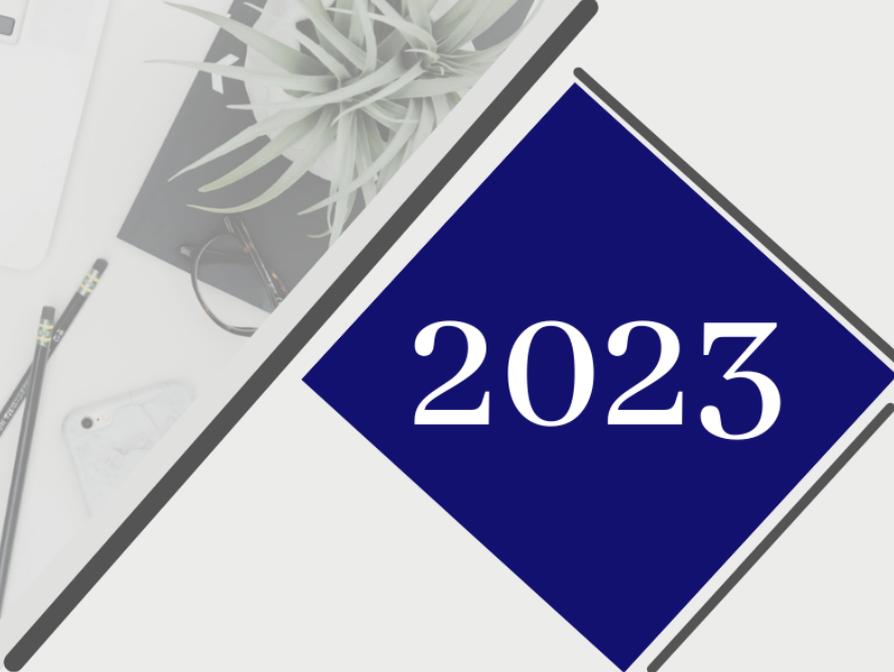


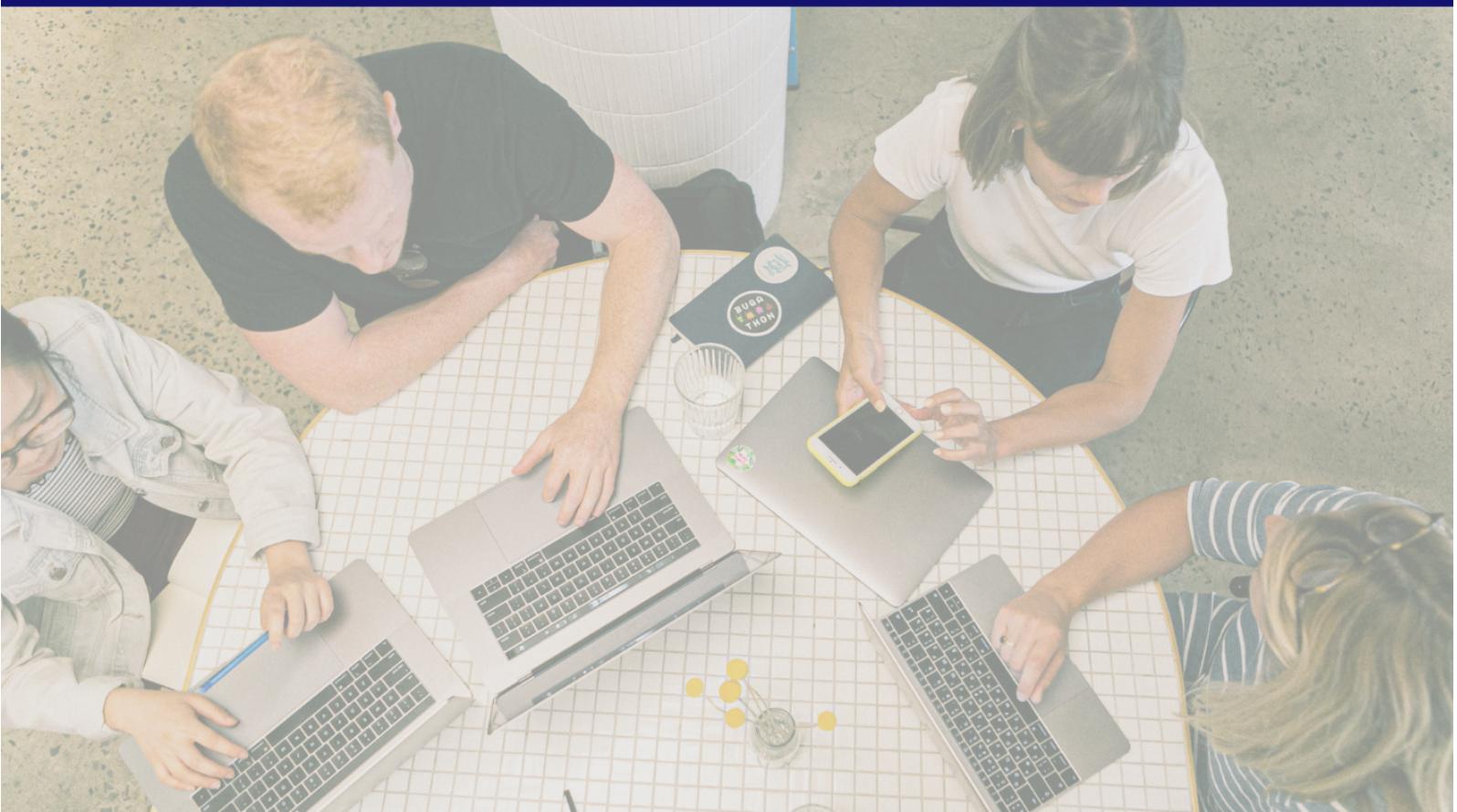


UNifeob
| ESCOLA DE NEGÓCIOS



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

TÍTULO DO PROJETO

INFINITECH

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

INFINITECH

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:

Caio Grilo da Cunha, RA 23001243

Daniel Soares Vanzela, 22001550

Eduardo Riquena Fonsêca, 22001751

Gian Carlos de Freitas Moroni, 22000843

Giovana Chiodeto Teixeira da Silva, 22001211

Haryel Araújo de Oliveira Caliar, 22001470

Jackeline Ayumi Kanekiyo, 22001803

Vyncius Gabriel Oblonczyk, 22001650

SÃO JOÃO DA BOA VISTA, SP

NOVEMBRO 2023

SUMÁRIO

1. INTRODUÇÃO	5
2. DESCRIÇÃO DA EMPRESA	7
3. PROJETO INTEGRADO	8
3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS	8
3.1.1 MODELO LÓGICO	9
3.1.1.1 TABELAS	9
3.1.1.2 RELACIONAMENTOS	10
3.1.2.1 VIEWS	11
3.1.2.2 PROCEDURES:	19
3.1.2.3 FUNCTIONS:	23
3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO	25
3.2.1 PROTOTIPAÇÃO	25
3.2.2 FRONT-END	27
3.2.4 CRUD	28
3.3 COMPUTAÇÃO EM NUVEM	29
3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING	29
3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO	30
3.3.3 VANTAGENS DA CLOUD COMPUTING	30
3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING	31
3.3.5 ESCOLHA DO PROVEDOR DE NUVEM (GOOGLE CLOUD OU AWS)	31
3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING	31
3.3.7 GOOGLE CLOUD ou AWS	32
3.4 ESTRUTURA DE DADOS	33
3.4.1 LEVANTAMENTO DE REQUISITOS	33
3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS	67
3.5.1 ENFRENTANDO ESTEREÓTIPOS	67
3.5.2 ESTUDANTES NA PRÁTICA	68
4. CONCLUSÃO	70
REFERÊNCIAS	71
ANEXOS	72

1. INTRODUÇÃO

Tópicos como o aquecimento global, escassez de água e poluição urbana tornaram-se cada vez mais comuns ao longo do século XXI, sendo agora temas recorrentes de discussão nas reuniões da ONU.

