturas como pilhas e filas são adequadas para situações onde a ordem de entrada e saída é crucial, enquanto árvores e grafos são indicados para problemas hierárquicos e de relacionamento complexo entre elementos. Estruturas de dados, portanto, desempenham um papel central na eficiência dos sistemas e algoritmos, sendo um tópico de estudo aprofundado e contínuo na ciência da computação.

3.4.1 LEVANTAMENTO DE REQUISITOS

Com base nos requisitos previamente levantados, será definida a estrutura de dados mais adequada para cada módulo do sistema. As decisões sobre as estruturas a serem utilizadas devem considerar os tipos de dados, a frequência das operações (como inserção, remoção, busca, etc.) e os requisitos de desempenho, como tempo de resposta e uso de memória.

Serão abordadas as opções mais eficientes em termos de complexidade algorítmica e adequação ao contexto do projeto, como listas, pilhas, filas, árvores, grafos, tabelas de dispersão, entre outras, dependendo da necessidade do sistema.

Requisitos Funcionais:

Gerenciar cadastro de animais

- Cadastrar novos animais, incluindo as informações de nome, raça, tamanho, status do animal, descrição do animal.
- consultar informações dos animais, e podendo ir diretamente nas específicas, como raça, status.
- Atualizar informações de animais existentes.
- Remover registros dos animais.

Gerenciar Adoção dos animais

- Registra o status da adoção, raça do animal, data da adoção.
- Consultar o histórico da data de adoção e por quem adotou.

Gerenciar Doações

- Gerenciar as doações recebidas, adicionando o que foi recebido e utilizado.
- valor da doação, data da doação, quem fez a doação, descrição do que foi doado.

Gerenciar Despesas

- Valor gasto, descrição do que foi gasto, datas incluindo vencimento, saída, pagamento e status das despesas.
- Gerenciar as despesas do ONG

Gerenciar Relatórios Clínicos

- Data de registro, descrição médica, animal usuário que foi consultado.
- Gerenciar todo relatório médico do animal.

Requisitos não funcionais:

Monitoramento de Desempenho:

- Integrar a API com Amazon CloudWatch para monitorar métricas como tempo de resposta, consumo de CPU, e falhas.

Escalabilidade

- O sistema deve ser capaz de escalar horizontalmente para suportar picos de uso, especialmente em eventos, como campanhas de adoção.
- Sugestão: usar Amazon DynamoDB com Auto Scaling habilitado.

Desempenho

- As requisições da API devem ter um tempo médio de resposta menor que 200ms para operações de leitura em condições normais de uso.

3.4.2 VALIDAÇÃO DOS REQUISITOS

A definição de requisitos funcionais e não funcionais para sistemas computacionais é essencial para garantir que a solução tecnológica atenda às expectativas do cliente e funcione adequadamente em termos de desempenho, escalabilidade e segurança. Os requisitos funcionais descrevem **o que o sistema deve fazer**, ou seja, as operações e comportamentos esperados diretamente relacionados à funcionalidade principal. Por outro lado, os requisitos não funcionais abordam **como o sistema deve se comportar**, destacando características de qualidade, desempenho e restrições técnicas que garantem a usabilidade e a eficiência da solução (Pressman, 2010).

No contexto da estrutura de dados, os requisitos funcionais estão diretamente relacionados às operações que os usuários realizarão. Por exemplo, o sistema deve permitir o registro de dados como informações de animais, incluindo nome, idade, tipo, status de adoção e histórico médico, além de suportar a atualização e exclusão de registros. Também é essencial que possibilite buscas e consultas avançadas, como filtrar animais por tipo ou faixa etária, e que gere relatórios detalhados, como o número de adoções realizadas em um período específico (Sommerville, 2011).

Já os requisitos não funcionais estabelecem parâmetros de qualidade para essas funcionalidades. No caso de uma API que manipula dados de animais para adoção, o desempenho é um requisito crucial. As consultas devem ser processadas em menos de 200ms para leitura simples e em até 500ms para operações mais complexas, como buscas com múltiplos filtros. Além disso, a escalabilidade da estrutura de dados é indispensável, garantindo que o sistema suporte o crescimento no número de registros sem prejuízo de desempenho. A flexibilidade também é importante, permitindo que novos campos sejam adicionados aos registros sem a necessidade de alterações estruturais no banco de dados, algo comum em sistemas baseados em NoSQL, como o DynamoDB (Harrison, 2020).

