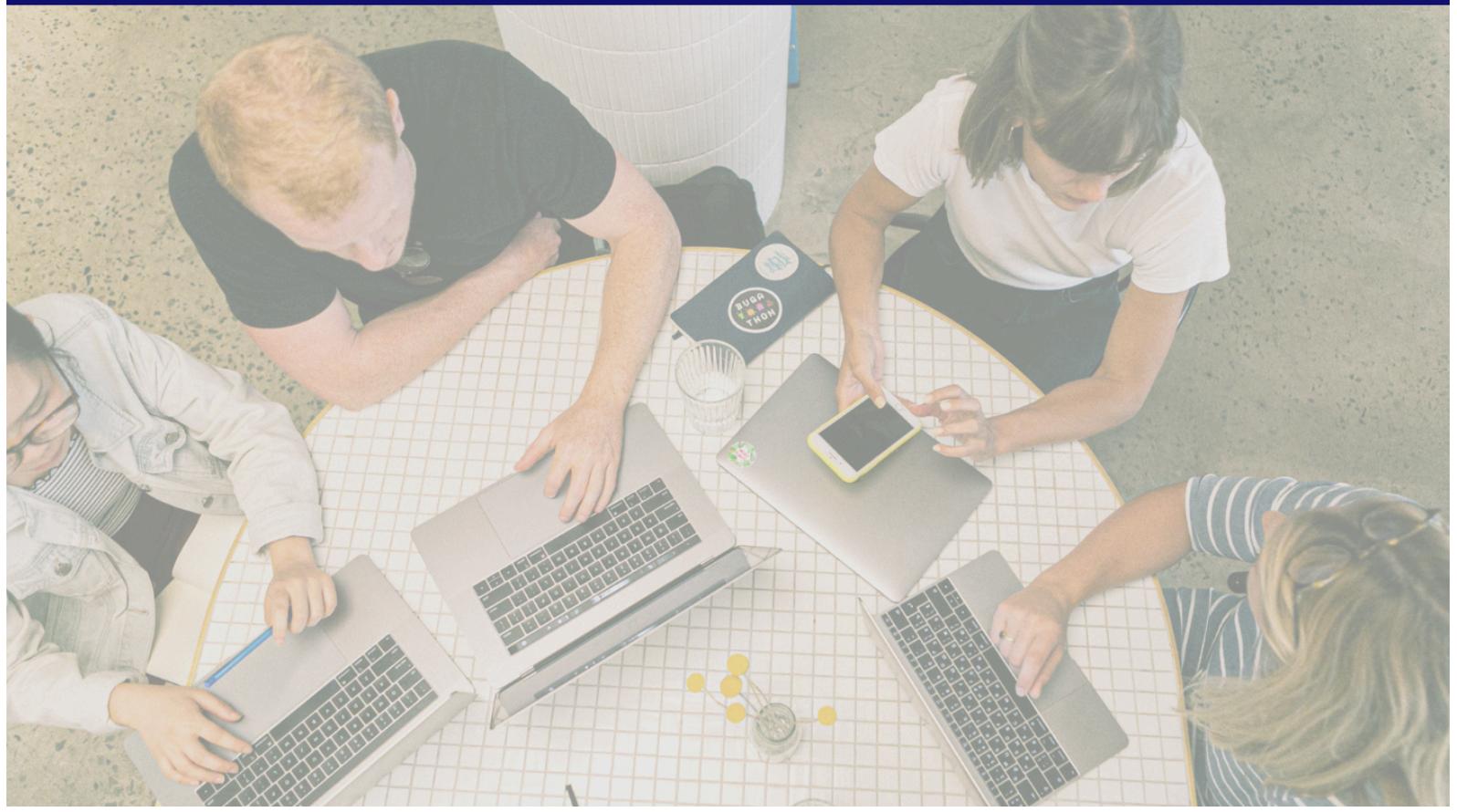




UNifeob
| ESCOLA DE NEGÓCIOS

2024

PROJETO INTEGRADO



UNIFEOB

**CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO
OCTÁVIO BASTOS**

ESCOLA DE NEGÓCIOS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

CIÊNCIA DA COMPUTAÇÃO

**SISTEMA DE GESTÃO E INTELIGÊNCIA DE NEGÓCIOS
PARA ORGANIZAÇÕES SOCIAIS**

CAACCH

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
CAACCH

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:

Augusto Bueno Sossai, RA 23000707

Felipe Assempção Estorari, RA 23000880

Gustavo Frutuoso, RA 23000595

João Lucas De Souza Domingos, RA 23000260

José Eduardo Teixeira, RA 23000271

Lucas Restani Ribeiro, RA 23000879

Samuel Pontes Portela, RA 23000540

SÃO JOÃO DA BOA VISTA, SP
NOVEMBRO 2024

SUMÁRIO

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	7
3.1.2 MODELO FÍSICO	8
3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO	9
3.2.1 APPLICATION PROGRAMMING INTERFACE (API) - BACK-END.	9
3.2.2 FRONT-END	11
3.3 COMPUTAÇÃO EM NUVEM	13
3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING	13
3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO	14
3.3.3 VANTAGENS DA CLOUD COMPUTING	14
3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING	16
3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)	17
3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING	18
3.3.7 GOOGLE CLOUD ou AWS	18
3.4 ESTRUTURA DE DADOS	19
3.4.1 LEVANTAMENTO DE REQUISITOS	19
3.4.2 VALIDAÇÃO DOS REQUISITOS	20
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS	21
3.5.1 ENFRENTANDO ESTEREÓTIPOS	21
3.5.2 ESTUDANTES NA PRÁTICA	21
4. CONCLUSÃO	23
REFERÊNCIAS	24
ANEXOS	26

1. INTRODUÇÃO

O Projeto Integrado continua a desenvolver o tema “Sistema de Gestão e Inteligência de Negócios para Organizações Sociais”, com foco na criação de um software personalizado para a Organização Social CAACCH. O sistema visa proporcionar um gerenciamento eficiente e otimizado de dados, essencial para o funcionamento diário da organização. A solução proposta não só irá facilitar o controle de informações como doações, cadastros de beneficiários e movimentações de estoque, mas também garantirá que essas informações possam ser acessadas de maneira rápida, segura e eficaz, otimizando a gestão administrativa e operacional da CAACCH.

Além disso, o projeto busca automatizar processos que, muitas vezes, são realizados manualmente, minimizando erros e aumentando a precisão das informações, o que contribuirá diretamente para a melhoria da tomada de decisões dentro da organização. A eficiência do sistema se traduz na capacidade de gerar relatórios dinâmicos e dashboards personalizados, que permitirão aos gestores da CAACCH monitorar, em tempo real, o fluxo de doações, o perfil dos beneficiários atendidos e a situação do estoque, entre outros dados relevantes. A transparência também será um ponto-chave, uma vez que o sistema facilitará o acompanhamento das atividades pela própria equipe da CAACCH, voluntários e até mesmo pelo público externo, quando necessário.