“Chegamos a um ponto na História em que devemos moldar nossas ações em todo o mundo, com maior atenção para as consequências ambientais. Através da ignorância ou da indiferença podemos causar danos maciços e irreversíveis ao meio ambiente, do qual nossa vida e bem-estar dependem. Por outro lado, através do maior conhecimento e de ações mais sábias, podemos conquistar uma vida melhor para nós e para a posteridade, com um meio ambiente em sintonia com as necessidades e esperanças humanas... (ONU, 1972).

Em agosto de 2015, durante a Cúpula das Nações Unidas com foco no Desenvolvimento Sustentável, foi negociada a adoção dos Objetivos de Desenvolvimento Sustentável (ODS) e a criação da Agenda Pós-2015 (Agenda 2030), visando a colaboração de 193 estados membros da Organização das Nações Unidas, totalizando 17 ODS. Elas integram e equilibram três dimensões essenciais do desenvolvimento sustentável, como econômico, social e ambiental (CARLOS GÓMEZ GIL, 2018). Dentre esses objetivos, assuntos como nível de pobreza, fome, educação de qualidade, biodiversidade, questões energéticas, poluição, entre outros, são tratados dentro das políticas nacionais, com data final para o ano de 2030.

A proposta do projeto apresentado tem como função ajudar na resolução de algumas ODS, sendo de impacto nacional e internacional. Consiste no desenvolvimento de um software que tem como objetivo aumentar e incentivar a doação de materiais de reciclagem, agasalhos, alimentos, sangue e outros itens essenciais.

Atendendo ODS como:

- ODS 2: Fome zero;
- ODS 3: Saúde e Bem-Estar;
- ODS 12: Consumo e Produção Sustentáveis;
- ODS 13: Ação contra a Mudança Global do Clima;
- ODS 17: Parcerias para as Metas

O software promove o incentivo através da gamificação, que faz uso do sistema de recompensa, que funciona através da liberação de dopamina, segundo Valenzuela (2016,

apud SUÁREZ, Gonzalo Mantero, 2018, p. 06)¹, o sistema de recompensa atua em ações que nos fazem sentir e nos dão prazer, como é o caso de videogames, esportes de risco ou consumo de certas drogas.

A gamificação funcionará conforme um processo de doação de materiais, visualizando um planeta terra todo degradado e poluído e a cada nível e atividade realizada o planeta irá se tornando mais limpo e habitável.

A início de desenvolver o sistema, foi realizada uma pesquisa direta com potenciais usuários, utilizando a plataforma do Google Forms. Tal pesquisa se orientou pela intenção de coletar informações abrangentes, direcionadas a uma compreensão mais precisa das vicissitudes e desafios associados à promoção da sustentabilidade. Os insights adquiridos desse processo se mostraram fundamentais para informar a direção a ser tomada no desenvolvimento da plataforma, possibilitando a identificação de soluções que endereçaram os entraves delineados pela comunidade.

Para melhor aproveitamento do desenvolvimento do projeto, visando obedecer as datas do cronograma de entregas, foi utilizado metodologia ágil, Al-Saqqa (*et al*, 2020) define que metodologias ágeis são uma estrutura conceitual para engenharia de software que começa com uma fase inicial de planejamento e segue o caminho para a fase de implantação com interações iterativas e incrementais durante todo o ciclo de vida do projeto, a escolhida para este projeto foi a metodologia Scrum, através da ferramenta Trello².

¹ Valenzuela I. ¿Cómo funciona el sistema de recompensa del cerebro? Vix, revista digital. 2016 [En línea]. [Consultado en Abril de 2018]. Disponible en: <https://www.vix.com/es/btg/curiosidades/5190/como-funciona-el-sistema-de-recompensadel-cerebro>;

² Consulte a tabela do trello em anexo.

2. DESCRIÇÃO DA EMPRESA

A empresa que está sendo alvo neste projeto do módulo Computação em Nuvem tem razão social Infitech, com CNPJ 86.244.503/0001-28 de São João da Boa Vista - SP localizado na Etores Martinelli, 1157, Nova São João.

Sua principal atividade é na inovação tecnológica, através de consultorias a pessoas físicas e jurídicas, seu foco é na entrega de softwares e hardwares personalizados para cada demanda singular.

A empresa tem como característica de cliente empresários que buscam trazer tecnologia e soluções para suas empresas.

3. PROJETO INTEGRADO

Para construção da plataforma, uma variedade de conhecimentos da área de tecnologia e elaboração de software foram utilizados, os quais serão descritos nesta sessão.

3.1 TÓPICOS AVANÇADOS DE BANCO DE DADOS

Segundo Date (2003, p.10), um banco de dados pode ser definido como um conjunto de dados armazenados para uso em atividades de uma determinada entidade.

Devido a isso, a importância de definir qual seria o banco de dados ideal para atender a todos os requisitos, se tornou necessário. Isso inclui determinar requisitos de dados, volumes de dados, padrões de acesso e considerações de segurança e conformidade.

A fim de escolher qual banco de dados seria utilizado, foram exploradas as diferenças entre bancos de dados relacionais “SQL” e não relacionais “NoSQL”, destacando suas características, vantagens e desvantagens.

Como a proposta consiste no desenvolvimento de um software que tem como objetivo aumentar e incentivar a doação de materiais de reciclagem, agasalhos, alimentos, sangue e outros itens essenciais, a melhor escolha foi utilizar um modelo em banco de dados relacionais, utilizando a linguagem SQL³, através do software MySQL Workbench da Microsoft, a fim de manter dados com estruturas bem definidas, garantir transações seguras e desejamos manter a integridade dos dados.

A escolha da hospedagem do SGBD, que oferece serviços de armazenamento, consulta e atualização de bancos de dados (Ferreira, Queiroz e Casanova, 2005, p.170), foi em busca de escalabilidade sob demanda, gerenciamento simplificado de infraestrutura e recursos avançados de segurança e conformidade, por fim, define-se utilizar um sistema em nuvem. Abadi (2009)⁴ fala que a hospedagem do SGBD, “Para muitas empresas, especialmente para startups e médias empresas, o pagamento baseado no uso do modelo de computação em nuvem, juntamente com o suporte para manutenção do hardware é muito atraente”.

A seguir, detalha-se a seleção e implementação de uma solução de banco de dados em ambiente cloud, enfatizando desafios técnicos e estratégias de gerenciamento de dados.

³ Structured Query Language

⁴ [Abadi 2009] Abadi, D. J. (2009). Data management in the cloud: Limitations and opportunities. IEEE Data Eng. Bull., 32:3–12.

3.1.1 MODELO LÓGICO

Nesta fase, os conceitos do modelo entidade-relacionamento devem ser traduzidos nas estruturas concretas que compõem a base de dados. Isto envolve a criação de tabelas, a definição de atributos, o estabelecimento de relacionamentos entre entidades e a incorporação de regras de integridade referencial.

O design lógico produz um esquema de banco de dados mais detalhado que fornece informações específicas sobre cada tabela, incluindo nome da tabela, colunas, tipos de dados, tamanho, restrições de integridade (como chaves primárias e estrangeiras) e regras de aplicação.

Isso estabelece a base técnica para a criação e gerenciamento eficientes de bancos de dados. Durante o processo de design lógico, garantimos que todas as decisões de design sejam consistentes com os objetivos do banco de dados, incluindo eficiência de recuperação de dados, integridade de dados e escalabilidade.

Pensando em como o banco de dados iria suprir nosso software desenvolvemos o modelo lógico pensando em algo que seria menos burocrático para o usuário e que nos facilitaria na análise e exportação desses dados.

3.1.1.1 TABELAS

- **Doações:** Armazena informações sobre as doações feitas, como a quantidade doada, descrição, data da doação e IDs das empresas, usuários e tipos de doações relacionados.
- **Empresas:** Contém detalhes sobre empresas, como nome, CNPJ, informações de contato (e-mail, telefone), endereço (CEP, cidade, rua, número, bairro) e datas de criação/atualização.
- **Tipos de Doações (tp_doacoes):** Registra os tipos de materiais doados, a quantidade em pontos e a unidade de medida desses materiais.
- **Usuários:** Armazena dados dos usuários, como nome, sobrenome, data de nascimento, CPF, informações de contato (telefone, e-mail), endereço (CEP) e datas de criação/atualização.
- **Log:** Registra eventos genéricos com seus detalhes e datas de ocorrência.

