



2023

PROJETO INTEGRADO



UNIFEOB

CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO OCTÁVIO
BASTOS

ESCOLA DE NEGÓCIOS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

CIÊNCIA DA COMPUTAÇÃO

PROJETO INTEGRADO

Plataforma de Gerenciamento de Dados em Nuvem

MALPA TECH

SÃO JOÃO DA BOA VISTA, SP

NOVEMBRO 2023

UNIFEOB
CENTRO UNIVERSITÁRIO DA FUNDAÇÃO DE ENSINO OCTÁVIO
BASTOS

ESCOLA DE NEGÓCIOS

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
CIÊNCIA DA COMPUTAÇÃO

PROJETO INTEGRADO

Plataforma de Gerenciamento de Dados em Nuvem

MALPA TECH

MÓDULO COMPUTAÇÃO EM NUVEM

Estrutura de Dados – Prof. Mauro Glória

Linguagem e Técnicas de Programação – Prof. Nivaldo 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:

Bruno Cardoso Silva, RA 22000657

Igor Guilherme dos Reis Melo ,RA 22000126

Júlio César Godoy, RA 22001748

Matheus Broesler, RA 22000961

Matheus Souza Pinto, RA 22000502

Nicole Martins Fernandes Corrêa, RA 22000733

SÃO JOÃO DA BOA VISTA, SP
NOVEMBRO 2023

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	6
3.1.2 MODELO FÍSICO	8
3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO	13
3.2.1 PROTOTIPAÇÃO	13
3.2.2 FRONT-END	16
3.2.3 BACK-END	17
3.2.4 CRUD	18
3.3 COMPUTAÇÃO EM NUVEM	19
3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING	19
3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO	19
3.3.3 VANTAGENS DA CLOUD COMPUTING	21
3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING	22
3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)	23
3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING	23
3.3.7 GOOGLE CLOUD ou AWS	24
3.4 ESTRUTURA DE DADOS	25
3.4.1 LEVANTAMENTO DE REQUISITOS	25
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS	28
3.5.1 ENFRENTANDO ESTEREÓTIPOS	29
3.5.2 ESTUDANTES NA PRÁTICA	29
4. CONCLUSÃO	31
REFERÊNCIAS	32
ANEXOS	33

1. INTRODUÇÃO

O projeto foi desenvolvido com a intenção de criar uma plataforma de gerenciamento de dados em nuvem, sendo um ambiente seguro e eficiente para o armazenamento, acesso e manipulação de informações. Nesse sentido, foi desenvolvido a plataforma Malpa, inspirado no nono e décimo primeiro Objetivos de Desenvolvimento Sustentável (ODS) estabelecidos pela ONU, nos quais visam: construir infra estruturas resilientes, promovendo a industrialização inclusiva, sustentável e inovação e, tornar as cidades e os assentamentos humanos segura, resiliente e sustentável, respectivamente.

Nesse contexto, a Malpa garante que o usuário faça seu cadastro na plataforma, o qual é redirecionado para a página principal. Nela, o indivíduo pode escolher uma empresa e fazer a sua denúncia. Sendo assim, a empresa tem acesso a essas informações, tornando mais fácil a solução de problemas.

Durante a elaboração do projeto, foi buscado integrar os conhecimentos adquiridos nas disciplinas de Estrutura de Dados, Linguagem e Técnicas de Programação, Tópicos Avançados de Banco de Dados e Computação em Nuvem.

Dessa forma, a plataforma foi desenvolvida sobre uma sólida arquitetura de computação em nuvem, reunindo os avanços tecnológicos contemporâneos, para garantir a escalabilidade e disponibilidade dos dados. Além disso, para o acompanhamento, foi incorporada técnicas avançadas de gerenciamento de banco de dados.

Portanto, durante o seu desenvolvimento, foi aplicado de forma criativa, conceitos essenciais nas áreas de estudo, para que pudesse oferecer um futuro mais sustentável e inclusivo com base nas metas oferecidas pelas ODS.

2. DESCRIÇÃO DA EMPRESA

A Małpa é uma empresa fictícia criada para a elaboração do projeto. Ela opera principalmente na área de tecnologia e, a princípio, o seu surgimento foi motivado pela constatação de desafios recorrentes enfrentados pelos habitantes da cidade de São João da Boa Vista. A companhia emergiu como uma solução para conectar de forma eficaz os cidadãos aos órgãos responsáveis por resolver questões que afetam o cotidiano da comunidade, tais como: a manutenção de ruas esburacadas, problemas na coleta de lixo, e outras demandas urbanas relacionadas à cidade.

Em suma, a principal contribuição da Małpa é a criação de uma plataforma de comunicação que aproxime os munícipes dos serviços públicos e das empresas que prestam serviços na região. Com esse propósito em mente, foi desenvolvido uma plataforma de denúncias que permite a qualquer cidadão apresentar reclamações e reportar problemas diretamente às empresas parceiras cadastradas na plataforma.

3. PROJETO INTEGRADO

3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS

Parafraseando o livro Princípios de Banco de Dados, escrito pelos professores Décio Lehmkuhl e Djayson Roberto Eger, para entender o que é um banco de dados, devemos saber a diferença entre dados e informações. Os dados são conteúdos que ainda não foram processados para gerar um significado. Assim que esses são coletados, eles são armazenados, analisados e formatados, gerando-se uma informação. Desse modo, as informações são como uma tradução desses dados, para que o entendimento seja facilitado e interpretado.

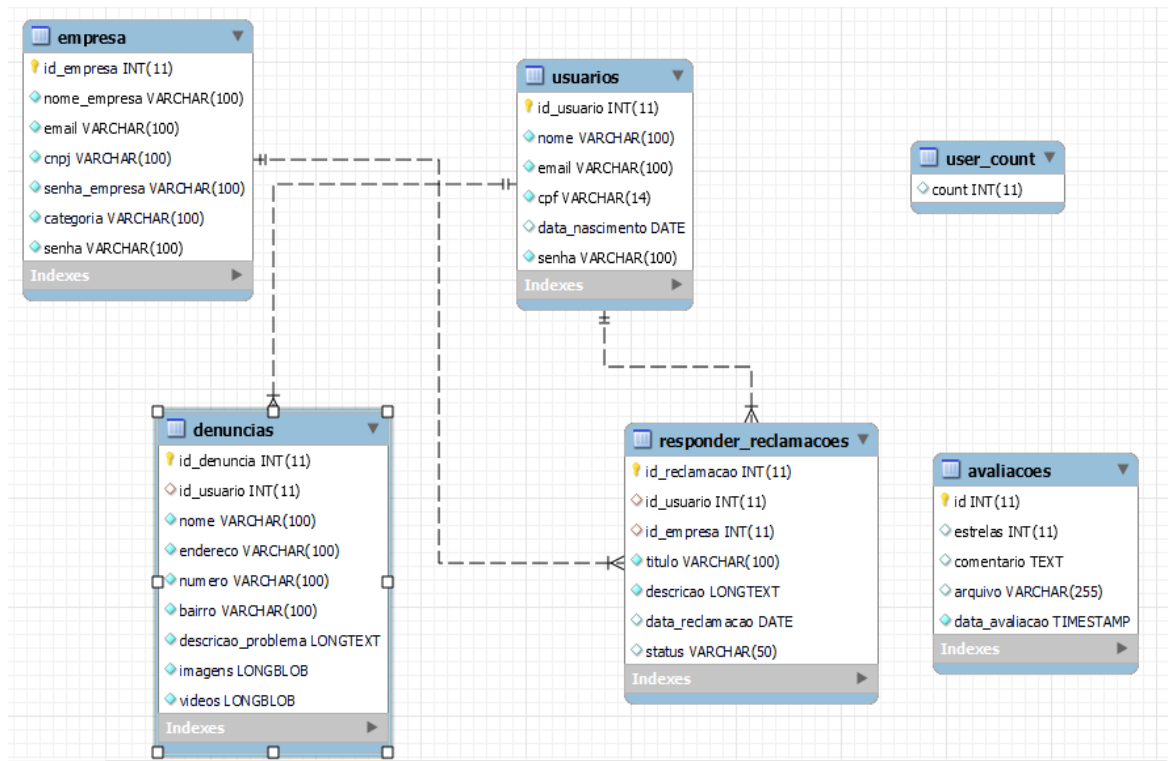
Todos esses dados e informações são armazenados em um software responsável por gerenciar um banco de dados, no qual, existem funções pré-definidas para facilitar a inclusão, remoção, consulta e atualização dessas informações. Sendo assim, um banco de dados serve para o armazenamento de forma segura e responsável desses dados, a fim de serem consumidos em seguida.