A abordagem técnica do projeto será baseada em uma arquitetura robusta e escalável, com o uso de APIs para o desenvolvimento do back-end. O front-end será construído com uma interface intuitiva e amigável, priorizando a experiência do usuário, o que facilitará o treinamento e a adaptação da equipe da CAACCH. A escolha do ambiente de desenvolvimento será focada na utilização de cloud computing, aproveitando as vantagens oferecidas pela nuvem, como escalabilidade, flexibilidade, segurança e disponibilidade. A análise cuidadosa entre provedores de nuvem, como Google Cloud e AWS, será realizada para garantir que a solução escolhida atenda não apenas às necessidades da CAACCH, mas também aos padrões de segurança e desempenho exigidos pelo projeto.

O levantamento e validação de requisitos será uma etapa crucial do desenvolvimento e será realizado utilizando TypeScript, que garantirá a criação de um código robusto e de fácil manutenção. Isso assegura que o sistema seja moderno, ágil e capaz de evoluir conforme as futuras necessidades da organização. Durante o processo de desenvolvimento, serão também realizados testes contínuos para garantir a qualidade e a confiabilidade do software.

Além de melhorar a gestão interna da CAACCH, o sistema também oferecerá uma plataforma de análise de dados, com recursos de Business Intelligence (BI), para que a organização possa avaliar seus resultados, identificar padrões e tomar decisões estratégicas com base em dados concretos. A implementação de ferramentas analíticas e relatórios gerenciais também permitirá um melhor planejamento das campanhas de arrecadação, otimização de recursos e gestão das atividades de apoio aos beneficiários.

Ao final, o projeto visa entregar não apenas uma solução tecnológica, mas uma ferramenta estratégica que maximizará a eficiência operacional da CAACCH, melhorará a transparência das suas operações e facilitará a análise de dados de forma ágil e precisa. Com isso, espera-se gerar um impacto positivo significativo na gestão da organização, contribuindo para o cumprimento de sua missão social de maneira mais eficaz e com maior alcance.

2. DESCRIÇÃO DA EMPRESA

O Centro de Apoio e Assistência à Criança e ao Adolescente com Humanismo de São João da Boa Vista (CAACCH) é uma entidade educacional sem fins lucrativos, registrada sob o CNPJ nº 03.295.254/0001-72. Localizada na Rua Marechal Deodoro, 244, no Bairro Centro da cidade de São João da Boa Vista, São Paulo, a organização atua no campo da assistência social.

A instituição se dedica a promover o bem-estar e o desenvolvimento integral de crianças e adolescentes, oferecendo oportunidades de educação, saúde, cultura e lazer. Além disso, a CAACCH apoia as famílias dos jovens atendidos, com o objetivo de fortalecer os vínculos familiares e melhorar a qualidade de vida de todos os envolvidos.

3. PROJETO INTEGRADO

Durante as aulas da disciplina de Tópicos Avançados de Banco de Dados ministradas pelo professor Max Streicher Vallim, a equipe estudou o uso de bancos de dados na nuvem, MongoDB, para acesso seguro e eficiente a dados de atendimentos. O modelo lógico organiza entidades e relacionamentos sem depender de um sistema específico, enquanto o modelo físico detalha a implementação, com tabelas como Doações e Estoque. Essa estrutura assegura eficiência, segurança e flexibilidade ao sistema de dados.

Nas aulas de Linguagem e Técnica de Programação, foram abordados tópicos e técnicas avançadas de programação para melhorar modularidade e desempenho, além de APIs RESTful seguras e escaláveis. No front-end, destaca design responsivo, o uso de frameworks modernos como Angular para criar interfaces dinâmicas.

Já durante as aulas de Computação em Nuvem, exploramos como a nuvem oferece infraestrutura escalável e segura para o desenvolvimento de sistemas, utilizando plataformas como AWS. Estudamos a centralização de dados, flexibilidade e escalabilidade, destacando os benefícios financeiros e operacionais. Discutimos também o uso de serviços como Amazon EC2, RDS e S3, e como esses recursos permitem otimizar o desempenho, garantir alta disponibilidade e oferecer soluções eficientes para organizações em crescimento.

3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS

Para o gerenciamento de dados da CAACCH, a equipe decidiu utilizar o banco de dados MySQL devido à sua estrutura eficiente, que oferece acesso seguro e rápido às informações sobre doações, estoque de produtos e solicitações. Além disso, utilizando as funções avançadas do MySQL, podemos criar automações que realizam atualizações automáticas em tabelas específicas, otimizando o gerenciamento e a integridade dos dados.

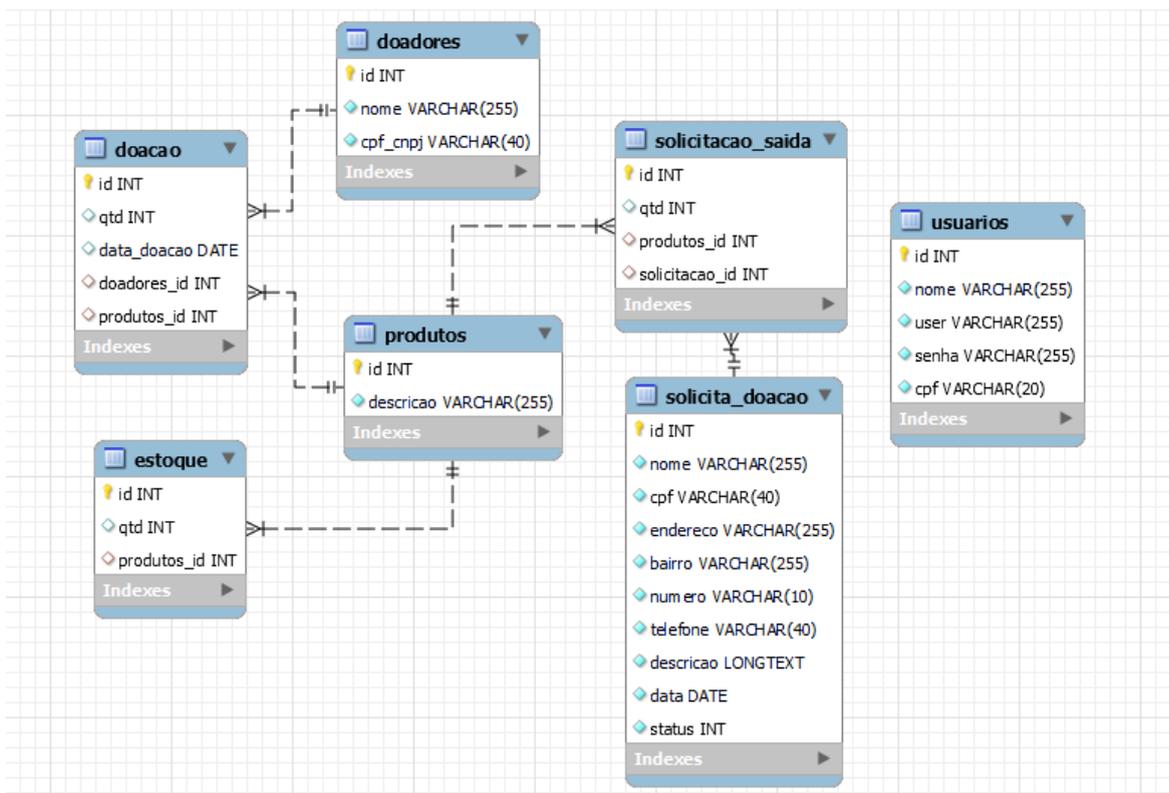
Com a implementação e testes do banco de dados em um ambiente de desenvolvimento, conseguimos garantir a eficácia da solução. Como resultado, o banco de dados foi migrado para o Amazon RDS (Relational Database Service), um serviço gerenciado da Amazon Web Services (AWS). Esse serviço possibilita hospedar o banco de dados em uma infraestrutura de alto desempenho e alta disponibilidade, garantindo excelente performance, escalabilidade e segurança para o nosso sistema.