3.1.1.2 RELACIONAMENTOS

- **Doações e Empresas:** Relacionadas por IDs para indicar de qual empresa veio a doação.

Relação: Muitos para um (1 para Muitos)

Várias doações podem ser feitas por uma única empresa, mas cada doação está associada a uma única empresa doadora. Portanto, é um relacionamento onde várias doações podem estar vinculadas a uma única empresa doadora.

- **Doações e Usuários:** Relacionadas por IDs para indicar quem fez a doação.

Relação: Muitos para um (1 para Muitos).

Diversas doações podem ser feitas por um único usuário, mas cada doação está ligada a um único usuário que a realizou. Assim, várias doações podem ser atribuídas a um único usuário.

- **Doações e Tipos de Doações:** Relacionadas por IDs para especificar o tipo de material doado e sua quantidade.

Relação: Muitos para um (1 para Muitos).

Múltiplas doações podem ser do mesmo tipo de doação. Cada doação está associada a um único tipo de doação, mas vários registros de doações podem ter o mesmo tipo.

- **Log:** Pode ser usado para registrar eventos relacionados às doações, empresas, usuários. Isso permite registrar informações sobre as doações (quem doou, para quem, o que foi doado, quando), detalhes das empresas, tipos de doações possíveis, detalhes dos usuários e eventos genéricos que podem ser relevantes para o sistema.

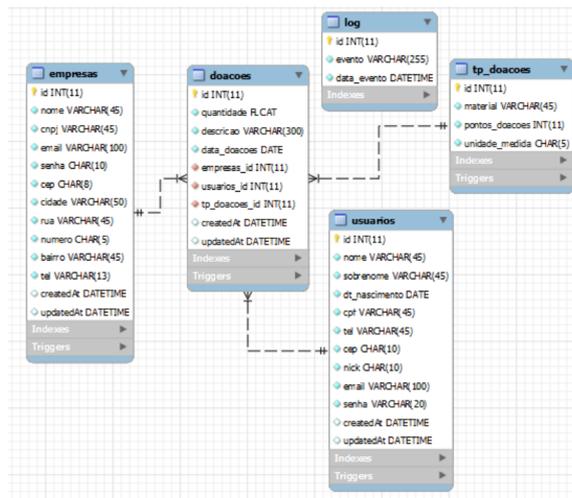
Quanto às relações de cada tabela, decidimos da seguinte forma:

Relação: Mantivemos sem relação

Pois um evento registrado pode estar relacionado a várias entidades diferentes (doações, empresas, usuários), e uma única entidade (doação, empresa, usuário) pode estar associada a vários eventos de log.

Observe como ficou nosso modelo lógico de banco de dados do nosso software.

Imagem 1: Modelo Lógico do Projeto



Fonte: Autoria própria

3.1.2 MODELO FÍSICO

Nesta fase são estabelecidos os detalhes de como os dados serão armazenados, como a organização de tabelas, índices e partições, criar scripts SQL para criar objetos no banco de dados, incluindo tabelas, visualizações, colunas, funções, gatilhos e procedimentos armazenados, esses arquivos são necessários para configurar o repositório de acordo com o projeto, as permissões de acesso são definidas para garantir que os usuários tenham o nível apropriado de acesso aos dados e para garantir segurança e integridade.

Define-se então, as configurações de desempenho do banco de dados, como base, partição e configurações de memória nesta seção, os projetos da entidade estão intimamente relacionados ao SGBD selecionado, pois cada SGBD tem seus próprios requisitos e melhorias.

3.1.2.1 VIEWS

1. PESQUISA DE DADOS BÁSICOS DE USUÁRIOS:

A view abaixo seleciona os dados do usuário exceto os dados sigilosos, como por exemplo o de login. Vale lembrar que nesta temos a possibilidade de selecionar pelo ID, graças ao WHERE. Com isso trazemos alguns dados do usuário para as empresas visualizarem.

Imagem 2: Pesquisa de Usuário

```
88 -----
89 • select * from dadosUsuarioBasic where id = 1;
90
91 • create view dadosUsuarioBasic as
92 select a.id, a.nome, a.sobrenome, a.cpf, a.dt_nascimento, a.cep, a.tel, a.nick
93 from usuarios a;
94 -----
95
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

id	nome	sobrenome	cpf	dt_nascimento	cep	tel	nick
1	DUNHA	FONSECA	12345678900	1999-03-08	12345678	12345678900	EDU
2	Viny	gaySR	12358746554	2003-11-04	13870000	19123456789	VGay
3	MAYUMI	CHAN	84620974551	2004-02-08	13710000	1999557157	JAJACK

Fonte: Autoria própria

2. PESQUISA DE DADOS BÁSICOS DA EMPRESA

Nesta, a view é criada da mesma forma, entretanto com dados da empresa, disponibilizando para o usuário alguns dados básicos. O objetivo é fazer com que o usuário possa visualizar de forma clara.

Imagem 3: Pesquisa de Dados Empresa

```
94 -----
95 • SELECT * FROM dadosEmpresaBasic;
96
97 • create view dadosEmpresaBasic as
98 select a.id, a.nome, a.cnpj, a.email, concat
99 ("Cep: ",a.cep, " - ", a.cidade, " - ",a.rua, ", ",a.numero, " - ",a.bairro)
100 as 'Endereço Completo' , a.tel from empresas a;
101 -----
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: IA

id	nome	cnpj	email	Endereço Completo	tel
1	TESTE1	12345678000199	TESTE1@TESTE.COM	Cep: 12345678 - CIDADE TESTE - RUA TESTE, ...	12345678900
2	TESTE2	12345678000199	TESTE2@TESTE.COM	Cep: 12345678 - CIDADE TESTE - RUA TESTE, ...	12345678900
4	TESTE AYUMI-CHAN	12965675730199	AYUMINHADAVL@TESTE.COM	Cep: 13710000 - CIDADE DE DEUS - RUA DOS ...	19992557157

Fonte: Autoria própria

3. PESQUISA DE DADOS DE LOGIN DE USUÁRIO:

Podemos exibir a senha e o login através dessa view, o objetivo é de modo geral, permitir ao usuário pesquisar seus próprios dados para, se necessário, fazer alguma alteração.

Imagem 4: Pesquisa de Login Usuário

```
111 -----
112 • SELECT * FROM loginUsuario where nick = 'EDU';
113 • create view loginUsuario as
114   select a.nick, a.senha, a.email from usuarios a;
115
116 -----
```



nick	senha	email
EDU	54321	TESTE@gmail.com

Fonte: Autoria própria

4. PESQUISA DE DADOS DE LOGIN DE EMPRESA:

Podemos exibir a senha e o login através dessa view, o objetivo é de modo geral, permitir a empresa pesquisar seus próprios dados para, se necessário, fazer alguma alteração

Imagem 5: Pesquisa de Login Empresa

```
116 -----
117 • select * from loginEmpresa where email = 'AYUMINHADAVL@TESTE.COM';
118 • create view loginEmpresa as
119   select a.email, a.senha from empresas a;
120
121 -----
122
```



email	senha
AYUMINHADAVL@TESTE.COM	44444

Fonte: Autoria própria

5. PESQUISA DE NICK POR CPF:

Nesta view podemos pesquisar o nick pelo CPF dentro do backend, isto se deve graças ao sequelize. O Sequelize é uma biblioteca JavaScript que funciona como um *Object-Relational Mapping*.

Imagem 6: Pesquisa Nick por CPF

```
121 -----
122
123 -----
124
125 • select a.nick from usuarios a where cpf = 12345678900
126
127 -----
128
```



nick
EDU

Fonte: Autoria própria

6. SELEÇÃO DE TOP 4 DOAÇÕES GERAL :

Nesta consulta estamos buscando pelas quatro principais doações de modo geral, a consulta nos retorna os dados com base no valor da doação. O objetivo é visualizar os maiores doadores entre os usuários.