3.1.1 MODELO LÓGICO

Seguindo o mesmo conceito do livro citado, o modelo lógico representa as estruturas responsáveis por armazenar os dados dentro de um banco de dados, é nele que são definidas as tabelas, relacionamento, regras, as colunas, entidades, atributos e, também, é definida a consideração de qual abordagem será utilizada (Relacional, Hierárquico ou de Rede). Dessa forma, é parecido com o modelo conceitual, porém possui mais detalhes e, não apenas conceitos pré definidos.

Ademais, para construção do modelo lógico da empresa Malpa, foi utilizado o software MySQL Workbench. Antes de tudo, foram criadas colunas nomeadas como: avaliações, denúncias, empresa, reclamações e usuário; em que cada uma possui seus atributos específicos, a fim de que, esses dados sejam capturados quando o usuário realizar uma interação com o site. Sendo assim, a imagem em anexo representa o modelo lógico do banco de dados do software.

Imagem 1: Tabela no banco de dados "BD_Malpa"



Fonte: autoria própria

As informações a seguir estão todas relacionadas com a imagem acima. Ela será usada para ajudar a entender melhor o que será discutido adiante. Vai ser apresentada as características principais e os propósitos de cada tabela no banco de dados chamado "BD_Malpa".

Tabela "usuarios": A tabela "usuarios" é projetada para armazenar informações sobre usuários. Cada registro possui um identificador único (id_usuario), que é incrementado automaticamente, garantindo unicidade. Além disso, são armazenados dados como nome, email, CPF, data de nascimento, e senha dos usuários. A coluna "email" é configurada como única para garantir que não haja duplicatas.

Tabela "empresa": A tabela "empresa" é destinada ao armazenamento de informações relacionadas a empresas. Cada empresa tem um identificador único (id_empresa), um nome, email, CNPJ, senha e categoria associada. Similar à tabela de usuários, a coluna "email" é configurada como única. Essa tabela pode ser usada para registrar dados essenciais sobre empresas cadastradas no sistema.

Tabela "denúncias": A tabela "denúncias" é projetada para registrar denúncias feitas pelos usuários. Cada denúncia é identificada por um único número (id_denuncia). A tabela inclui informações sobre o usuário que fez a denúncia, como o ID do usuário, nome, endereço, número,

bairro, descrição do problema, imagens e vídeos relacionados à denúncia. A coluna "id_usuario" é uma chave estrangeira que faz referência à tabela de usuários.

Tabela "avaliacao": A tabela "avaliação" é utilizada para armazenar avaliações relacionadas a alguma entidade, embora o propósito específico não esteja claramente indicado. Cada avaliação possui um identificador único (id_avaliacao), uma quantidade de estrelas, além de registros para data de criação e modificação.

Tabela "user_count": A tabela "user_count" tem um único propósito de manter um contador de usuários, com um valor padrão inicial de zero. Essa tabela pode ser utilizada para armazenar e controlar informações sobre a quantidade de usuários no sistema.

Tabela "responder_reclamacoes": A tabela "responder_reclamacoes" é destinada ao registro de respostas a reclamações. Cada registro possui um identificador único (id_reclamacao) e inclui informações sobre o usuário que fez a reclamação, a empresa envolvida, título, descrição, data da reclamação e status da reclamação. As colunas "id_usuario" e "id_empresa" são chaves estrangeiras que fazem referência às tabelas de usuários e empresas, respectivamente.

3.1.2 MODELO FÍSICO

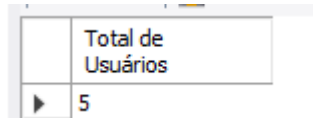
O modelo físico é construído a partir do modelo lógico, ele tem o intuito de ser aplicado dentro do software, responsável por armazenar o banco de dados. Nele entram questões relacionadas, por exemplo: ao tipo de tamanho do campo e restrições..

É o modelo físico que é responsável pela etapa final do projeto do banco de dados, é nele que é utilizada a linguagem e funções pré-definidas para atualizações, remoções e incrementações de dados. Ou seja, é nele que é permitido a utilização de scripts e serão responsáveis pela implementação do projeto físico em diversos softwares.

Sendo assim, foram utilizadas procedures para facilitar a manutenção, ganhar de tempo ao realizar os códigos, foi usado para contar os usuários, empresas e denúncias, além de triggers, que são gatilhos e estão associados a uma tabela e, foi empregado para atualizar datas de modificações, impedir que a senha e dados importantes sejam nulas, verificar a integridade dos mesmos e, por último, foram utilizado comandos básicos de select e update. Para que o banco esteja sempre com dados atualizados evitando erros futuros.

Imagem 2: Procedure de contar usuários

```
CREATE PROCEDURE ContarUsuarios()  
{ BEGIN  
    DECLARE total_usuarios INT;  
    SELECT COUNT(*) INTO total_usuarios  
    FROM usuarios;  
    SELECT total_usuarios AS 'Total de Usuários';  
- END;
```



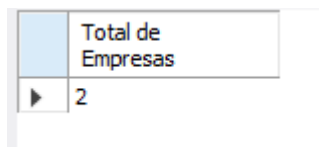
Total de Usuários
5

Fonte: autoria própria

A primeira stored procedure, chamada "ContarUsuarios", tem como objetivo calcular o número total de registros na tabela "usuarios". Ela inicia declarando uma variável local chamada "total_usuarios" do tipo INT. Em seguida, realiza uma contagem dos registros na tabela "usuarios" e armazena esse valor na variável previamente declarada. A stored procedure retorna o valor de "total_usuarios" como um conjunto de resultados identificado pelo nome "Total de Usuários". Em resumo, essa procedure fornece uma maneira de obter o total de usuários registrados na tabela "usuarios".

Imagem 3: Procedure de contar empresas

```
CREATE PROCEDURE ContarEmpresas()  
BEGIN  
    DECLARE total_empresas INT;  
    SELECT COUNT(*) INTO total_empresas  
    FROM empresa;  
    SELECT total_empresas AS 'Total de Empresas';  
END;
```



Total de Empresas
2

Fonte: autoria própria

A segunda stored procedure, denominada "ContarEmpresas", possui um propósito semelhante à anterior, mas voltado para a tabela "empresa". Ela declara uma variável local chamada "total_empresas", conta os registros na tabela "empresa" e armazena esse valor na variável. O

resultado é então retornado como um conjunto de resultados identificado pelo nome "Total de Empresas". Essa procedure fornece uma contagem do total de empresas registradas na tabela "empresa".