3.1.1 MODELO LÓGICO

“Um modelo de dados lógico (MDL) é um tipo de modelo de dados que descreve elementos de dados em detalhes e é usado para desenvolver entendimentos visuais de entidades de dados, atributos, chaves e relacionamentos. Este tipo de modelo é exclusivamente independente de um banco de dados específico, a fim de estabelecer uma estrutura fundacional para componentes da camada semântica em sistemas de gerenciamento de dados.”

No modelo lógico, utilizamos o MySQL Workbench para desenvolver as tabelas, definir os atributos e estabelecer os relacionamentos entre as entidades. A partir dessa ferramenta, foi possível projetar a estrutura do banco de dados de maneira visual e eficiente, garantindo que todos os aspectos do modelo relacional fossem adequadamente representados. A seguir, foi desenvolvida a seguinte estrutura:

Imagem 1 : Modelo Lógico do banco CAACCH.



Fonte: Autores (2024).

Doacao: para registrar as doações realizadas e seus detalhes.

Doadores: para armazenar informações sobre os doadores que contribuem com a organização.

Solicitacao_saida: para gerenciar as solicitações de retirada de produtos do estoque.

Usuarios: para controlar o acesso dos usuários ao sistema.

Solicita_doacao: para registrar as solicitações de doações feitas por famílias ou outras entidades.

Produtos: para detalhar os produtos que são recebidos, doados ou armazenados.

Estoque: para monitorar a quantidade e o status dos produtos disponíveis.

O modelo lógico do banco de dados e sua estrutura asseguram a integridade e a organização dos dados, garantindo que os relacionamentos entre as tabelas sejam bem definidos.

3.1.2 MODELO FÍSICO

No desenvolvimento do projeto de software para a CAACCH, a criação do modelo físico do banco de dados é fundamental. Nessa fase, definem-se os detalhes técnicos da implementação, como o armazenamento dos dados e a criação de objetos do banco (tabelas, visões, funções, triggers, stored procedures). Também são estabelecidas as permissões de acesso, garantindo a segurança e integridade dos dados.

Com a intenção de melhorar a eficiência de inserção de dados implementamos uma trigger que faz a inserção de dados na tabela de estoque de forma automatizada.

Imagem 2 : Trigger de inserção

```
CREATE TRIGGER after_doacao_insert
AFTER INSERT ON doacao
FOR EACH ROW
BEGIN
    DECLARE produto_existe INT;

    -- Verifica se o produto existe na tabela estoque
    SELECT COUNT(*)
    INTO produto_existe
    FROM estoque
    WHERE produtos_id = NEW.produtos_id;

    -- Se o produto já existe, atualiza a quantidade
    IF produto_existe > 0 THEN
        UPDATE estoque
        SET qtd = qtd + NEW.qtd
        WHERE produtos_id = NEW.produtos_id;
    ELSE
        -- Caso contrário, insere o novo produto na tabela estoque
        INSERT INTO estoque (produtos_id, qtd)
        VALUES (NEW.produtos_id, NEW.qtd);
    END IF;
END//
DELIMITER ;
```

Essa trigger verifica se o item já está cadastrado no banco de dados. Se o item estiver cadastrado, ela atualiza a quantidade com o comando UPDATE; caso contrário, ela insere o novo produto automaticamente no estoque. A trigger será disparada após a inserção de um registro na tabela de doações, criando assim um controle de estoque automático.

Além disso, também implementamos procedures para inserção de doador, produto e solicitação, o que tornou a parte de inserção e manipulação de dados mais fácil e intuitiva.

Imagem 3 : Trigger de inserção

```
Corpo da rotina:
1 BEGIN
2     INSERT INTO solicita_doacao (nome, cpf, endereco, bairro, numero, telefone, descricao, data, status)
3     VALUES (p_nome, p_cpf, p_endereco, p_bairro, p_numero, p_telefone, p_descricao, p_data, p_status);
4 END
```

Fonte: Autores (2024).

3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO

“A programação orientada a objetos (POO) surgiu como um paradigma que apresentava uma disciplina para novos projetos de desenvolvimento de software. Etapas com análise e projeto de software que, em outros paradigmas, acabavam sendo minimizados na POO, se tornam requisitos fundamentais. “(SILVA, Fabricio M.; LEITE, Márcia C D.; OLIVEIRA, Diego B. Paradigmas de programação. p. 59)

No projeto da CAACCH, utilizamos a linguagem de programação JavaScript juntamente com o framework Node.js para o desenvolvimento do backend. Adotamos o paradigma de orientação a objetos para melhorar a organização e a reutilização dos dados.

Para o frontend, utilizamos o EJS (Embedded JavaScript), uma ferramenta de template que permite gerar páginas dinâmicas no servidor. O EJS facilita a inserção de dados diretamente no HTML, permitindo que o conteúdo seja gerado dinamicamente com base nas variáveis passadas pelo servidor. Com isso, conseguimos consumir a API e apresentar os

resultados em formato JSON nas interfaces visuais da nossa página de maneira eficiente e interativa.