Imagem 7: Seleção Top Doadores

```
125 -----
126 • select * from top4Doacao;
127 • alter VIEW top4Doacao AS
128 SELECT c.id, c.quantidade, c.descricao
129 FROM doacoes c
130 ORDER BY quantidade DESC
131 LIMIT 4;
132 -----
133
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

id	quantidade	descricao
1	100	latao
8	75	Garrafinha
6	40	Latinha
2	25	Garrafas

Fonte: Autoria própria

7. SELEÇÃO DE DOAÇÃO POR PERÍODO:

Neste modelo podemos selecionar as doações por data, a fim de facilitar as consultas para o usuário. A tabela de doações é acionada com a condição dada pelo usuário.

Imagem 8: Seleção Doadores por Período

```
134 -----
135 -----
136 -----
137 • select * from doacoes where data_doacoes = '2023-10-18' -- será rodado dentro do beackend para selecionar o período;
138 -----
139 -----
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

id	quantidade	descricao	data_doacoes	empresas_id	usuarios_id	tp_doacoes_id
6	40	Latinha	2023-10-18	1	1	1
*	NULL	NULL	NULL	NULL	NULL	NULL

Fonte: Autoria própria

8. SELEÇÃO DE PESO TOTAL POR TIPO DE MATERIAL:

Calcula o peso total para cada tipo de material. A visão é construída a partir da tabela *doacoes* e da tabela relacionada *tp_doacoes*, agrupando os resultados pelo tipo de material e exibindo a soma da quantidade como 'Peso'

Imagem 9: Seleção Peso por Material

```
143 -----
144 • select * from pesoTotal;
145
146 -- pesquisa de peso totais
147 • create view pesoTotal as
148 SELECT b.material AS 'Descricao', SUM(a.quantidade) AS 'Peso'
149 FROM doacoes a
150 INNER JOIN tp_doacoes b ON a.tp_doacoes_id = b.id
151 GROUP BY b.material
152 ORDER BY b.material;
153 -----
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Descricao	Peso
▶	FERRO	140
	PLASTICO	100

Fonte: Autoria própria

9. PESQUISA DE ENDEREÇO DA EMPRESA:

Combina informações da tabela empresas. A visão seleciona o nome da empresa como 'Empresa' e cria uma string concatenada para representar o 'Endereço Completo', incluindo o CEP, rua, número, bairro e cidade.

Imagem 10: Endereço da Empresa

```
163 -----
164
165 • select * from enderecoEmpresa;
166
167 • create view enderecoEmpresa as
168 SELECT a.nome AS 'Empresa', CONCAT('CEP: ',a.cep, ' - ', a.rua, ' ', a.numero, ' - ', a.bairro, ' - ', a.cidade) AS 'Endereço Completo'
169 FROM empresas a
170
171 -----
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Empresa	Endereço Completo
▶	TESTE1	CEP: 12345678 - RUA TESTE, 123 - BAIRRO TE...
	TESTE2	CEP: 12345678 - RUA TESTE, 123 - BAIRRO TE...
	TESTE AYUMI-CHAN	CEP: 13710000 - RUA DOS DOGS, 321 - BAIRR...

Fonte: Autoria própria

10. SELEÇÃO DE PONTOS TOTAIS:

Calcula os 'Pontos Totais' para cada usuário, com base nas doações feitas. A visão envolve as tabelas *usuarios*, *doacoes*, e *tp_doacoes*, relacionando-as através de chaves estrangeiras. Os resultados são agrupados pelo 'Nick' do usuário e ordenados em ordem decrescente. Pode ser usado um WHERE para filtrar os resultados para um usuário específico, como indicado no comentário

Imagem 11: Seleção por Pontos Totais

```
171 -----
172 select * from pontosTotais
173
174 create view pontosTotais as
175 SELECT SUM(b.quantidade * c.pontos_doacoes) AS 'Pontos Totais', a.nick AS 'Nick'
176 FROM usuarios a
177 INNER JOIN doacoes b ON a.id = b.usuarios_id
178 INNER JOIN tp_doacoes c ON b.tp_doacoes_id = c.id
179 -- WHERE a.nick = 'EDU' (PODE SER INSERIDO UM WHERE PARA USAR A VIEW PARA CADA USUARIO)
180 GROUP BY a.nick
181 ORDER BY a.nick DESC;
182 -----
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [FA](#)

Pontos Totais	Nick
1900	EDU

Fonte: A autoria própria

11. SOMA OS PONTOS TOTAIS POR USUÁRIO:

Esta consulta SQL cria um resumo de pontos totais para cada usuário, multiplicando a quantidade de doações pelo valor dos pontos de doação correspondentes. Os resultados são agrupados por apelido (Nick) e ordenados em ordem decrescente. Pode-se adicionar um WHERE para filtrar por usuário, como exemplificado com 'EDU'.

Imagem 12: Soma dos Pontos por Usuários

```
183 -----
184 select * from pontosOrganizados
185 create view pontosOrganizados as
186 SELECT c.material as 'Material', (b.quantidade * c.pontos_doacoes) AS 'Pontos', a.nick AS 'Nick'
187 FROM usuarios a
188 INNER JOIN doacoes b ON a.id = b.usuarios_id
189 INNER JOIN tp_doacoes c ON b.tp_doacoes_id = c.id
190 -- WHERE a.nick = 'EDU' -- (PODE SER INSERIDO UM WHERE PARA USAR A VIEW PARA CADA USUARIO)
191 GROUP BY a.nick, c.material
192 ORDER BY c.material DESC;
193 -----
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [FA](#)

Material	Pontos	Nick
PLASTICO	125	EDU
FERRO	1000	EDU

Fonte: A autoria própria

3.1.2.2 PROCEDURES:

1. Delete:

delete_doacoes: Exclui registros da tabela "doacoes" com base no valor do parâmetro p_id.

BEGIN

DELETE FROM doacoes WHERE id = p_id;

END

delete_empresas: Exclui registros da tabela "empresas" com base no valor do parâmetro p_id.

BEGIN

DELETE FROM empresas WHERE id = p_id;

END

delete_tp_doacoes: Exclui registros da tabela "tp_doacoes" com base no valor do parâmetro p_id.

BEGIN

DELETE FROM tp_doacoes WHERE id = p_id;

END

delete_usuarios: Exclui registros da tabela "usuarios" com base no valor do parâmetro p_id.

BEGIN

DELETE FROM usuarios WHERE id = p_id;

END

2. Insert:

insert_doacoes: Insere novos registros na tabela "doacoes" com base nos parâmetros fornecidos.

```
BEGIN
```

```
INSERT INTO doacoes (quantidade, descricao, data_doacoes, empresas_id,  
usuarios_id, tp_doacoes_id)
```

```
VALUES (p_quantidade, p_descricao, p_data_doacoes, p_empresas_id,  
p_usuarios_id, p_tp_doacoes_id);
```

```
END
```

insert_empresas: Insere novos registros na tabela "empresas" com base nos parâmetros fornecidos.

```
BEGIN
```

```
INSERT INTO empresas (nome, cnpj, email, senha, cep, cidade, rua, numero, bairro,  
tel)
```

```
VALUES (p_nome, p_cnpj, p_email, p_senha, p_cep, p_cidade, p_rua, p_numero,  
p_bairro, p_tel);
```

```
END
```

insert_tp_doacoes: Insere novos registros na tabela "tp_doacoes" com base nos parâmetros fornecidos.

```
BEGIN
```

```
INSERT INTO tp_doacoes (material, pontos_doacoes, unidade_medida)
```

```
VALUES (p_material, p_pontos_doacoes, p_unidade_medida);
```

END

insert_usuarios: Insere novos registros na tabela "usuarios" com base nos parâmetros fornecidos.