Imagem 4: Procedure de contar denúncias por usuários

```
CREATE PROCEDURE ContarDenunciasPorUsuario(idUsuario INT)
BEGIN
    DECLARE total_denuncias INT;
    SELECT COUNT(*) INTO total_denuncias
    FROM denuncias
    WHERE id_usuario = idUsuario;
    SELECT total_denuncias AS 'Total de Denúncias do Usuário';
END;
```

Fonte: autoria própria

A terceira stored procedure, intitulada "ContarDenunciasPorUsuario", é projetada para calcular o número total de denúncias associadas a um usuário específico. Ela recebe um parâmetro chamado "idUsuario" para identificar o usuário-alvo das contagens. Dentro da procedure, é declarada uma variável local chamada "total_denuncias", que conta os registros na tabela "denuncias" onde o "id_usuario" corresponde ao parâmetro fornecido. O valor resultante é retornado como um conjunto de resultados identificado pelo nome "Total de Denúncias do Usuário". Em resumo, essa stored procedure oferece uma maneira de obter o total de denúncias associadas a um usuário específico com base no "idUsuario" fornecido.

Imagem 5: Procedure de contar denúncias

```
DELIMITER $$
CREATE PROCEDURE ContarDenuncias()
BEGIN
    DECLARE total_denuncias INT;
    SELECT COUNT(*) INTO total_denuncias
    FROM denuncias;
    SELECT total_denuncias AS 'Total de Denúncias';
END;
$$
```

Fonte: autoria própria

Essa stored procedure, chamada "ContarDenuncias", foi projetada para calcular o número total de registros na tabela "denuncias". Ela começa declarando uma variável local chamada "total_denuncias" do tipo INT. Em seguida, utiliza uma instrução SQL para contar os registros na tabela "denuncias" e armazena esse valor na variável previamente mencionada. A stored procedure retorna o valor de "total_denuncias" como um conjunto de resultados identificado pelo nome "Total de Denúncias". Em resumo, esse procedimento armazenado oferece uma maneira eficiente de obter e apresentar o total de denúncias registradas na tabela "denuncias".

Imagem 6: Trigger da tabela empresa de atualizar data de modificação

```
CREATE DEFINER=`root`@`localhost` TRIGGER AtualizarDataModificacão
BEFORE UPDATE ON empresa
FOR EACH ROW
SET NEW.modified = NOW()
```

Fonte: autoria própria

Esta trigger é acionada antes de qualquer atualização na tabela "empresa" e tem a finalidade de atualizar a coluna "modified" para refletir a data e hora da modificação sempre que um registro é atualizado. Em resumo, ela garante que a coluna "modified" seja automaticamente atualizada para o valor atual (obtido através da função NOW()) sempre que houver uma atualização em um registro na tabela "empresa". Isso pode ser útil para rastrear quando um registro foi modificado pela última vez.

Imagem 7: Trigger de impedir que a senha seja nula

```
DELIMITER $$
CREATE TRIGGER ImpedirSenhaNula
BEFORE UPDATE ON empresa
FOR EACH ROW
) BEGIN
)   IF NEW.senha IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'A senha não pode ser nula.';
~   END IF;
~ END;
$$
DELIMITER ;
```

Fonte: autoria própria

Esta trigger impede a atualização de registros na tabela "empresa" se o novo valor da coluna "senha" for nulo. Se alguém tentar realizar uma atualização que resulte em um valor nulo para a senha, a trigger interrompe a operação e emite uma mensagem de erro indicando que a senha não pode ser nula. Em resumo, ela assegura que a informação da senha seja sempre fornecida e não seja deixada em branco ou nula.

Imagem 8: Trigger de impedir que o cnpj seja nulo

```
DELIMITER $$
CREATE TRIGGER ImpedirCnpjNulo
BEFORE UPDATE ON empresa
FOR EACH ROW
BEGIN
    IF NEW.cnpj IS NULL THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'O CNPJ não pode ser nulo.';
    END IF;
END;
$$
DELIMITER ;
```

Fonte: autoria própria

Esta trigger atua como uma medida preventiva para garantir que não seja possível atualizar registros na tabela "empresa" com um valor nulo na coluna "cnpj". Se alguém tentar realizar uma

atualização que resulte em um valor nulo para o CNPJ, a trigger interrompe a operação e emite uma mensagem de erro ('O CNPJ não pode ser nulo.') indicando a violação da restrição. Em resumo, ela garante a integridade dos dados, assegurando que a informação do CNPJ seja sempre fornecida e não seja deixada em branco ou nula.

3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO

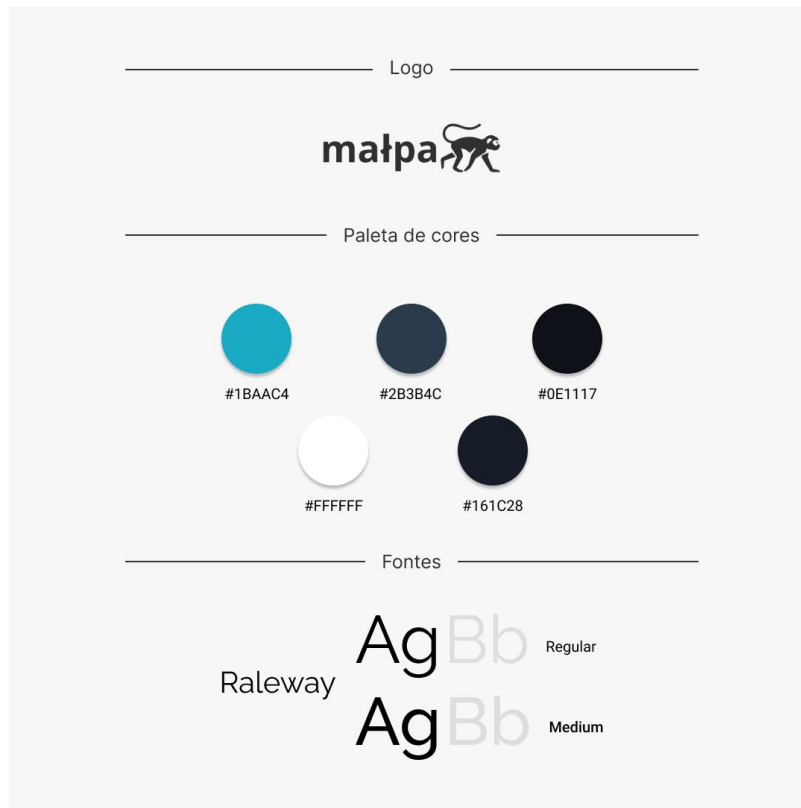
A linguagem técnica de programação refere-se às regras e convenções que os programadores usam para se comunicar com o computador. No livro Estruturas de Dados e Algoritmos em Java de Robert Lafore exemplifica como funciona a linguagem técnica de programação. Ele aborda a compreensão das estruturas de dados e algoritmos, que são essenciais para resolução de problemas de programação. Nesse modo, ajuda a entender como são usadas as linguagens de programação para criar soluções eficazes e eficientes.