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

“No início da história dos computadores, quem se achesse a utilizar um computador era obrigado a ter um conhecimento profundo das entranhas deste e seus sistemas complexos. Portanto, um usuário que necessitava realizar operações consideradas finais nos computadores deveria ter entendimento de toda a complexidade deles. Por operações finais, entenda qualquer atividade que deveria ser simples ao usuário final, que só deseja ou necessita utilizar o sistema e não ter conhecimento completo de todas as vertentes intrínsecas ao ambiente computacional inserido.”(BARRETO, Jeanine dos S.; JR., Paulo A P.; BARBOZA, Fabrício F M.; et al. Interface humano-computador. p 13).

No desenvolvimento de uma API para o "Sistema de Gestão da CAACCH", foram desenvolvidos diversos aspectos fundamentais sobre a API. Inicialmente, foram explorados os conceitos de arquitetura RESTful, que são essenciais para a criação de APIs escaláveis e de fácil manutenção. Além disso, foi discutida a importância da autenticação e autorização para garantir a segurança dos dados sensíveis das organizações sociais.

Foram analisadas a utilização de boas práticas de desenvolvimento, como a implementação de testes automatizados e a adoção de padrões de versionamento através do GitHub, para assegurar a confiabilidade e a evolução contínua do sistema.

Neste projeto, foi utilizado o framework de JavaScript Node.js, que proporciona uma plataforma eficiente e de alto desempenho para o desenvolvimento de aplicações web. Junto ao Node.js, várias bibliotecas foram empregadas para otimizar a funcionalidade e segurança do sistema. O Express, por exemplo, facilita a criação de servidores web e rotas, enquanto o Nodemon automatiza o processo de reinicialização do servidor durante o desenvolvimento, permitindo um fluxo de trabalho mais ágil.

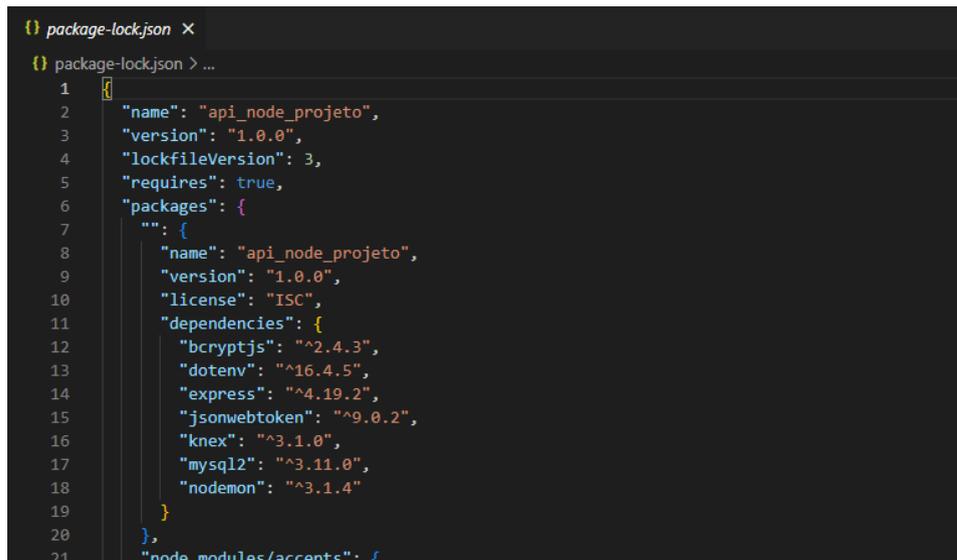
Imagem 4 : Trecho de código de Login CAACCH.

```
JS UsuariosController.js X
src > controllers > JS UsuariosController.js > UsuariosController
1  const Usuarios = require('../models/Usuarios');
2  const bcrypt = require('bcryptjs');
3  const jwt = require('jsonwebtoken');
4
5  class UsuariosController{
6
7      async login(req,res){
8          let {email, senha} = req.body
9          let user = await Usuarios.findByUser(email)
10         if (!user.status){
11             user.err === undefined
12             ? res.status(406).json({sucess: user.status, message: 'E-mail não encontrado'})
13             : res.status(404).json({sucess: user.status, message: user.err})
14         }else{
15
16             let isPassword = bcrypt.compareSync(senha, user.values.senha)
17
18             if (isPassword){
19
20                 let token = jwt.sign({email: user.values.user, role: user.values.senha},process.env.SEC
21                 res.status(200).json({sucess: isPassword, token: token})
22
23             }else{
24                 res.status(406).json({sucess: isPassword, message: 'Senha Inválida'})
25             }
26         }
27     }
28 }
```

Fonte: Autores (2024).

A biblioteca mysql2 foi escolhida para interação com o banco de dados MySQL, oferecendo uma interface moderna e rápida para realizar consultas e manipulações de dados. Para simplificar e abstrair as consultas SQL, utilizamos o Knex.js, que é um construtor de consultas SQL baseado em JavaScript, permitindo escrever consultas de maneira mais organizada e segura, além de garantir compatibilidade com diferentes bancos de dados. Para garantir a segurança das informações sensíveis, a biblioteca bcrypt é utilizada para criptografar senhas e outros dados confidenciais. Já o dotenv é empregado para gerenciar variáveis de ambiente, permitindo que dados como credenciais de banco de dados ou chaves de API sejam mantidos de forma segura e facilmente configuráveis em diferentes ambientes de desenvolvimento e produção.

Imagem 5 : Bibliotecas Integradas CAACCH.



```
package-lock.json X
package-lock.json > ...
1  {}
2  "name": "api_node_projeto",
3  "version": "1.0.0",
4  "lockfileVersion": 3,
5  "requires": true,
6  "packages": {
7    "": {
8      "name": "api_node_projeto",
9      "version": "1.0.0",
10     "license": "ISC",
11     "dependencies": {
12       "bcryptjs": "^2.4.3",
13       "dotenv": "^16.4.5",
14       "express": "^4.19.2",
15       "jsonwebtoken": "^9.0.2",
16       "knex": "^3.1.0",
17       "mysql2": "^3.11.0",
18       "nodemon": "^3.1.4"
19     }
20   },
21   "node_modules/accents": {
```

Fonte: Autores (2024).

3.2.2 FRONT-END

No desenvolvimento front-end deste projeto, foi adotado o EJS (Embedded JavaScript) para renderizar páginas dinâmicas no lado do servidor, permitindo que o conteúdo fosse gerado de forma flexível e eficiente. O EJS possibilita a inclusão de variáveis JavaScript diretamente no código HTML, garantindo uma maior organização e clareza no desenvolvimento. Isso resultou em uma estrutura mais limpa e fácil de manter, já que o código HTML e as lógicas de exibição são separadas de maneira clara.