BEGIN

INSERT INTO usuarios (nome, sobrenome, dt_nascimento, cpf, tel, cep, nick, email, senha)

VALUES (p_nome, p_sobrenome, p_dt_nascimento, p_cpf, p_tel, p_cep, p_nick, p_email, p_senha);

END

3. Update:

update_doacoes: Atualiza registros na tabela "doacoes" com base no valor do parâmetro p_id e nos novos valores fornecidos.

BEGIN

UPDATE doacoes

SET quantidade = p_quantidade, descricao = p_descricao, data_doacoes = p_data_doacoes, empresas_id = p_empresas_id, usuarios_id = p_usuarios_id, tp_doacoes_id = p_tp_doacoes_id

WHERE id = p_id;

END

update_empresas: Atualiza registros na tabela "empresas" com base no valor do parâmetro p_id e nos novos valores fornecidos.

BEGIN

UPDATE empresas

SET nome = p_nome, cnpj = p_cnpj, email = p_email, senha = p_senha, cep = p_cep, cidade = p_cidade, rua = p_rua, numero = p_numero, bairro = p_bairro, tel = p_tel

WHERE id = p_id;

END

update_tp_doacoes: Atualiza registros na tabela "tp_doacoes" com base no valor do parâmetro p_id e nos novos valores fornecidos.

BEGIN

UPDATE tp_doacoes

SET material = p_material, pontos_doacoes = p_pontos_doacoes, unidade_medida = p_unidade_medida

WHERE id = p_id;

END

update_usuarios: Atualiza registros na tabela "usuarios" com base no valor do parâmetro p_id e nos novos valores fornecidos.

BEGIN

UPDATE usuarios

SET nome = p_nome, sobrenome = p_sobrenome, dt_nascimento = p_dt_nascimento, cpf = p_cpf, nick = p_nick, email = p_email, senha = p_senha

WHERE id = p_id;

END

3.1.2.3 FUNCTIONS:

calcularPontosDoacao: Calcula o total de pontos de uma doação, multiplicando a quantidade pela pontuação associada, filtrando pelo tipo de doação e pelo usuário.

BEGIN

DECLARE total_pontos INT;

SELECT SUM(a.quantidade * b.pontos_doacao)

INTO total_pontos

FROM doacao a

INNER JOIN tp_doacao b ON (a.id = b.id)

WHERE a.tp_doacao_id = 1 AND a.usuario_id = usuario_id;

RETURN total_pontos;

END

calcularPontosdoacoes: Calcula o total de pontos de doações, multiplicando a quantidade pela pontuação associada, filtrando pelo tipo de doação e pelo usuário.

BEGIN

DECLARE total_pontos INT;

```

SELECT SUM(a.quantidade * b.pontos_doacoes)

INTO total_pontos

FROM doacoes a

INNER JOIN tp_doacoes b ON (a.id = b.id)

WHERE a.tp_doacoes_id = 1 AND a.usuarios_id = usuarios_id;

RETURN total_pontos;

END

```

3.1.2.4 TRIGGERS:

doacoes_AFTER_INSERT

Evento: Disparada após a inserção de um novo registro na tabela doacoes.

Ação: Obtém o nome da empresa que realizou a inserção usando o empresas_id do novo registro inserido na tabela doacoes. Em seguida, registra no log a adição do novo registro na tabela doacoes, mencionando a empresa responsável e a data do evento.

```

doacoes_AFTER_INSERT INSERT doacoes

"BEGIN

DECLARE nome_empresas VARCHAR(255);

-- Obter o nome da empresas que fez a inserção

SELECT nome INTO nome_empresas FROM empresas WHERE id =
NEW.empresas_id;

-- Inserir a mensagem no log com o nome da empresas

INSERT INTO log (evento, data_evento)

VALUES (CONCAT('Novo registro inserido na tabela doacoes por ',
nome_empresas), NOW());

```

END"

doacoes_AFTER_UPDATE

Evento: Disparada após a atualização de um registro na tabela doacoes.

Ação: Similar à trigger de inserção, porém acionada após uma atualização. Obtém o nome da empresa relacionada ao registro atualizado e registra no log a atualização na tabela doacoes, mencionando a empresa responsável e a data do evento.

doacoes_AFTER_UPDATE UPDATE doacoes

"BEGIN

DECLARE nome_empresas VARCHAR(255);

-- Obter o nome da empresas que fez a inserção

SELECT nome INTO nome_empresas FROM empresas WHERE id =
NEW.empresas_id;

-- Inserir a mensagem no log com o nome da empresas

INSERT INTO log (evento, data_evento)

VALUES (CONCAT('Registro atualizado na tabela doacoes por ', nome_empresas),
NOW());

END"

doacoes_AFTER_DELETE

Evento: Disparada após a exclusão de um registro na tabela doacoes.

Ação: Obtém o nome da empresa relacionada ao registro excluído e registra no log a remoção do registro na tabela doacoes, mencionando a empresa responsável e a data do evento.

doacoes_AFTER_DELETE DELETE doacoes

"BEGIN

DECLARE nome_empresas VARCHAR(255);

-- Obter o nome da empresas que fez a inserção

```
SELECT nome INTO nome_empresas FROM empresas WHERE id =  
old.empresas_id;
```

-- Inserir a mensagem no log com o nome da empresas

```
INSERT INTO log (evento, data_evento)
```

```
VALUES (CONCAT('Registro deletado na tabela doacoes por ', nome_empresas),  
NOW());
```

```
END"
```

empresas_AFTER_INSERT

Evento: Disparada após a inserção de um novo registro na tabela empresas.

Ação: Registra no log a adição de uma nova empresa na tabela empresas, registrando a data do evento.

```
empresas_AFTER_INSERT INSERT empresas
```

```
"BEGIN
```

-- Inserir a mensagem no log com o nome da empresas

```
INSERT INTO log (evento, data_evento)
```

```
VALUES ('Nova empresa cadastrada ',NOW());
```

```
END"
```

empresas_AFTER_UPDATE

Evento: Disparada após a atualização de um registro na tabela empresas.

Ação: Obtém o nome da empresa atualizada e registra no log a atualização na tabela empresas, mencionando a empresa alterada e a data do evento.

```
empresas_AFTER_UPDATE UPDATE empresas
```

```
"BEGIN
```

```

DECLARE nome_empresas VARCHAR(255);

-- Obter o nome da empresas que fez a ATUALIZACAO

SELECT nome INTO nome_empresas FROM empresas WHERE id = NEW.id;

-- Inserir a mensagem no log com o nome da empresas

INSERT INTO log (evento, data_evento)

VALUES (CONCAT('Dados atualizados pela empresa ', nome_empresas), NOW());

END"

```

tp_doacoes_AFTER_INSERT

Evento: Disparada após a inserção de um novo registro na tabela tp_doacoes.

Ação: Registra no log a adição de um novo tipo de doação na tabela tp_doacoes, indicando a data do evento.

```
tp_doacoes_AFTER_INSERT INSERT tp_doacoes
```

```
"BEGIN
```

```
    -- Inserir a mensagem no log com o novo registro
```

```
    INSERT INTO log (evento, data_evento)
```

```
    VALUES ('Registro inserido na tabela tipo de doações', NOW());
```

```
END"
```

tp_doacoes_AFTER_UPDATE

Evento: Disparada após a atualização de um registro na tabela tp_doacoes.

Ação: Registra no log a atualização na tabela tp_doacoes, indicando a data do evento.

```
tp_doacoes_AFTER_UPDATE UPDATE tp_doacoes "BEGIN
```

```
    -- Inserir a mensagem no log com o atualiação do registro
```

```
    INSERT INTO log (evento, data_evento)
```

```
    VALUES ('Registro atualizado na tabela tipo de doações', NOW());
```

END"

usuarios_AFTER_INSERT

Evento: Disparada após a inserção de um novo registro na tabela usuarios.

Ação: Obtém o nome do usuário inserido e registra no log o cadastro do novo usuário na tabela usuarios, indicando a data do evento.

usuarios_AFTER_INSERT INSERT usuarios

"BEGIN

DECLARE nome_usuario VARCHAR(255);

-- Obter o nome do usuario que fez a inserção

SELECT nome INTO nome_usuario FROM usuarios WHERE id = NEW.id;

-- Inserir a mensagem no log com o nome da usuarios

INSERT INTO log (evento, data_evento)

VALUES (CONCAT('Novo usuario cadastrado ',nome_usuario),NOW());

END"

usuarios_AFTER_UPDATE

Evento: Disparada após a atualização de um registro na tabela usuarios.