Ademais, nesse conteúdo foi abordado linguagens específicas como: Node.JS, JavaScript e React. Elas foram fundamentais para o desenvolvimento da lógica do software, pois permitiram descrever tarefas específicas para serem executadas. Sendo assim, com elas, pôde-se dar abertura para outras linguagens e técnicas durante a criação da plataforma.

3.2.1 PROTOTIPAÇÃO

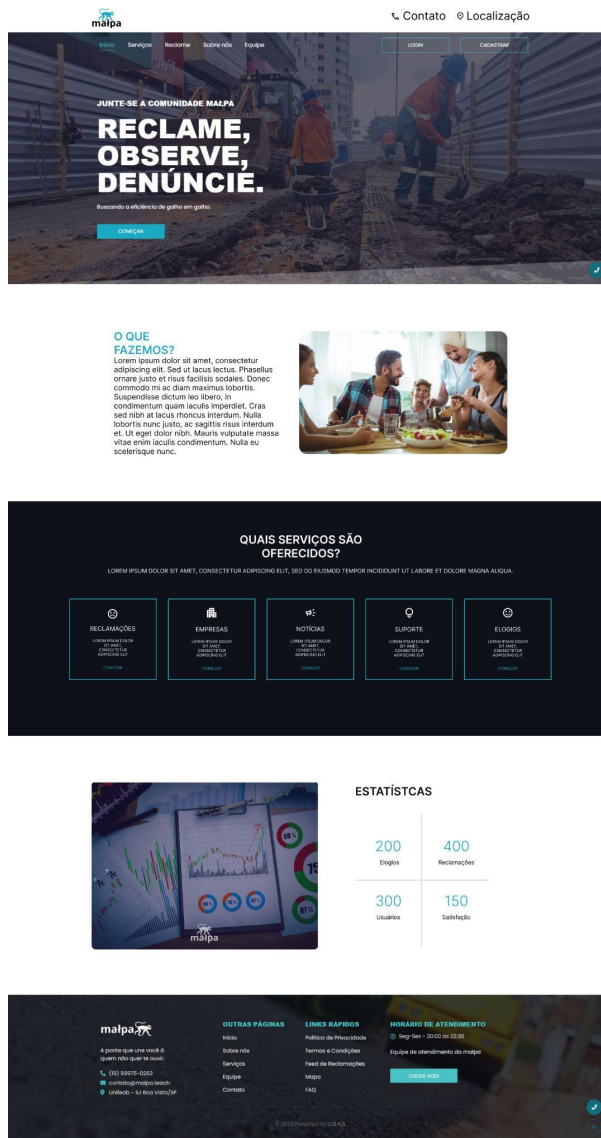
Para facilitar o esclarecimento das ideias, foi desenvolvido um protótipo de média fidelidade da página inicial e de perfil das empresas. Junto a isso, foi criado um guia de estilo que possui a logo, fonte e paleta de cores. Com a finalidade de produzir uma representação interativa e visual do design do site antes da sua implementação. Assim, torna-se mais fácil a aprovação visual e economiza tempo ao ser programado. As imagens 3 e 4 a seguir representam o protótipo elaborado pela equipe.

Imagem 9: Guia de estilo da empresa Malpa



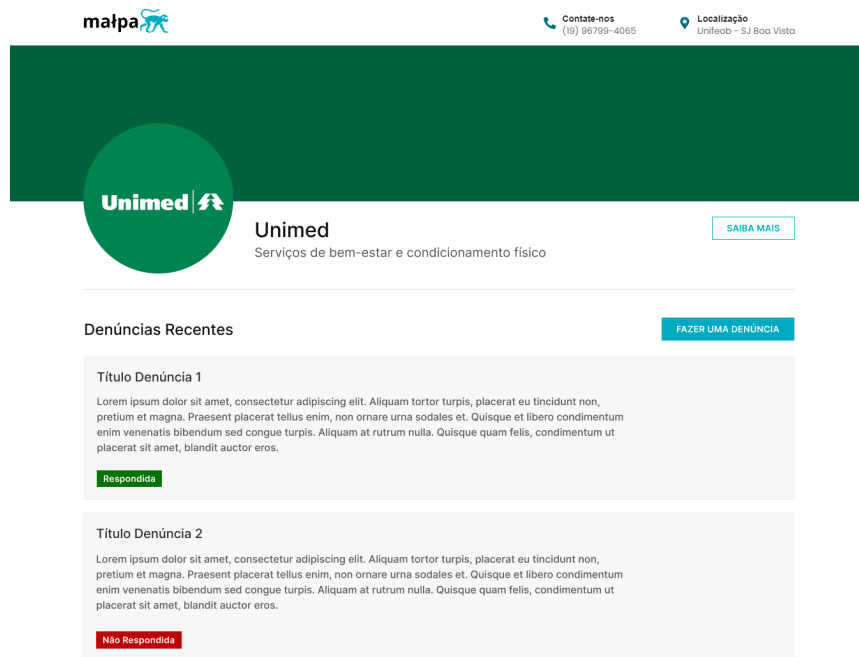
Fonte: autoria própria

Imagem 10: Página inicial



Fonte: autoria própria

Imagem 11: Página de perfil



Fonte: autoria própria

3.2.2 FRONT-END

A criação do presente website fundamentou-se na integração sinérgica de HTML, CSS, JavaScript e React. O HTML desempenhou um papel crucial na conformação estrutural da página, enquanto o CSS assumiu a responsabilidade pela estilização, conferindo uma apresentação visualmente atraente e congruente. Por sua vez, o JavaScript foi empregado para incorporar interatividade, permitindo animações e a emissão de alertas ao usuário.

No intuito de otimizar a estética e o layout do site, optou-se por empregar o framework Bootstrap. Este framework proporcionou uma ampla gama de componentes, classes e estilos predefinidos, simplificando a criação de um design responsivo e esteticamente agradável.

Além disso, a aplicação do JavaScript possibilitou a implementação de animações e interações dinâmicas, conferindo ao site uma atração adicional e proporcionando uma experiência interativa ao usuário. Por meio do JavaScript, foram desenvolvidos recursos como transições animadas, feedbacks visuais e alertas personalizados.

A conjunção destas tecnologias, linguagens e frameworks contribuiu substancialmente para a concepção do website preconizado, enriquecendo tanto a sua apresentação quanto a sua funcionalidade. Através desta abordagem integrada, almejou-se conferir aos usuários uma experiência de navegação fluida, agradável e interativa.

3.2.3 BACK-END

O desenvolvimento do Back-End para o presente projeto utiliza o servidor Node.js em conjunto com as tecnologias Express, EJS e Sequelize, e PHP para outras funções, todas fundamentadas na linguagem de programação JavaScript no âmbito do servidor. O Node.js representa uma plataforma de código aberto que possibilita a execução de código JavaScript fora do navegador, viabilizando, dessa forma, a criação de servidores web robustos e escaláveis. O Express é um framework web desenvolvido para o Node.js, destinado a simplificar o processo de desenvolvimento de aplicativos web, provendo recursos para gerenciar aspectos como rotas, autenticação, gestão de sessões, dentre outros.

Adicionalmente, o EJS (Embedded JavaScript) consiste em um mecanismo de template que permite a inserção direta de código JavaScript em arquivos HTML, simplificando a geração dinâmica de conteúdo HTML com base em dados fornecidos pelo servidor. Quanto ao Sequelize, trata-se de um ORM (Object-Relational Mapping) que simplifica a interação com bancos de dados relacionais no contexto do Node.js, oferecendo uma camada de abstração para realizar operações como consultas, inserções e atualizações no banco de dados.