Uma das vantagens do EJS é sua capacidade de integrar dados dinâmicos de forma simples, o que se mostrou essencial para garantir uma aplicação responsiva e funcional. O EJS pode facilmente interagir com APIs e manipular dados em tempo real, permitindo que a página se adapte a diferentes dispositivos como computadores, tablets e smartphones, sem a necessidade de reescrever o código para cada plataforma.

O objetivo principal ao usar o EJS no front-end foi otimizar a experiência do usuário (UX) e a interface do usuário (UI). O design das páginas foi pensado para ser intuitivo e atraente, proporcionando uma navegação fluida e de fácil usabilidade. O EJS contribuiu diretamente para isso, ao permitir a inserção dinâmica de conteúdo.

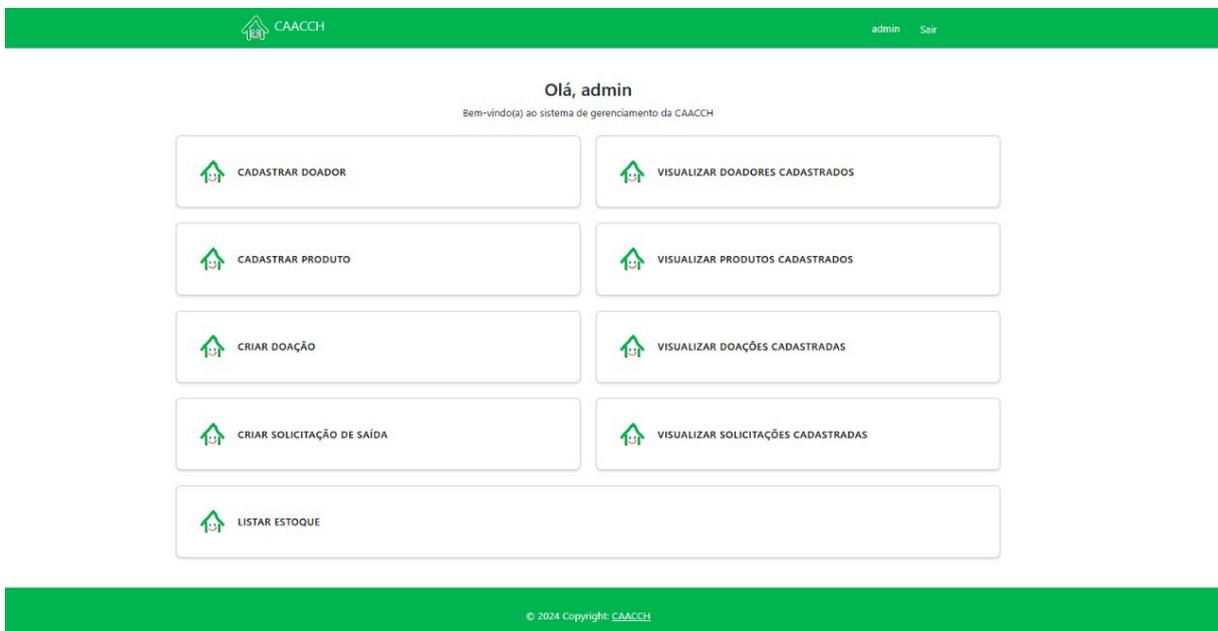
Imagem 6 : Estrutura usada para o desenvolvimento do front-end

```
EXPLORER
PROJETO_CAACH_FRONTEND
  > node_modules
  > projeto_caach_frontend
  > public
  > views
    <> criar-doacao.ejs
    <> criar-doador.ejs
    <> criar-produto.ejs
    <> criar-solicitacao-doacao.ejs
    <> listar-doacoes.ejs
    <> listar-doadores.ejs
    <> listar-estoque.ejs
    <> listar-produtos.ejs
    <> listar-solicitacoes-cadastradas.ejs
    <> login.ejs
    <> menu.ejs
  JS app.js
  {} package-lock.json
  {} package.json

JS app.js > app.get('/listar-estoque') callback
573 app.get('/listar-solicitacoes-cadastradas', async (req, res) => {
587   const solicitacoesSaidaPessoa = await axios.get('http://50.17.89.123:4040/find-solici
590   );
591   });
592
593   const produto = await axios.get('http://50.17.89.123:4040/find-produtos', {
594     headers: {
595       'Authorization': `Bearer ${authToken}`
596     }
597   });
598
599   console.log('Resposta completa:', solicitacoesDoacaoQuantidade.data);
600   console.log('Resposta completa:', solicitacoesSaidaPessoa.data);
601   console.log('Resposta completa:', produto.data);
602
603   res.render('listar-solicitacoes-cadastradas', { nome: nome, authToken: authToken, pesso
604   });
605
606 app.get('/listar-estoque', async (req, res) => {
607   const authToken = req.cookies.auth_token;
608   const nome = req.cookies.nome;
609
610   if (!authToken) {
611     return res.redirect('/login');
612   }
}
```

Fonte: Autores (2024).

Imagem 7 : Menu interativo do sistema.



Fonte: Autores (2024).

3.3 COMPUTAÇÃO EM NUVEM

“A computação em nuvem é, em sua essência, uma forma de prestação de serviço. Como qualquer outro tipo de serviço que desejamos contratar ou terceirizar (relacionado ou não a TI), compreendemos que seremos confrontados por um mercado composto por prestadores de serviço de qualidade e confiabilidade variáveis. “(ERL, Thomas; MONROY, Eric B. Computação em Nuvem: Conceitos, Tecnologia, Segurança e Arquitetura. 2nd ed. Porto Alegre. p 1)

O grupo fará uso da computação em nuvem para o desenvolvimento do projeto, aproveitando os recursos oferecidos pela AWS (Amazon Web Services). A nuvem será a base para o funcionamento do sistema, proporcionando uma infraestrutura escalável, segura e de alta disponibilidade. Entre os principais serviços utilizados, destacam-se o Amazon EC2 para a execução do back-end, o Amazon RDS para o gerenciamento do banco de dados relacional e o Amazon S3 para o armazenamento de arquivos e dados não estruturados. O uso dessas tecnologias permitirá que o sistema tenha um desempenho otimizado, com a capacidade de escalar conforme a demanda, garantindo alta confiabilidade e segurança para o gerenciamento das informações da organização. A escolha da AWS como plataforma base visa garantir que a solução seja eficiente, robusta e preparada para atender às necessidades crescentes da CAACCH.