Ação: Obtém o nome do usuário atualizado e registra no log a atualização na tabela usuarios, indicando a data do evento.

usuarios_AFTER_UPDATE UPDATE usuarios

"BEGIN

DECLARE nome_usuario VARCHAR(255);

-- Obter o nome do usuario que fez a ATUALIZACAO

SELECT nome INTO nome_usuario FROM usuarios WHERE id = NEW.id;

-- Inserir a mensagem no log com o nome do usuario

```
INSERT INTO log (evento, data_evento)
VALUES (CONCAT('Dados atualizados pelo usuario ', nome_usuario), NOW());
END"
```

3.2 LINGUAGEM E TÉCNICAS DE PROGRAMAÇÃO

Abrangendo os estágios de prototipagem, criação de front-end, criação de back-end e integração de repositório, o desenvolvimento web apresentou uma jornada desafiadora.

A colaboração entre a equipe de Front-End e Back-End foi crucial para garantir a integração perfeita das partes da aplicação. O desenvolvimento de uma aplicação web é um processo contínuo. Receber feedback do professor responsável foi parte crucial para realizar melhorias e adicionar novos recursos.

O ciclo de desenvolvimento não termina com a implantação, mas continua à medida que a aplicação evolui para atender às necessidades em constante mudança.

3.2.1 PROTOTIPAÇÃO

Como ideia parte do desenvolvimento de um software gamificado com o foco em ajudar o meio ambiente, foi optado por usar cores primárias que chamem atenção, como intuito de manter uma plataforma clean.

O protótipo foi desenvolvido utilizando a plataforma Figma, durante esta etapa, explorasse ideias e conceitos, criando protótipos de baixa e alta fidelidade para visualizar a interface e a experiência do usuário. Essa fase permite refinar ideias e alinhar as expectativas da equipe, onde consegue chegar a uma noção real de como a ideia estava sendo implementada e de como o projeto ficaria esteticamente. Ficou definido que todo o layout seria em lowpoly dando realmente a impressão de um desenho animado.

Low Poly é uma técnica de ilustração que consiste basicamente em traçar polígonos por cima de uma imagem e depois colorir os polígonos usando as cores da imagem como referência. Nos gráficos 3D, as superfícies são formadas por malhas (mesh) compostas por polígonos. Quando as malhas tem milhares de polígonos bem pequenos temos alta resolução (High Resolution), com superfícies bem detalhadas e transições suaves. Cryah (2015).

Para a logo do software foi escolhido uma capivara em desenho 3D com uma roupa de astronauta, conforme imagem 13, a escolha de um animal como mascote possui o significado de que a plataforma trabalha com aspectos ambientais, como também o animal ser de grande importância dentro da fauna brasileira, e muito presente na região onde a empresa está localizada. O uso da roupa de astronauta se dá pelo fato de possuir um tema tecnológico.

Imagem 13: Logo Capivara Astronauta



Fonte: Autoria própria

O nome do sistema ficou definido como “Building My World” que em tradução para o português do Brasil, fica “Construindo Meu Mundo”, um nome claro e com grande impacto.

O guia de estilo ficou definido com as seguintes cores e fontes das letras, para os textos com destaques utilizamos a fonte “**MONTERRAT**” e para os textos sem destaques “**ROBOTO**”.

Imagem 14: Guia de Estilo



Fonte: Autoria Própria

Como o intuito era ter um software divertido, os elementos como os botões, foram desenvolvidos com suas bordas mais arredondadas, para conseguir trazer algo não muito sério, define-se esse padrão em todas as telas.

Imagem 15: Design de Botão



Fonte: Autoria Própria

E em cada tela pode-se encontrar uma imagem com desenhos, como segue imagens em anexos.

A prototipação facilita no desenvolvimento final do projeto, pois tem suas cores, ícones, imagens e proporções já definidas e validadas por toda a equipe.

3.2.2 FRONT-END

Com o protótipo pronto do software, inicia-se o desenvolvimento do Front-End, que pode ser definido como a "cara" da aplicação, sua construção envolve a criação de interfaces atraentes e interativas.

A equipe focou na usabilidade e na experiência do usuário. A seleção de tecnologias como HTML, CSS, JavaScript e frameworks modernos foi fundamental para a criação de um Front-End robusto e amigável.

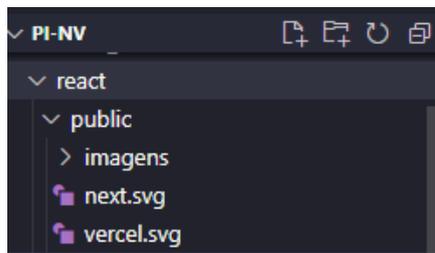
No desenvolvimento do Front-End do software utilizando o VSCode para transformar o protótipo em aplicação e, para isso, foram utilizadas três bibliotecas, a principal sendo o ReactJS, que foi aplicado em JavaScript, uma para máscara de input onde foi estilizado as informações inseridas pelo usuários em formulários, formatando esses dados para facilitar a validação e outra, sendo uma biblioteca de ícones, a MUI, utilizada pelo Spotify, Netflix, Nasa, Amazon, a final os ícones deixam a aplicação bem mais interessante e elegante, além de deixar o site fácil de entendimento, melhorando assim a experiência de usabilidade do usuário.

O foco do ReactJS é criar interfaces em JavaScript para páginas web, com diversas telas e conseguindo trabalhar e padronizar cada uma delas.

Pode-se dizer que o React decompõe uma tela em vários componentes para, em seguida, trabalhar sobre cada uma delas de forma individualizada. Os componentes permitem o reaproveitamento de códigos e a padronização de interfaces, o que faz do React uma tecnologia extremamente flexível para a solução de problemas e para a construção de interfaces reutilizáveis, afinal, cada componente pode ser trabalhado de modo distinto. DURÃES (2021).

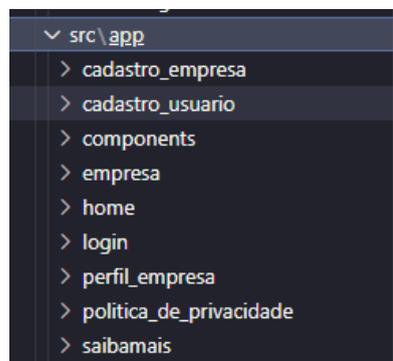
Para a interface foi utilizado o ReactJS, criando duas pastas principais dentro o arquivo, a pasta "*Public*", onde foi colocado uma subpasta de "imagens", alocando todos os arquivos em *.svg*, podendo assim concentrar todas as imagens do software em apenas um local deixando o código clean e com caminhos bem definidos. A outra pasta foi a "*src*" com a subpasta "*app*", onde foram desenvolvidos todos os arquivos das telas da página web, cada uma delas tem sua interface exclusiva.

Imagem 16: Pasta *public*



Fonte: Autoria Própria

Imagem 17: Pasta *src*



Fonte: Autoria Própria

Para facilitar as solicitações HTTP e a comunicação eficiente entre o Front-End e o Back-End, a equipe de desenvolvimento optou pelo uso da biblioteca Axios no lado do cliente. O Axios é uma biblioteca JavaScript baseada em *promises* que, simplifica a realização de requisições HTTP assíncronas. Essa escolha proporciona uma integração suave entre o NodeJS no servidor e o Front-End, permitindo a troca de dados de forma eficaz.

3.2.3 BACK-END

Para o desenvolvimento do Back-End, foi utilizado o servidor o NodeJS, interligando com Express, EJS e Sequelize, mediados pela linguagem de programação JavaScript. O Back-End é, segundo Canady *et al* (1974, p.4), como uma interface entre o computador host e seu banco de dados. Nesta etapa, foi selecionado tecnologias de servidor, desenvolvendo APIs robustas e criando uma base sólida para a funcionalidade da aplicação.