No contexto deste projeto, o Node.js é empregado para implementar o servidor que irá receber as requisições originadas do Front-End. Essas solicitações são tratadas pelo Express, que utiliza rotas para determinar o processamento adequado de cada uma delas. Deste modo, quando um usuário preenche o formulário no Front-End com informações como nome, endereço, CPF, detalhes do problema e envia uma foto/vídeo, tais dados são encaminhados ao servidor Node.js.

Com os dados recebidos em mãos, o servidor Node.js faz uso do Sequelize para estabelecer a conexão com o banco de dados e efetuar a persistência das informações. Posteriormente, é possível empregar soluções de armazenamento em nuvem, como serviços de hospedagem de arquivos, para o envio da foto ou vídeo capturados pelo formulário.

A conexão entre o Node.js e a nuvem pode ser estabelecida através de serviços específicos, tais como Amazon S3 ou Google Cloud Storage, os quais disponibilizam APIs para a transferência e armazenamento seguro dos arquivos no ambiente de nuvem.

Esta abordagem assegura o suporte apropriado para as informações originadas do Front-End, permitindo o armazenamento seguro e o subsequente acesso a esses dados, tais como nome, endereço, CPF, detalhes do problema e foto/vídeo, por meio de consultas ao banco de dados ou aos serviços de armazenamento em nuvem.

A escolha por PHP foi motivada pela sua facilidade de manipulação e pela familiaridade da equipe com essa linguagem, o que permitiu uma implementação mais ágil e eficiente dadas as restrições de tempo do projeto.

Em nosso projeto, o Back-End desempenha um papel crucial na gestão das denúncias feitas pela população em relação a problemas nas ruas, como vazamentos de esgoto. O processo ocorre da seguinte forma:

Recepção da Denúncia: Quando um usuário preenche o formulário no Front-End com informações como nome, endereço, CPF, detalhes do problema e envia uma foto/vídeo, esses dados são encaminhados ao servidor PHP.

Processamento pelo PHP: O servidor PHP, por meio do Node.js, recebe e processa essas informações. Utilizando o Sequelize para estabelecer a conexão com o banco de dados MySQL, o PHP realiza a persistência das informações relacionadas à denúncia.

Notificação à Empresa Responsável: Após a persistência dos dados, o sistema envia automaticamente uma notificação à empresa responsável pela manutenção naquela região. Essa notificação contém os detalhes da denúncia, permitindo à empresa tomar as medidas necessárias para solucionar o problema.

Integração com Serviços de Armazenamento em Nuvem: Caso o formulário contenha foto/vídeo, o PHP também gerencia a integração com serviços de armazenamento em nuvem, como Amazon S3 ou Google Cloud Storage. Isso garante o armazenamento seguro e eficiente desses arquivos, permitindo à empresa acessá-los conforme necessário.

A utilização do PHP proporcionou uma implementação mais acessível para a equipe, garantindo uma resposta rápida às demandas da população e permitindo uma eficiente comunicação entre o Front-End e o Banco de Dados. Esta abordagem contribuiu para a agilidade na resolução de problemas nas ruas da cidade, alinhando-se ao propósito fundamental do projeto.

3.2.4 CRUD

Nesse contexto, adota-se a linguagem MySQL em conjunto com a IDE Workbench para a implementação e administração do Banco de Dados. Esta escolha é respaldada pela extensa aceitação e confiabilidade que essa linguagem possui no âmbito do desenvolvimento web, além das funcionalidades e recursos proporcionados pela IDE Workbench no que concerne à modelagem, consulta e gestão do Banco de Dados.

Uma das motivações preponderantes para a utilização de uma solução de armazenamento em nuvem reside na garantia de maior segurança para o sistema. A modalidade de armazenamento em nuvem, de maneira geral, oferece redundância de dados e capacidade de backup, o que se revela

fundamental. Dessa maneira, mesmo em situações de falhas ou perda de dados, é viável recuperar as informações armazenadas de forma expedita e confiável.

No âmbito do sistema concebido, faz-se necessário armazenar uma variedade de informações, com destaque para imagens e vídeos submetidos pelos usuários, aliados aos pormenores das denúncias apresentadas. Estas informações assumem importância primordial para que a empresa possa analisar as denúncias e responder de maneira eficaz aos usuários. Portanto, a capacidade de armazenamento proporcionada pela solução em nuvem representa um elemento crítico para o adequado funcionamento do sistema.

Nesse sentido, ao integrar o sistema desenvolvido ao Banco de Dados em nuvem, torna-se possível registrar e persistir de forma segura as informações pertinentes, tais como os detalhes do usuário, os vídeos/fotos das denúncias e dados suplementares. Dessa forma, a empresa terá acesso a todas essas informações armazenadas, permitindo analisar as denúncias e fornecer respostas aos usuários de maneira ágil e eficiente.

3.3 COMPUTAÇÃO EM NUVEM

A computação em nuvem é uma forma de você não depender somente de um servidor ou de um computador. Dessa forma, você pode usar vários servidores para armazenar seus dados e realizar tarefas específicas, tudo através da internet. Sendo assim, o projeto foi desenvolvido com a intenção de criar uma plataforma de gerenciamento de dados em nuvem, no qual, a computação em nuvem, permitiu que os recursos fossem dimensionados conforme a demanda, sem precisar de um investimento em hardware adicional, em consequente, a empresa pôde economizar na criação de um servidor físico.

3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING

Para elaborar a empresa, foi optado a escolha de priorizar a computação em nuvem. Primeiramente foram listados os principais objetivos do seu uso. Dentre eles estão a otimização de processos, eliminação de gastos com hardware e manutenção, flexibilidade para dimensionar os recursos e a segurança para manter os dados protegidos.

Portanto, pôde-se criar um ambiente seguro para o armazenamento de dados da empresa, além de redução de gastos desnecessários durante o estudo do projeto.

3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO

A implementação da computação em nuvem representou uma evolução significativa alinhada aos objetivos do projeto. Ao adotar essa abordagem, a empresa visou aprimorar a flexibilidade e a agilidade no desenvolvimento de soluções, otimizando simultaneamente seus custos operacionais.

A nuvem proporciona casos de uso concretos e relevantes, como a capacidade de escalar recursos conforme a demanda do projeto, facilitando a adaptação rápida a picos de atividade. Além disso, a utilização de serviços gerenciados na nuvem simplificou as operações diárias, permitindo que a equipe concentrasse seus esforços na inovação, em vez de gerenciar a infraestrutura.

Os benefícios específicos da computação em nuvem incluíram uma maior flexibilidade na alocação de recursos, resultando em eficiência operacional e custos reduzidos. A agilidade proporcionada pela nuvem permitiu à empresa responder rapidamente às mudanças no ambiente de negócios, acelerando o ciclo de desenvolvimento de soluções e promovendo a inovação contínua.

Dessa forma, a adoção da computação em nuvem emergiu não apenas como uma necessidade tecnológica, mas como uma estratégia essencial que impulsionou a empresa em direção a um ambiente mais dinâmico, eficiente e economicamente vantajoso.

3.3.3 VANTAGENS DA CLOUD COMPUTING

Para a construção do projeto inovador, foi adotado a infraestrutura em nuvem da AWS, utilizando-a como uma solução de Infraestrutura como Serviço (IaaS). Essa escolha estratégica não apenas possibilitou a agilidade no desenvolvimento, mas também trouxe benefícios tangíveis para a população da cidade que servimos.