Outro requisito não funcional relevante é a segurança. Dados sensíveis devem ser protegidos por criptografia em trânsito e em repouso, e o acesso às operações de criação, atualização e exclusão deve ser restrito a usuários autorizados. A integridade dos dados também é crucial: transações, como a mudança de status de um animal para "adotado", devem ser consistentes e refletidas corretamente em todas as buscas e relatórios gerados (Silberschatz et al., 2020). Além disso, a disponibilidade do sistema deve ser garantida com um uptime de pelo menos 99,9%, o que pode ser alcançado com estratégias de replicação e uso de serviços em nuvem.

Por fim, os requisitos funcionais e não funcionais não são independentes, mas complementares. Enquanto os primeiros determinam as funcionalidades que o sistema deve oferecer, os segundos asseguram que essas funcionalidades sejam entregues com qualidade e eficiência, atendendo tanto às expectativas do cliente quanto às demandas técnicas e operacionais do projeto.

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS

A Formação para a Vida é um dos eixos do Projeto Pedagógico de Formação por Competências da UNIFEQB.

Esta parte do projeto está diretamente relacionada com a extensão universitária, ou seja, o objetivo é que seja aplicável e que tenha real utilidade para a sociedade, de um modo geral.

3.5.1 ENFRENTANDO ESTEREÓTIPOS

Está disponível para os estudantes no Classroom, o tema “Enfrentando estereótipos”.

Nesta parte do Projeto, os estudantes deverão realizar uma síntese dos 4 (quatro) tópicos deste tema, quais sejam:

- **Tópico 1:** Estereótipo e convívio social

Estereótipos são crenças simplificadas sobre grupos de pessoas que influenciam o convívio social, levando a preconceitos e discriminações. Por exemplo, a ideia de que homens são mais racionais que mulheres pode afetar decisões no trabalho, enquanto estereótipos raciais, como associar determinadas etnias à criminalidade, geram discriminação e desconfiança. Da mesma forma, acreditar que pessoas mais velhas não se adaptam a tecnologias pode excluí-las de oportunidades valiosas, e a noção de que indivíduos de classes mais baixas são menos educados pode criar barreiras em ambientes sociais. Desconstruir esses estereótipos é essencial para promover relacionamentos mais inclusivos e empáticos, onde cada pessoa é valorizada por suas singularidades.

- **Tópico 2:** Estereótipo e representação

Estereótipos e representações estão interligados na forma como grupos de pessoas são retratados na mídia e no cotidiano, moldando percepções e comportamentos. Estereótipos são simplificações que podem gerar preconceitos. Por exemplo, filmes muitas vezes mostram mulheres como personagens submissas, reforçando visões limitadas de gênero. Anúncios que ligam homens à tecnologia e mulheres a beleza perpetuam ideias sobre interesses naturais. Na educação, expectativas raciais podem afetar o desempenho e a autoestima de alunos de diferentes etnias. Além disso, estereótipos de classe social podem levar a julgamentos em situações cotidianas, como no atendimento em lojas. Desconstruir esses estereótipos e promover representações mais autênticas são essenciais para uma sociedade mais inclusiva e justa.

- **Tópico 3:** Troco likes: a idealização da vida na internet

"Troco Likes: a idealização da vida na internet" discute como as redes sociais criam uma percepção distorcida da realidade, onde a busca por aprovação em forma de likes gera pressão para apresentar uma vida perfeita. Influenciadores digitais frequentemente mostram estilos de vida luxuosos, criando expectativas irreais entre seguidores. O uso de filtros e edições em fotos provoca inseguranças sobre a aparência real, enquanto a tendência de compartilhar apenas os melhores momentos da vida leva a comparações e desvalorização da rotina cotidiana. Além disso, a validação através de likes pode priorizar conteúdo superficial em detrimento de interações autênticas. Essa idealização pode impactar a autoestima e a saúde mental, ressaltando a importância de um uso equilibrado das redes sociais.

- **Tópico 4:** Convivendo com a diferença

"Convivendo com a diferença" enfatiza a importância da diversidade e aceitação em um mundo plural. Aprender a lidar com diferenças culturais, étnicas, de gênero e de habilidades enriquece as relações sociais e promove inclusão. Por exemplo, em ambientes de trabalho, equipes diversas geram mais inovação e melhores soluções. Nas escolas, a inclusão de alunos com necessidades especiais ensina respeito e valorização das habilidades individuais. Festivais culturais celebram tradições de diferentes grupos étnicos, quebrando preconceitos e

promovendo o respeito mútuo. Além disso, redes sociais que incentivam diálogos sobre temas como racismo e desigualdade ampliam a empatia entre os grupos. A convivência com a diferença é fundamental para construir uma sociedade mais justa, onde cada indivíduo é valorizado por suas particularidades.