A fim de criptografar informações sensíveis foi escolhida a biblioteca *bcrypt*, que por sua vez possui a função de “Salt”, que significa literalmente “Tempero”, portanto a equipe de

desenvolvimento define quanto “Tempero” ou criptografia será implementada nos dados, juntamente foi utilizado JSON Web Token para realizar logins e autenticação, através de tokens digitais, também chamados de assinaturas digitais, que seguem um padrão de segurança aberto (RFC 7519).

O Sequelize, incorporado ao ecossistema, desempenhou um papel vital na implementação e gestão eficiente do banco de dados relacional. Ao facilitar a interação com o banco de dados SQL por meio de uma abordagem baseada em modelos e consultas em JavaScript, o Sequelize simplificou significativamente o processo de desenvolvimento, proporcionando uma camada de abstração poderosa sobre as operações do banco de dados.

A capacidade de definir modelos de dados, estabelecer relações entre tabelas e realizar migrações de esquema de forma programática permitiu à equipe criar uma base de dados robusta e escalável. Além disso, o suporte do Sequelize a vários dialetos SQL oferece flexibilidade na escolha do banco de dados, alinhando-se perfeitamente às exigências específicas do projeto. Ao integrar o Sequelize com outras tecnologias do Back-End, como NodeJS, Express e EJS, a equipe consolidou uma infraestrutura coesa para o desenvolvimento de APIs robustas, promovendo a funcionalidade e a estabilidade da aplicação.

3.2.4 CRUD

A etapa final deste projeto consistiu em conectar o sistema web com um banco de dados em nuvem. Isso permitiu que as informações geradas dentro da aplicação fossem armazenadas de forma segura e acessadas de maneira eficiente.

A escolha de um banco de dados em nuvem foi baseada em considerações de escalabilidade, segurança e acessibilidade, e essa integração foi crucial para a funcionalidade completa da aplicação.

Com a integração do banco de dados em nuvem, a aplicação foi capaz de realizar todas as operações fundamentais de um sistema web, incluindo Create (Criação), Read (Leitura), Update (Atualização) e Delete (Exclusão), comumente conhecidas como operações CRUD. Isso permitiu aos usuários criar, visualizar, atualizar e excluir informações de acordo com suas necessidades, concluindo o fluxo completo da aplicação.

3.3 COMPUTAÇÃO EM NUVEM

3.3.1 OBJETIVOS DO PROJETO DE CLOUD COMPUTING

Conforme definido por Mell e Grace (2011)⁵, a computação em nuvem é um modelo computacional que permite o acesso global, adequado, sob demanda e diante a rede a um *pool* de recursos computacionais configuráveis.

Adotar o paradigma de computação em nuvem para o sistema, se tornou prioritário, visto o grande número de tarefas e acessos que irão acontecer de maneira momentânea, sendo possível com o controle de escalabilidade.

Para a utilização, o modelo solicitado que mais atenderia as necessidades é o de *Software* como um Serviço (SaaS), onde o consumidor contrata a utilização de uma aplicação que está hospedada e executada na nuvem (LIMA, 2014). A escolha se deu por motivo de que o objetivo deste produto é fornecer o serviço gamificado para usuários, não havendo necessidade de configuração da infraestrutura.

A aplicação de uma nuvem híbrida a princípio para alcançar interesse tanto do usuário quanto do administrador (empresa ou equipe de TI). Pensando em utilizar a nuvem pública para dados com importância menor ou que não geram perdas significativas, enquanto a nuvem privada para dados relevantes, como dados pessoais.

Para Mell e Grace (2011)⁶, a utilização da nuvem híbrida se mostra mais vantajosa, visto que oferece a elasticidade rápida e o baixo custo da nuvem pública e a maior segurança e controle de recursos presente nas nuvens privadas.

3.3.2 APLICABILIDADE E BENEFÍCIOS DA CLOUD COMPUTING NO PROJETO

A aplicação da nuvem será de início, para a hospedagem da plataforma, com o objetivo de conseguir lidar com as altas demandas de tráfego, aproveitando a escalabilidade ofertada pela arquitetura.

Outro ponto de enorme importância para o projeto é a segurança dos dados de usuários, sendo assim tira-se proveito do fato de que provedores de computação em nuvem

⁵ Mell, P. e Grace, T. (2011). The NIST definition of cloud computing (draft). NIST special publication, 800(145):7.

⁶ Mell, P. e Grace, T. (2011). The NIST definition of cloud computing (draft). NIST special publication, 800(145):7.

possuem medidas avançadas de segurança, incluindo criptografia, autenticação de múltiplos fatores e monitoramento constante.

Por fim, a tomada de decisão dá-se à redução de custo na construção de espaços para infraestrutura física, para Velte et al (2010), a utilização da computação em nuvem tem como principal objetivo reduzir os custos operacionais, permitindo que o departamento de TI concentre seus esforços em atividades estratégicas, em vez de gerenciar um data center.

3.3.3 VANTAGENS DA CLOUD COMPUTING

A aplicação desse paradigma oferece inúmeras vantagens que podem funcionar de forma concreta a fim de atender às necessidades da empresa.

Uma das principais vantagens da computação em nuvem é a elasticidade dos recursos. A capacidade de dimensionar recursos de computação para cima ou para baixo, de acordo com as necessidades, oferece uma economia significativa. Sem a necessidade de manter uma infraestrutura ociosa durante os períodos de baixa demanda. Isso resulta em uma redução direta dos custos operacionais.

Outro benefício econômico de grande valia, é o modelo de pagamento conforme o uso. Ao eliminar a necessidade de grandes investimentos iniciais em infraestrutura, a empresa economiza capital que pode ser alocado para outras áreas críticas do negócio.

A alta disponibilidade e a segurança aprimorada fornecidas pelos provedores de serviços em nuvem têm impacto direto nos custos e na continuidade dos negócios. Os Acordos de Nível de Serviço (SLAs), contratos este que, segundo Sturm *et al* (2000)⁷, garantem que a empresa possa evitar perdas devido a interrupções, devido a descrição displicente dos produtos e os índices a serem atingidos.

3.3.4 DESENVOLVIMENTO EM CLOUD COMPUTING

A aplicação da nuvem neste projeto, consiste na implementação de um espaço virtualizado na nuvem e um banco de dados, ocorrerá de maneira a desenvolver um sistema de nuvem robusta, escalável e que possa ser adaptável a qualquer futura mudança.

Como falado anteriormente, a plataforma será hospedada em formato SaaS (*Software como um Serviço*), a fim de permitir o acesso local, nacional e internacional do sistema. A escalabilidade será progressiva, podendo ser expandida com o crescimento da empresa.

⁷ Sturm, R., Morris, W., Jander, M. (2000) "Foundations of Service Level Management". Estados Unidos: Sams. 288 p

Para ser mais preciso, no tópico seguir será descrito a escolha do sistemas em nuvem:

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

A decisão de optar pela AWS para o sistema foi impulsionada pelos resultados positivos obtidos em testes, aliados a uma análise criteriosa que revelou uma relação custo-benefício mais favorável, especialmente considerando que, inicialmente, a empresa prevê um alcance de usuários relativamente reduzido. A escolha estratégica da região South America (São Paulo) para a alocação na nuvem visa otimizar o desempenho e a latência.

Para virtualização dos sistemas, a plataforma adotará o Amazon EC2, oferecendo uma instância de carga de trabalho em um ambiente de locação compartilhada. Com um sistema operacional baseado em Linux, a escolha visa proporcionar eficiência e flexibilidade. O custo mensal associado a essa configuração específica do EC2 é estimado em 27,16 USD.

No que tange ao banco de dados, a preferência recai sobre o Amazon RDS, que será configurado com 20 GB de armazenamento SSD sob demanda. Essa escolha estratégica é fundamentada na confiabilidade e desempenho oferecidos pela solução. O custo mensal estimado para o Amazon RDS é de 155,69 USD.