Ao optarmos pelo modelo de IaaS, evitamos a aquisição de servidores físicos, o que não apenas reduziu significativamente os custos de produção, mas também proporcionou uma flexibilidade excepcional em escalar nossos recursos conforme a demanda do projeto. Com a AWS, conseguimos direcionar mais recursos para a inovação, ao invés de investir em infraestrutura física.

Além disso, a aplicação do modelo de cloud computing abrange aspectos cruciais como Software como Serviço (SaaS), Plataforma como Serviço (PaaS) e IaaS, garantindo que possamos adaptar nossa abordagem conforme as necessidades específicas do projeto. Destaca-se a importância do balanceamento de carga na nuvem para assegurar um desempenho consistente, disponibilidade contínua e eficiência operacional.

A arquitetura de cloud computing que foi implementada engloba diversos componentes e elementos fundamentais. Desde servidores virtualizados até serviços gerenciados, cada parte desempenha um papel crucial no suporte às operações do projeto. A interconexão desses elementos é cuidadosamente projetada para atender aos requisitos específicos da empresa, garantindo uma infraestrutura ágil e escalável.

Em resumo, o uso estratégico do AWS como provedor de serviços de nuvem permitiu que Malpa, alcançasse eficiência operacional, inovação ágil e, o mais importante, oferecesse benefícios tangíveis para a população da cidade. A implementação cuidadosa dos modelos de cloud computing e a arquitetura robusta contribuem para o sucesso contínuo do nosso projeto, alinhando-se perfeitamente com as demandas dinâmicas do mundo digital.

3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING

Antes de iniciar qualquer implementação, foi conduzido um extenso processo de planejamento estratégico. Nessa fase, foram analisados minuciosamente os requisitos do projeto, identificando as necessidades específicas e delineando as metas a serem alcançadas. Esse processo permitiu compreender a complexidade do projeto e determinar como a infraestrutura em nuvem poderia ser melhor aplicada.

Após uma análise detalhada dos principais provedores de serviços em nuvem, optou-se pela AWS devido às suas credenciais sólidas em confiabilidade e à sua presença global. A variedade de serviços oferecidos pela AWS também desempenhou um papel crucial na decisão, permitindo selecionar as soluções mais adequadas para os requisitos específicos do projeto.

Com a escolha da AWS, implementou-se modelos de serviço em nuvem, especificamente adotando o Infraestrutura como Serviço (IaaS). Isso significava que não era necessário investir em servidores físicos, uma decisão estratégica que não apenas reduziu custos, mas também proporcionou a flexibilidade necessária para escalar recursos de acordo com a demanda do projeto.

A aplicação prática de Software como Serviço (SaaS) e Plataforma como Serviço (PaaS) foi uma parte fundamental do processo. Estes modelos permitiram focar no desenvolvimento do software em si, enquanto a AWS gerenciava aspectos como a infraestrutura de servidor e a escalabilidade. Isso não apenas acelerou o desenvolvimento, mas também permitiu uma abordagem mais centrada no cliente.

Destacamos a importância do balanceamento de carga na nuvem para garantir desempenho consistente e disponibilidade contínua. Foi implementada uma arquitetura cuidadosamente projetada, integrando componentes como servidores virtualizados, serviços gerenciados e outros elementos críticos para atender às demandas específicas do projeto.

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

A escolha da AWS para o projeto foi estratégica e baseou-se, em grande parte, na familiaridade prévia que a equipe possuía com a plataforma, graças ao curso fornecido pela UNIFEOB sobre o cloud da AWS. A equipe já havia adquirido conhecimentos sólidos durante o curso, o que facilitou a decisão de implementar a infraestrutura em nuvem da AWS. Essa familiaridade prévia não apenas reduziu a curva de aprendizado, mas também permitiu que a equipe aproveitasse eficientemente os recursos da AWS desde o início do desenvolvimento do projeto.

Além disso, a AWS oferece um suporte robusto e uma variedade de recursos educacionais. A combinação do conhecimento adquirido durante o curso da UNIFEOB e o acesso contínuo à documentação e suporte técnico da AWS proporcionou à equipe um ambiente propício para lidar com desafios específicos do projeto e aprimorar suas habilidades em nuvem.

A escolha da AWS também foi influenciada pela escalabilidade da plataforma e pela diversidade de serviços oferecidos. Essa flexibilidade permitiu que a equipe selecionasse e implementasse soluções específicas para atender às demandas do projeto, otimizando o desempenho, a segurança e a eficiência operacional.

Além disso, a reputação consolidada da AWS como líder no mercado de serviços em nuvem e sua comprovada confiabilidade foram fatores determinantes na decisão. A confiança na estabilidade e na segurança da AWS contribuiu para a tranquilidade da equipe durante o desenvolvimento do projeto, reforçando a escolha estratégica da AWS como a infraestrutura em nuvem ideal para atender às necessidades da empresa.

3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING

No desenvolvimento do projeto em cloud computing, adotou-se um modelo híbrido de aplicação, combinando elementos de Software como Serviço (SaaS) e Infraestrutura como Serviço (IaaS). A escolha de SaaS permitiu concentrar-se no desenvolvimento da aplicação em si, enquanto o IaaS ofereceu a flexibilidade necessária para gerenciar e escalar a infraestrutura conforme necessário. Essa abordagem híbrida foi crucial para otimizar custos e garantir eficiência operacional.

O balanceamento de carga desempenhou um papel vital no ambiente em nuvem, assegurando que a distribuição de trabalho entre os servidores fosse uniforme. Isso não apenas

otimizou o desempenho da aplicação, mas também contribuiu para a disponibilidade contínua. Ao direcionar o tráfego de forma equitativa, evitou-se sobrecarregar servidores individuais, garantindo eficiência operacional e uma experiência do usuário consistente, mesmo em momentos de alta demanda.

Além disso, a anatomia de cloud computing no projeto incluiu elementos essenciais, como servidor virtualizado, armazenamento em nuvem e serviços gerenciados. O servidor virtualizado fornece a base flexível para a execução de aplicativos, enquanto o armazenamento em nuvem oferece uma solução escalável para gerenciar dados. Os serviços gerenciados, por sua vez, simplificaram tarefas operacionais, permitindo que a equipe se concentrasse no desenvolvimento e na inovação.

Ademais, os paradigmas tecnológicos subjacentes à infraestrutura em nuvem estavam alinhados com os princípios fundamentais de cloud computing. A elasticidade, permitindo a escalabilidade sob demanda, foi um paradigma-chave que orientou a alocação eficiente de recursos. A virtualização desempenhou um papel crucial na criação de ambientes isolados e flexíveis, enquanto a automação impulsionou a eficiência operacional, permitindo implementações rápidas e consistentes. A orquestração de serviços foi outro paradigma essencial, coordenando a interação entre diferentes componentes para garantir um funcionamento harmonioso da aplicação em nuvem. Esses paradigmas, trabalhando em conjunto, formaram a base sólida para a eficácia e agilidade do projeto em cloud computing.

3.3.7 GOOGLE CLOUD ou AWS

Para a implantação bem-sucedida do nosso projeto no Google Cloud, foi proposta uma abordagem estruturada que aproveita as funcionalidades específicas desta plataforma. Sendo assim, optamos por uma combinação de serviços, incluindo o uso de Infraestrutura como Serviço (IaaS) para a flexibilidade na gestão de recursos e o Plataforma como Serviço (PaaS) para simplificar o desenvolvimento da aplicação. O Google Cloud oferece uma variedade de serviços nesses modelos, proporcionando a agilidade necessária ao nosso projeto.