3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING

O projeto de computação em nuvem visa otimizar a infraestrutura da empresa, reduzindo custos com servidores físicos e oferecendo escalabilidade. A centralização dos dados na nuvem permite acesso remoto seguro, melhor colaboração e proteção contra perdas. Com a AWS, usamos o EC2 para gerenciar a API e o front-end, oferecendo flexibilidade e escalabilidade automática conforme a demanda. O RDS simplifica a administração do banco de dados, garantindo alta disponibilidade e fácil escalabilidade. Já o S3 oferece armazenamento seguro e escalável para arquivos, com custo controlado conforme o uso. Com esses serviços, a empresa reduz custos operacionais, melhora a segurança e se prepara para crescer sem a necessidade de grandes investimentos em infraestrutura física.

3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO

A utilização da computação em nuvem no projeto traz diversas vantagens práticas para a CAACCH, especialmente em termos de centralização de informações. Dados como doações, cadastros de famílias e outras informações essenciais serão armazenados de forma centralizada, permitindo que os membros autorizados acessem as informações em tempo real, de qualquer lugar, e sem risco de duplicidade. Isso facilita a tomada de decisões rápidas e o acesso remoto, ideal para uma organização com equipes distribuídas e que precisa garantir eficiência na comunicação e no acompanhamento das ações.

Outro benefício importante é a segurança aprimorada proporcionada pela nuvem, com criptografia de dados, backups automáticos e controles rigorosos de acesso. Esses recursos asseguram que as informações sensíveis sejam protegidas contra perdas e acessadas apenas por usuários autorizados, garantindo conformidade e minimizando riscos de segurança.

A computação em nuvem também oferece à CAACCH agilidade na adaptação a novas demandas, permitindo que a infraestrutura de TI se ajuste automaticamente conforme o crescimento da organização, sem a necessidade de grandes investimentos iniciais. Além disso, a flexibilidade da nuvem facilita a implementação de novas ferramentas ou alterações no sistema, adaptando-se rapidamente às necessidades da CAACCH, seja para aumentar a capacidade de armazenamento ou ajustar a performance de serviços específicos. Esses benefícios concretos não só otimizam os processos internos da organização, mas também garantem que ela possa escalar suas operações de maneira eficiente, melhorando a gestão e a produtividade da equipe.

3.3.3 VANTAGENS DA CLOUD COMPUTING

“A nuvem modificou significativamente o modo como a computação é utilizada. Sendo formada a partir de outras tecnologias disruptivas, como é o caso da virtualização, a cloud oferece um ambiente com facilidades e benefícios para os negócios, além de financeiramente superior ao da computação tradicional. Tanto a indústria (um ambiente altamente competitivo) como a academia (para a realização de pesquisa e desenvolvimento), buscando esses e outros benefícios, estão continuamente migrando seus ambientes computacionais para a nuvem. “(SILVA, Fernanda R.; SOARES, Juliane A.; SERPA, Matheus da S.; et al. Cloud Computing. Porto Alegre.p 233)

A computação em nuvem oferece à **CAACCH** vantagens como redução de custos e maior flexibilidade. A capacidade de escalabilidade permite ajustar rapidamente recursos

conforme a demanda, sem investimentos em infraestrutura física. Além disso, a centralização dos dados facilita o acesso remoto e seguro, melhorando a colaboração e a tomada de decisões. A nuvem também garante maior segurança com criptografia e backups automáticos. Em resumo, a nuvem otimiza a gestão e amplia a eficiência operacional da instituição.

Redução de custos: A nuvem elimina a necessidade de investimentos pesados em infraestrutura física, como servidores e data centers, permitindo que a CAACCH pague apenas pelos recursos que usa, otimizando os custos operacionais.

Escalabilidade: A nuvem permite aumentar ou diminuir a capacidade de processamento e armazenamento conforme a demanda. Durante períodos de pico, como campanhas de doações, a CAACCH pode ajustar facilmente a infraestrutura, voltando a um nível mais simples e menos exigente após o evento.

Acesso Remoto: A centralização dos dados na nuvem permite que membros da equipe acessem informações e sistemas de qualquer lugar, facilitando o trabalho remoto e a colaboração em tempo real, mesmo com equipes distribuídas.

Segurança Avançada: A computação em nuvem oferece criptografia de dados, backups automáticos e controles de acesso rigorosos, assegurando que as informações sensíveis da CAACCH sejam protegidas e acessíveis apenas por usuários autorizados.

Alta Disponibilidade: A nuvem garante alta disponibilidade e redundância de dados, minimizando o risco de falhas ou interrupções nos serviços essenciais para a CAACCH, mantendo os sistemas sempre operacionais.

Agilidade e Flexibilidade: A nuvem permite à CAACCH implementar e ajustar rapidamente soluções conforme as necessidades do momento, seja para novos programas, eventos ou mudanças nas demandas, sem longos prazos de espera ou investimentos em novos recursos.

Facilidade de Backup e Recuperação: A nuvem proporciona backups automáticos e recuperação de dados rápida, o que garante que a CAACCH possa se recuperar rapidamente de qualquer falha, sem risco de perder informações críticas.

Colaboração e Produtividade: Ferramentas baseadas em nuvem possibilitam que as equipes da CAACCH colaborem de forma eficaz e compartilhem dados em tempo real, aumentando a produtividade e a agilidade na tomada de decisões, além de otimizar o atendimento às famílias e programas sociais.

3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING

Implantação da EC2 (Elastic Compute Cloud).

Configuração da Instância EC2: Acesse a seção EC2 no AWS Management Console. Crie uma nova instância, escolhendo o tipo adequado de máquina virtual (ex. t3.micro para ambientes de teste ou m5.large para produção). Selecione o sistema operacional desejado (ex. Ubuntu, Amazon Linux, etc.). Defina o armazenamento (EBS) e a quantidade de disco necessária.

Configuração de Segurança: Crie um grupo de segurança (Security Group) que defina as regras de acesso (por exemplo, permitindo acesso via SSH ou HTTP). Gere um par de chaves (key pair) para acesso seguro à instância.

Lançamento e Acesso à Instância EC2: Revise as configurações e lance a instância. Acesse a instância via SSH ou RDP, conforme o sistema operacional escolhido.

Implantação do RDS (Relational Database Service)

Criação do Banco de Dados RDS: Acesse a seção RDS no AWS Management Console. Selecione o tipo de banco de dados (ex. MySQL, PostgreSQL, SQL Server). Configure as opções de instância (tamanho da instância, armazenamento e backups automáticos). Defina a segurança e as regras de acesso (crie um grupo de segurança para o banco de dados e permita acesso apenas de instâncias EC2 ou IPs específicos).

Configuração de Acesso: Defina as credenciais de acesso (usuário e senha). Configure a VPC para garantir que o banco de dados esteja na rede privada da empresa.

Backup e Recuperação: Ative backups automáticos e configure as janelas de manutenção. Teste o processo de restauração de backups para garantir a disponibilidade de dados.