3.5.2 ESTUDANTES NA PRÁTICA

Segue abaixo o link para acesso do podcast.

https://drive.google.com/drive/folders/197_4o9uWo5jaK4Y78MY8eb8jCfBC9ua5?usp=sharing

4. CONCLUSÃO

Após um processo de desenvolvimento cuidadoso e dedicado, é com satisfação que concluímos a criação deste sistema destinado a aprimorar a gestão e administração da ONG APAN. Esta jornada representou não apenas um marco em nossa busca pela modernização e eficiência de nossos processos internos, mas também reflete nosso compromisso contínuo com o bem-estar dos animais e o sucesso de nossa missão. Ao longo deste projeto, dedicamo-nos a criar uma ferramenta que não apenas simplifica tarefas administrativas, como também permite uma gestão mais eficaz e centrada nas necessidades dos animais que cuidamos. Desde o gerenciamento de doações até o acompanhamento de resgates e adoções, este sistema foi concebido para oferecer praticidade e precisão em cada etapa do processo.

A implementação deste sistema representa não apenas um avanço tecnológico em nossa organização, mas também uma demonstração tangível do potencial transformador da tecnologia na gestão de ONGs. Estamos confiantes de que essa iniciativa nos permitirá alcançar novos patamares de eficiência e impacto, capacitando-nos a fazer ainda mais em prol dos animais que servimos. À medida que avançamos para a próxima fase de implementação e integração deste sistema em nossas operações diárias, estamos entusiasmados com as possibilidades que se apresentam. Com uma gestão mais precisa e eficiente ao nosso alcance, estamos preparados para enfrentar os desafios do futuro com confiança e determinação, mantendo nosso foco inabalável em nossa missão e no bem-estar dos animais que dependem de nós.

Em resumo, este sistema não é apenas uma ferramenta tecnológica, mas sim um reflexo do nosso compromisso inabalável com a causa animal e nossa determinação em fazer a diferença em suas vidas. Estamos gratos por esta oportunidade de avançar e evoluir, e ansiosos para ver os frutos deste trabalho em ação, transformando positivamente a realidade dos animais e daqueles que se dedicam a protegê-los.

REFERÊNCIAS

1. Sebesta, R. W. (2019). Concepts of Programming Languages. 12^a ed. Boston: Pearson.
2. Knuth, D. E. (1997). The Art of Computer Programming. Vol. 1-4. Addison-Wesley.
3. Elmasri, R., & Navathe, S. B. (2015). Fundamentals of Database Systems. 7^a ed. Pearson.
4. Chen, M., Mao, S., & Liu, Y. (2012). Big Data: A Survey. Mobile Networks and Applications, 19(2), 171-209.
5. Gray, J., & Reuter, A. (1993). Transaction Processing: Concepts and Techniques. Morgan Kaufmann.
6. Schneier, B. (2015). Cryptography and Network Security: Principles and Practice. 7^a ed. Pearson.
7. Garcia-Molina, H., Ullman, J. D., & Widom, J. (2009). Database Systems: The Complete Book. 2^a ed. Pearson.
8. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50-58.
9. Chaffey, D. (2019). *Digital Marketing: Strategy, Implementation, and Practice*. Pearson.
10. Kumar, R., & Singh, S. (2017). Cloud computing in non-profit organizations: Benefits and challenges. *Journal of Cloud Computing: Advances, Systems and Applications*, 6(1), 1-13.
11. Marston, S., Li, Z., Bandyopadhyay, S., & Zhang, J. (2011). Cloud computing—The business perspective. *Decision Support Systems*, 51(1), 176-189.
12. Rao, M. (2016). *Cloud Computing: A Practical Approach for Learning and Adoption*. Springer.

13. Zhang, L., Cheng, P., & Boutaba, R. (2010). Cloud computing: State-of-the-art and research challenges. **Journal of Internet Services and Applications**, 1(1), 7-18.
14. Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1983). *Data Structures and Algorithms*. Addison-Wesley.
15. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.
16. Sedgewick, R., & Wayne, K. (2011). *Algorithms* (4th ed.). Addison-Wesley.
17. Weiss, M. A. (2012). *Data Structures and Algorithm Analysis in Java* (3rd ed.). Pearson.