Aproveitamos os serviços gerenciados do Google Cloud, como Cloud Storage para armazenamento de dados escalável, Cloud SQL para um banco de dados gerenciado e App Engine para hospedar nossas aplicações sem a necessidade de gerenciamento de infraestrutura. Isso simplificará significativamente a administração operacional e permitirá que a equipe foque na inovação.

Outrossim, utilizaremos os recursos de balanceamento de carga do Google Cloud para distribuir uniformemente o tráfego entre os servidores, garantindo desempenho consistente. Além

disso, aproveitamos a escalabilidade automática para ajustar dinamicamente os recursos conforme a demanda, otimizando custos e garantindo eficiência operacional.

Adicionalmente, o Google Cloud oferece ferramentas robustas de monitoramento, como o Stackdriver, que fornecerá visibilidade em tempo real do desempenho da aplicação. Juntamente com o Cloud Identity and Access Management (IAM) para gerenciamento de acesso, essas ferramentas garantirão um ambiente seguro e controlado.

A equipe identifica cenários específicos de implantação, considerando requisitos de desempenho, segurança e custo. Cenários de implementação podem incluir a criação de instâncias de máquinas virtuais para partes específicas da aplicação, implementação de serviços gerenciados para funções críticas e a configuração de redes privadas virtuais para garantir a segurança dos dados.

Além disso, o Google Cloud oferece uma gama completa de serviços, desde computação e armazenamento até machine learning e análise de dados. A integração estreita com tecnologias como Kubernetes para orquestração de contêineres e BigQuery para análise de dados em grande escala destaca a versatilidade da plataforma.

O Console do Google Cloud proporciona uma interface intuitiva para gerenciar recursos, enquanto o Cloud Shell oferece uma interface de linha de comando para automação. O suporte eficaz a DevOps é possível com ferramentas como Cloud Build e Deployment Manager.

3.4 ESTRUTURA DE DADOS

Uma das primeiras fases para se elaborar um software, é o levantamento do escopo e de seus requisitos. Segundo o livro Requisitos de Software, de Plínio Ventura, os requisitos funcionais são de extrema importância, pois, sem eles não há funcionalidades e sem funcionalidades não a sistema. Nesse sentido, os requisitos funcionais são as características funcionais que fazem o sistema agir e, muitas vezes, essas estão ligadas diretamente ao cliente.

Já os requisitos não funcionais, estão diretamente ligados aos atributos e qualidades que o programa deve ter, como por exemplo: seu custo, confiança e manutenção. Eles têm um papel de extrema importância para o desenvolvimento do sistema, pois caso ocorra um erro após a incrementação do sistema, pode ser muito difícil de ser resolvido.

3.4.1 LEVANTAMENTO DE REQUISITOS

Dessa forma, levando em consideração a extrema importância de listar esses requisitos. Foi proposto em aula a elaboração de uma tabela, nela foram listados requisitos funcionais como: registro de usuário, anexo de mídias e envio de problemas; e como não funcionais: a facilidade de entendimento, linguagem escolhida para o desenvolvimento e o tempo de resposta do sistema.

REQUISITOS FUNCIONAIS	REQUISITOS NÃO FUNCIONAIS
RF001 Registro de Usuário: Permitir que os usuários criem contas pessoais com informações básicas.	RNF001: O sistema deverá ser desenvolvido em PHP/HTML/CSS/JAVASCRIPT/JAVA.
RF002 Login e Autenticação: Fornecer um sistema de login seguro para acessar a plataforma.	RNF002: O sistema deverá ser de fácil entendimento.
RF003 Envio de Problemas: Permitir que os usuários relatem problemas específicos das cidades, como infraestrutura, serviços públicos, segurança, etc.	RNF003: O sistema deverá conter um sistema de ajuda ao usuário.
RF004 Categorização de Problemas: Oferecer uma variedade de categorias para classificar os problemas relatados, como trânsito, iluminação pública, coleta de lixo, etc.	RNF004: O sistema deverá ter contato com as empresas.
RF005 Anexar Mídias: Permitir que os usuários enviem fotos, vídeos ou documentos relevantes para ilustrar o problema.	RNF005: O sistema deverá marcar pelo menos um tempo de respostas da empresa
RF006 Acompanhamento de Status: Permitir que os usuários acompanhem o status de resolução dos problemas relatados.	RNF006: O sistema deverá possuir modo escuro de tela, para maior proteção ocular para o usuário.
RF007 Notificações: Enviar notificações por e-mail ou push sobre atualizações.	RNF007: O sistema deverá conter rodapé nas páginas informando os desenvolvedores do site.
RF008 Avaliação de Resoluções: Permitir que os usuários avaliem a eficácia das soluções fornecidas pela cidade.	RNF008: O sistema será responsivo.
RF009 Busca Avançada: Fornecer uma funcionalidade de busca avançada para encontrar problemas específicos ou tópicos relevantes	
RF010 Página empresarial: Empresas responsáveis devem possuir uma página individual, onde recebem as denúncias direcionadas.	

RF011 Alertas de Emergência: Oferecer um sistema de alerta rápido para situações de emergência, como desastres naturais ou incidentes graves.	
RF012 Acesso Móvel: Garantir que o sistema seja acessível e funcional em dispositivos móveis.	
RF013 Localização: Permitir que os usuários escolham sua localização para personalizar a denúncia.	
RF014 Feedback dos Usuários: Incluir um mecanismo para os usuários fornecerem feedback sobre a plataforma e suas funcionalidades	
RF015 Integração com Autoridades Públicas: Possibilitar o compartilhamento direto de relatórios com autoridades municipais para agilizar a resolução.	
RF016 FAQ e Centro de Ajuda: Deve ter uma página contendo perguntas frequentes para orientar os usuários sobre como usar a plataforma.	

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS

Estereótipos são representações generalizadas de determinados grupos de pessoas, baseadas em características superficiais como raça, gênero, religião, nacionalidade, entre outros. Essas representações podem ser preconceituosas e limitantes, pois não levam em consideração a individualidade e diversidade presentes em cada pessoa, e a relação entre os estereótipos e o convívio social, que é algo que se refere à interação e relacionamento entre indivíduos, além de ser complexa, pode ter grandes impactos na forma em como nos relacionamos com outros que são diferentes de nós.

3.5.1 ENFRENTANDO ESTEREÓTIPOS

Quando nos baseamos em estereótipos para julgar ou tratar as pessoas, estamos agindo de forma preconceituosa e discriminatória, algo que, pode levar a divisões sociais e discriminação, criando barreiras entre as pessoas e impedindo o desenvolvimento de relacionamentos saudáveis e genuínos. Nesse sentido, se alguém pensa que uma pessoa é pobre por morar em uma favela ou periferia é, com certeza, uma atitude e um pensamento preconceituoso.

Ademais, as redes sociais também ajudam a espalhar muitos estereótipos. A busca validação social através dos likes pode levar à idealizações de uma vida que é completamente diferente da realidade. Nelas, as pessoas querem parecer perfeitas, estabelecendo assim, aos consumidores de conteúdo, o estereótipo de uma vida utópica. Logo, esse fato favorece as pessoas se sentirem mal consigo mesmas e acarreta diversos danos psicológicos porque não se acham boas o suficiente.