Implantação do S3 (Simple Storage Service)

Criação de Bucket no S3: Acesse a seção S3 no AWS Management Console. Crie um novo bucket, definindo o nome único globalmente e a região de hospedagem. Selecione o tipo de armazenamento adequado (Standard, Intelligent-Tiering, Glacier) com base nas necessidades de acesso e custo.

Configuração de Políticas e Permissões: Defina as permissões do bucket e configure o controle de acesso (IAM policies ou ACLs) para garantir que apenas usuários e serviços autorizados possam acessar os dados. Ative o versionamento no bucket, se necessário, para rastrear as alterações nos arquivos.

Upload de Arquivos: Realize o upload de arquivos para o bucket S3 e organize-os em pastas conforme necessário. Utilize o AWS CLI ou SDKs para automatizar o processo de upload de dados (se necessário).

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

A escolha do provedor pela equipe foi AWS da Amazon pela grande gama e maturação de serviços que são oferecidos para o clientes, além da empresa ser uma potência reconhecida mundialmente, agregando valor e segurança para os dados da instituição.

As necessidades e requisitos da CAACCH seriam o gerenciamento de doações e estoques para o controle funcional de estoque, BI para apresentações detalhadas de relatórios, segurança para os dados e por fim, não menos importante, a escalabilidade e flexibilidade dos serviços AWS, que atualmente está à frente neste quesito do que o seu concorrente. Amazon S3 será a principal ferramenta utilizada para o salvamento dos dados da instituição, enquanto Amazon RDS será a escolha para o banco de dados.

3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING

Para este projeto de um site de supervisão de doações em cloud computing para uma ONG, o modelo mais adequado é PaaS, dada a facilidade de desenvolvimento e gestão que não requer operação de infraestrutura. Load balance é crucial para garantir disponibilidade, escalabilidade e desempenho por distribuir tráfego entre servidores e impedir a sobrecarga. A anatomia da nuvem inclui virtualização, elasticidade e redundância, permitindo escalabilidade automática e alta disponibilidade. Os paradigmas tecnológicos subjacentes a cloud computing incluem virtualização, computação distribuída, armazenamento em nuvem e serviços gerenciados para proporcionar eficiência e flexibilidade operacional à ONG para cumprir sua missão, sem se preocupar com a infraestrutura.

3.3.7 GOOGLE CLOUD ou AWS

Optamos pela AWS (Amazon Web Services) devido à sua excelente gama de serviços e custos acessíveis. Para a infraestrutura de computação, utilizamos as máquinas EC2, especificamente a instância t2.micro, que oferece um ótimo custo-benefício, atendendo perfeitamente às nossas necessidades de processamento com um preço competitivo.

Para o banco de dados, escolhemos um provedor MySQL, utilizando uma máquina de configuração simples, com 1 CPU e 1 GB de RAM, que é suficiente para suportar o tráfego e as operações da aplicação. Essa configuração não só garante desempenho adequado, mas também se mantém dentro de um custo acessível.

Além disso, utilizamos o Amazon S3 para o armazenamento de arquivos, proporcionando uma solução de armazenamento escalável e de baixo custo para a instituição.

Com essa escolha, conseguimos uma infraestrutura robusta, flexível e de baixo custo, permitindo que a aplicação opere de maneira eficiente e econômica.

3.4 ESTRUTURA DE DADOS

“Para o profissional da área da Ciência da Computação, é importante conhecer a forma como os dados são armazenados, organizados e manipulados, pois essa atividade é uma das principais que são desenvolvidas pelos sistemas computacionais. Nesse sentido, a estrutura de dados é um elemento que vem, para auxiliar na representação e na abstração de estruturas mais complexas, executando operações de armazenamento e busca de dados na memória, de maneira mais sofisticada e robusta.” (CURY, Thiago E.; BARRETO, Jeanine dos S.; SARAIVA, Maurício de O.; et al. Estrutura de Dados. Porto Alegre. p.21)

O projeto visa otimizar a gestão de dados como doações, beneficiários e estoque, utilizando TypeScript para garantir eficiência e escalabilidade. O levantamento de requisitos foi elaborado com foco na escolha de estruturas de dados adequadas, listas ordenadas e etc., para atender às operações frequentes do sistema. A integração com a infraestrutura em nuvem da AWS assegurará a alta disponibilidade, escalabilidade e segurança necessárias para

gerenciar grandes volumes de dados, oferecendo uma solução robusta e eficiente para a organização.

3.4.1 LEVANTAMENTO DE REQUISITOS

Podemos entender que o gerenciamento de requisitos “[...] inclui todas as atividades que mantêm a integridade, a acurácia e a atualidade dos acordos sobre os requisitos ao longo do projeto” (WIEGERS; BEATYY, 2013, p. 458, tradução nossa). Pohl e Rupp (2015, p. 111, tradução nossa) definem que:

[...] o gerenciamento de requisitos compreende atribuir intencionalmente atributos aos requisitos, definindo visões sobre os requisitos, priorizando requisitos e rastreando os requisitos, bem como versionando os requisitos, gerenciando mudanças nos requisitos e medindo os requisitos.

O levantamento de requisitos do projeto foi focado na escolha das estruturas de dados adequadas para garantir a eficiência e o desempenho do sistema.

Para garantir uma gestão eficiente dos dados, foram selecionadas diferentes estruturas de dados, cada uma com suas vantagens específicas, que serão utilizadas para armazenar e acessar rapidamente informações como doações (ID único), beneficiários (CPF) e estoque.

Além disso, para organizar dados ordenados, como nomes e datas de doações. Buscas rápidas serão especialmente úteis para consultas que dependem de critérios específicos como data ou nome, permitindo que o sistema busque dados de forma otimizada e ágil.

Filas (Queues) serão utilizadas para controlar a ordem de entrada e saída de itens, como no gerenciamento do estoque ou em processos sequenciais que seguem uma ordem específica, como o atendimento a beneficiários ou a verificação de doações, onde as filas são perfeitas para operações em que os elementos devem ser processados conforme a ordem de chegada, garantindo que o sistema execute tarefas de maneira eficiente e sem sobrecarregar o processamento.

Listas Ordenadas foram escolhidas para armazenar dados que exigem ordenação constante, como doações por data ou quantidade de itens no estoque. Estas listas permitem buscas rápidas, sendo eficientes quando o volume de dados não exige uma constante reordenação e quando há a necessidade de fazer consultas frequentes de dados ordenados.