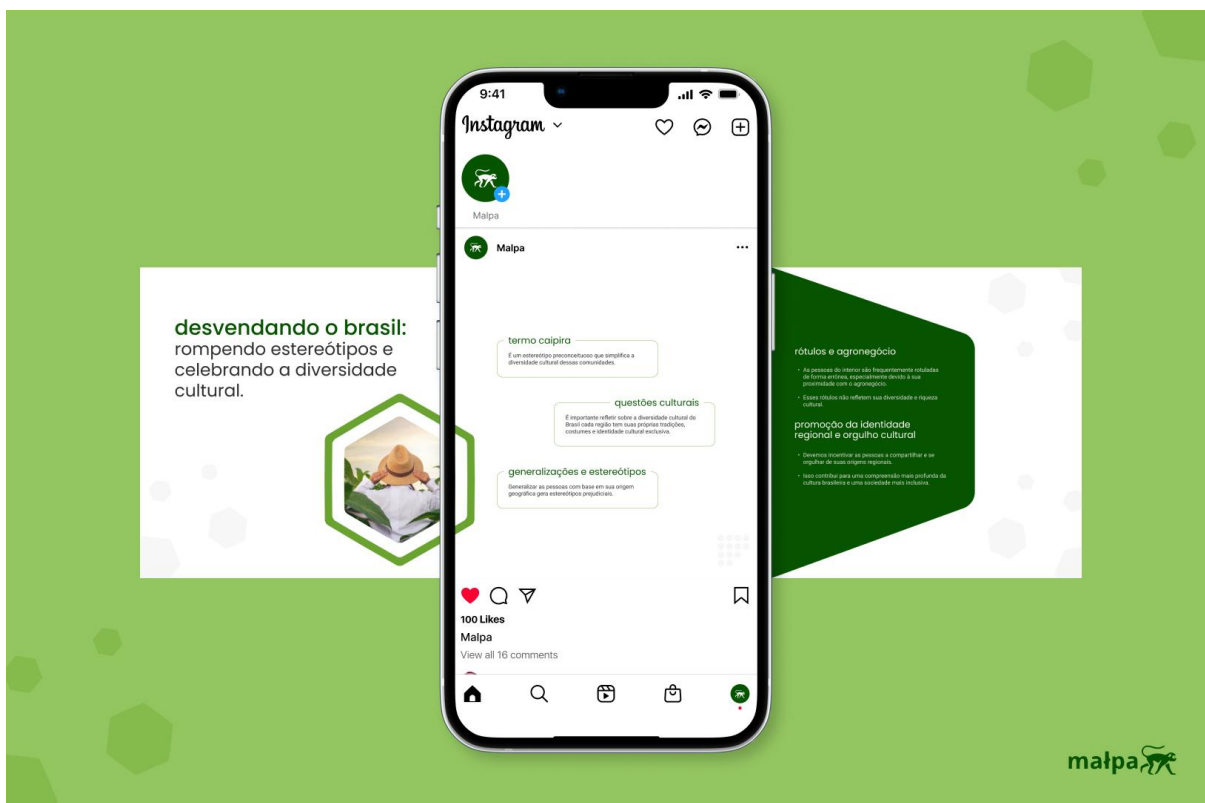
Sendo assim, por exemplo, uma pessoa que migra do interior para uma cidade mais desenvolvida pode ser discriminada, pois as pessoas associam a ela uma imagem ingênua ou menos educada, em consequente, pode levar ao bullying, exclusão social e problemas emocionais. Portanto, é muito importante promover a inclusão e o respeito para garantir que todos tenham oportunidades iguais dentro da sociedade.

3.5.2 ESTUDANTES NA PRÁTICA

Segundo a Forbes, o Brasil é o terceiro maior consumidor de redes sociais em todo o mundo. Sendo assim, foi criado um carrossel no Instagram para conscientizar sobre estereótipos das pessoas do interior.

O carrossel é uma forma de marketing de compartilhar várias imagens em uma única postagem, criando uma sensação de vínculo entre as imagens e possibilitando uma apresentação sequencial de informações. Desse modo, cria-se uma maior interação com o usuário, uma vez que, ao deslizar o dedo, a absorção do conteúdo se torna mais dinâmica. Portanto, coopera para um maior engajamento e compartilhamento das informações e amplia o alcance da conscientização sobre os estereótipos com pessoas que moram na zona rural.

Imagem 12: Carrossel sobre estereótipos



Fonte: autoria própria

Portanto, caso você opte por querer ver essa imagem em tamanho maior, você pode [clique aqui](#). Nesse link, você tem acesso aos textos de forma clara e objetiva e, também, ao guia de estilo e créditos de imagem. Para ver todos esses detalhes, você deve clicar no “F” que está disponível no canto superior esquerdo da tela.

4. CONCLUSÃO

Portanto, o projeto teve como sua principal base criar uma plataforma de gerenciamento de dados em nuvem. Com os avanços dos estudos, pôde-se tomar como nota a importância da computação em nuvem para o desenvolvimento do projeto, pois fornece uma série de servidores responsáveis por armazenar dados de forma segura sem depender de uma construção de um hardware, evitando um gasto desnecessário.

Dessa forma, foi criada a empresa Malpa, responsável por trazer soluções para conectar de forma eficaz os cidadãos aos órgãos responsáveis por resolver questões que afetam o cotidiano da comunidade. O software foi desenvolvido com base em uma arquitetura de computação em nuvem e incorporado técnicas avançadas de gerenciamento de banco de dados, levando em consideração tudo que foi adquirido durante as aulas de Estrutura de Dados, Linguagem e Técnicas de Programação, Tópicos Avançados de Banco de Dados e Computação em Nuvem, para que pudesse disponibilizar uma solução para as metas oferecidas pelas ODS.

REFERÊNCIAS

CYSNEIROS, Luiz. Requisitos Não Funcionais: Da Elicitação ao Modelo Conceitual. Departamento de Informática. Disponível em: <https://www-di.inf.puc-rio.br/~julio/Tese%20-%205.pdf>. Acesso em: Novembro de 2023.

Forbes. Brasil é o terceiro país que mais consome redes sociais em todo o mundo. Disponível em: <https://forbes.com.br/forbes-tech/2023/03/brasil-e-o-terceiro-pais-que-mais-consome-redes-sociais-em-todo-o-mundo/>. Acesso em: Outubro de 2023.

Lafore, R. 2005. Estruturas de Dados e Algoritmos em Java. Ciência Moderna.

Lehmkuhl, D.; Eger, D. R. Princípios de Banco de Dados. UNIASSELVI, 2013. 200 p. Disponível em: <https://www.uniasselvi.com.br/extranet/layout/request/trilha/materiais/livro/livro.php?codigo=13315>. Acesso em: Outubro de 2023.

NoventiQ. IaaS, PaaS e SaaS na Nuvem. Disponível em: <https://noventiq.com.br/sobre-a-empresa/blog/iaas-paas-saas-nuvem>. Acesso em: Novembro de 2023.

Ventura, P. Requisitos de Software. Formato (ebook). Disponível em: <https://www.indtech.com.br/eBookRequisitosSoftwarePlinioVentura.pdf>. Acesso em: Novembro de 2023.

ANEXOS

Imagem 13: Tela de Cadastro



Fonte: autoria própria

Imagem 13: Tela de Logi



Fonte: autoria própria