3.4.2 VALIDAÇÃO DOS REQUISITOS

A integração das estruturas de dados com a infraestrutura de computação em nuvem da AWS oferece as condições ideais para criar um sistema escalável, resiliente e eficiente. Utilizando serviços como S3, RDS e EC2, o sistema será capaz de gerenciar grandes volumes de dados, fornecer alta disponibilidade e garantir a integridade das informações. A elasticidade da nuvem permitirá que o sistema se ajuste automaticamente às variações de demanda, mantendo um desempenho elevado mesmo durante períodos de pico. Essa abordagem garantirá que o software atenda de maneira eficaz às suas necessidades operacionais e estratégicas.

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

Preconceitos se injetam em nossas vidas diárias dando nomes a classes com base em atributos como gênero, raça e ocupação, o que pode negar oportunidades como a noção de que meninas devem brincar com bonecas e meninos com carrinhos de brinquedo. Eles são ainda mais enfatizados pela mídia, impactando assim nossa visão de mundo, como por meio de comerciais em que mulheres são tratadas como objetos. A busca por aprovação também foi levada para as redes sociais, onde resulta na idealização de vidas que levam à insatisfação e ao traçado de paralelos; influenciadores podem postar fotos impecáveis que não retratam a realidade. Viver com uma distinção significa respeitar a singularidade que refuta estereótipos que homogeneizam indivíduos e que servem como formas de preconceito ou discriminação como a xenofobia. Precisamos de crítica e autorreflexão sobre preconceito para buscar inclusão para uma sociedade mais justa.

3.5.2 ESTUDANTES NA PRÁTICA

O Material prático solicitado foi realizado em forma de um banner sendo exposto durante a apresentação final do projeto, abordando o enfrentamento dos estereótipos e o impacto destes no cotidiano.

Imagem 8 : Menu interativo do sistema.



Fonte: Autores (2024).

4. CONCLUSÃO

O desenvolvimento do Sistema de computação em nuvem para a CAACCH dá-se continuidade do semestre anterior, marcando uma grande mudança na gestão de processos internos da organização. O Projeto apresenta uma arquitetura em nuvem escalável, proporcionando mais segurança, eficiência e adaptabilidade, facilitando o crescimento da CAACCH.

Ao incorporar uma plataforma de Business Intelligence (BI) e automatizar processos como o monitoramento de doações e a gestão de estoque, o sistema aprimora a acuracidade das operações, promovendo transparência e rapidez nas decisões. Foi implementado banco de dados na nuvem da amazon(AWS), utilizando estruturas de dados que garantem capacidade de expansão e alta disponibilidade otimiza o funcionamento sem precisar de maiores investimentos pela instituição, principalmente em infraestrutura física.

Se utilizando de ferramentas como o typescript, javascript e a computação em nuvem, por exemplo, foi possível oferecer maior flexibilidade e segurança para a ONG.

Adicionalmente, a plataforma analítica permitirá à organização fazer escolhas fundamentadas em dados concretos, otimizando os recursos disponíveis e conseqüentemente aprimorando as campanhas de arrecadação. Centralizando as informações e o acesso remoto, os membros da CAACCH utilizaram de uma ferramenta mais robusta e eficiente para as suas operações.

As principais dificuldades encontradas na realização do projeto se dão por parte da implementação de novas ferramentas apresentadas ao longo do semestre, e também por parte da conciliação com as necessidades da instituição. Porém, foi realizada uma conclusão satisfatória por parte dos integrantes da equipe, atendendo os requisitos da ONG.

De acordo com os elementos supracitados pode se concluir que o sistema aprimora a organização interna e também aumenta o impacto social e o alcance que a ONG CAACCH pode alcançar.

REFERÊNCIAS

ELMASRI, Ramez; **Sistemas de banco de dados** / Ramez Elmasri e Shamkant B. Navathe; revisor técnico Luis Ricardo de Figueiredo. -- São Paulo : Pearson Addison Wesley, 2005. ISBN 85-88639-17-3. Acesso em 26 ago. 2024

CARDOSO, Giselle C.; CARDOSO, Virgínia M. **Sistemas de Banco de Dados, 1ª Edição**. Rio de Janeiro: Saraiva, 2012. *E-book*. p.[Inserir número da página]. ISBN 9788502162839. Acesso em 26 ago. 2024.

SILVA, Fabricio M.; LEITE, Márcia C D.; OLIVEIRA, Diego B. Paradigmas de programação. Porto Alegre: SAGAH, 2019. *E-book*. p.59. ISBN 9788533500426. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788533500426/>. Acesso em 08 set. 2024.

BARRETO, Jeanine dos S.; JR., Paulo A P.; BARBOZA, Fabrício F M.; et al. **Interface humano-computador**. Porto Alegre: SAGAH, 2018. *E-book*. p.47. ISBN 9788595027374. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788595027374/>. Acesso em 08 set. 2024.

SILVA, Fernanda R.; SOARES, Juliane A.; SERPA, Matheus da S.; et al. **Cloud Computing**. Porto Alegre: SAGAH, 2020. *E-book*. p.258. ISBN 9786556900193. Acesso em 18 set. 2024.

ERL, Thomas; MONROY, Eric B. **Computação em Nuvem: Conceitos, Tecnologia, Segurança e Arquitetura**. 2nd ed. Porto Alegre: Bookman, 2024. *E-book*. p.1. ISBN 9788582606599. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788582606599/>. 2nd ed. Porto Alegre: Bookman, 2024. *E-book*. Acesso em 18 set. 2024.

CURY, Thiago E.; BARRETO, Jeanine dos S.; SARAIVA, Maurício de O.; et al. Estrutura de Dados. Porto Alegre: SAGAH, 2018. *E-book*. p.21. ISBN 9788595024328. Disponível em: <https://integrada.minhabiblioteca.com.br/reader/books/9788595024328/>. Acesso em 10 out. 2024.

DROZDEK, Adam. Estrutura de Dados e Algoritmos em C++ – Tradução da 4ª edição norte-americana. 2nd ed. São Paulo: Cengage Learning Brasil, 2018. E-book. p.18. ISBN 9788522126651. Disponível em:
<https://integrada.minhabiblioteca.com.br/reader/books/9788522126651/>. Acesso em 10 out. 2024.

REINEHR, Sheila. Engenharia de requisitos. Porto Alegre: SAGAH, 2020. E-book. p.285. ISBN9786556900674. Disponível em:
<https://integrada.minhabiblioteca.com.br/reader/books/9786556900674/>. Acesso em: 10 out. 2024.

ANEXOS

ANEXO 1: MODELO FÍSICO no drive Disponível em: [Imagem 1](#)

ANEXO 2: ARMAZENAMENTO em drive das imagens do desenvolvimento do projeto (Back-end e Front-end). Disponível em:

<https://drive.google.com/drive/folders/1g3WVPxnLPjaYfqgLVPrvCzT3UMSjyuP5>

ANEXO 3: BANNER feito no Canva. Disponível em: [BANNE](#)