Ao considerar o panorama geral, a análise de custos e benefícios destaca que a implementação na AWS apresenta uma vantagem econômica significativa. O custo inicial é nulo, proporcionando uma entrada financeira mais leve para a empresa, enquanto o custo mensal totaliza 182,85 USD. Essa decisão estratégica reflete a busca por eficiência operacional e otimização financeira.

Em comparação com outras opções de provedores de serviços em nuvem, como o Google Cloud, a AWS demonstrou ser a escolha mais vantajosa para atender aos requisitos específicos do projeto. A infraestrutura robusta, juntamente com os serviços escaláveis oferecidos pela AWS, posiciona a empresa de forma sólida para enfrentar os desafios futuros, garantindo um ambiente de nuvem confiável e eficiente.

3.3.6 DESENVOLVIMENTO EM CLOUD COMPUTING

O balanceamento de carga desempenha um papel crucial na otimização do desempenho e na eficiência dos serviços de computação em nuvem. Existem diversas abordagens, incluindo o balanceamento estático, que direciona os usuários para servidores específicos com base em seus endereços IP; o dinâmico, que considera critérios como carga do servidor, localização do usuário e tipo de solicitação; e o por demanda, que adiciona

servidores conforme necessário para atender à demanda. Essas estratégias garantem a distribuição equitativa de carga, melhorando a confiabilidade e a capacidade de resposta dos serviços na nuvem. A escolha do modelo de balanceador sendo de carga sob demanda, é por motivo do sistema possuir variações de usuários e dados, fazendo assim que o sistema possa se adaptar em momentos de pico.

A anatomia da computação compreende componentes essenciais, sendo a plataforma de infraestrutura responsável pelos recursos físicos, políticas e gerenciamento pela administração e segurança, e aplicações e serviços pela entrega de funcionalidades aos usuários. Esses componentes trabalham em conjunto para oferecer uma infraestrutura escalável e flexível. Toda essa estrutura será fornecida e de responsabilidade da AWS.

3.3.7 GOOGLE CLOUD ou AWS

A tecnologia do Cloud Computing vem revolucionando a forma como organizações gerenciam aplicativos, dados e recursos computacionais, este projeto propõe a implementação de soluções da AWS para melhorar a flexibilidade, otimizar operações, aumentar a segurança e reduzir os custos para a empresa parceira.

A decisão de implementar o projeto de Cloud Computing baseia-se em uma análise detalhada dos recursos oferecidos pela Amazon Web Services (AWS). Esta seção apresenta uma visão geral das funcionalidades da AWS e suas estratégias de gerenciamento, destacando como essas características específicas atendem aos requisitos do nosso projeto.

Ao integrar as funcionalidades e estratégias de gerenciamento da AWS na documentação, mostra confiança na abordagem robusta e eficaz para a implementação bem-sucedida do nosso projeto de Cloud Computing. Esta análise detalhada servirá como um guia fundamental para a equipe durante todo o processo de implementação, garantindo que seja aproveitado ao máximo os recursos oferecidos pela AWS em benefício do projeto e da organização como um todo.

3.4 ESTRUTURA DE DADOS

Será trabalhado o levantamento de requisitos que será utilizado para os projetos integrados. Serão trabalhadas também as estruturas de dados e como serão aplicadas ao Projeto, tendo em vista a linguagem utilizada.

3.4.1 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos é uma parte de suma importância no desenvolvimento de um software, é nesta ação que são definidas as funcionalidades do sistema.

A seguir, estão descritas os requisitos levantados:

3.4.1.1 REQUISITOS FUNCIONAIS

RF001	O sistema deve permitir que o usuário acesse através de login único tanto para o cliente como a empresa
FUNÇÃO	Acesso
DESCRIÇÃO	O sistema deve permitir que o usuário acesse tanto como cliente quanto como empresa usando um único login.
ENTRADAS	Credenciais de login (nome de usuário e senha)
ORIGEM	Usuário (cliente ou empresa)
SAÍDAS	Página inicial personalizada
DESTINO	Página inicial do usuário (cliente ou empresa)
AÇÃO	Autenticação e autorização
REQUER	Existência de contas de usuário para cliente e empresa
PRECONDIÇÃO	O usuário deve estar registrado como cliente ou empresa
PÓS-CONDIÇÃO	O usuário é redirecionado para a página inicial correspondente ao tipo de usuário após o login bem-sucedido
EFEITOS COLATERAIS	Nenhum

RF002	O sistema deva conter uma página de login para a empresa
FUNÇÃO	Página de Login Empresa

DESCRIÇÃO	O sistema deve conter uma página de login dedicada para as empresas acessarem suas contas.
ENTRADAS	Credenciais de login da empresa (nome de usuário e senha)
ORIGEM	Empresa
SAÍDAS	Página de painel de controle da empresa
DESTINO	Painel de controle da empresa
AÇÃO	Autenticação e autorização
REQUER	Existência de contas de empresas
PRECONDIÇÃO	Nenhum
PÓS-CONDIÇÃO	A empresa é redirecionada para o painel de controle após o login bem-sucedido
EFEITOS COLATERAIS	Caso haja falha na autenticação, a empresa não consegue acessar suas informações.

RF003	O sistema deve conter uma página inicial
FUNÇÃO	Página inicial
DESCRIÇÃO	Exibir uma página inicial ao usuário que o dará um norte
ENTRADAS	-
ORIGEM	Sistema
SAÍDAS	Página inicial
DESTINO	Interface do usuário
AÇÃO	Renderizar a página inicial

REQUER	Um index funcional
PRECONDIÇÃO	O sistema está ativo e funcional
PÓS-CONDIÇÃO	O usuário tem acesso a página
EFEITOS COLATERAIS	Nenhum

RF004	O sistema deve permitir que o usuário acesse o mapa
FUNÇÃO	Acesso ao Mapa
DESCRIÇÃO	O sistema deve permitir que o usuário acesse e visualize um mapa.
ENTRADAS	Nenhum
ORIGEM	Usuário
SAÍDAS	Visualização do mapa com informações geográficas
DESTINO	Página do mapa
AÇÃO	Visualização e interação com o mapa
REQUER	Nenhum
PRECONDIÇÃO	O usuário estar autenticado no sistema
PÓS-CONDIÇÃO	O usuário é capaz de visualizar informações geográficas no mapa
EFEITOS COLATERAIS	Nenhum

RF005	O sistema deve permitir que o usuário insira as informações de quantidade de reciclagem
FUNÇÃO	Inserção de Quantidade de Reciclagem
DESCRIÇÃO	O sistema deve permitir que o usuário insira as informações relacionadas à quantidade de reciclagem realizada.
ENTRADAS	Quantidade de reciclagem (valor numérico)
ORIGEM	Usuário
SAÍDAS	Confirmação de inserção bem-sucedida
DESTINO	Banco de Dados do sistema
AÇÃO	Inserção de dados
REQUER	Nenhum
PRECONDIÇÃO	O usuário estar autenticado no sistema
PÓS-CONDIÇÃO	As informações de quantidade de reciclagem são registradas no Banco de Dados
EFEITOS COLATERAIS	Caso haja erro na inserção dos dados, as informações de reciclagem podem não ser registradas corretamente no sistema.

RF006	O sistema deve permitir que o usuário insira o dia da entrega da reciclagem
FUNÇÃO	Inserir dia de entrega da reciclagem
DESCRIÇÃO	Permitir que o usuário insira no site as datas
ENTRADAS	Data da reciclagem inserida pelo usuário
ORIGEM	Usuário

SAÍDAS	Registro das datas
DESTINO	Banco de dados
AÇÃO	Inserir no banco as datas
REQUER	Que o usuário faça uma doação
PRECONDIÇÃO	O sistema e o banco estarem ativos e funcionais
PÓS-CONDIÇÃO	Ocorre o registro da data da reciclagem
EFEITOS COLATERAIS	Nenhum

RF007	O sistema deve permitir que o usuário insira metas para serem cumpridas
FUNÇÃO	Definição de Metas
DESCRIÇÃO	O sistema deve permitir que o usuário insira as metas a serem cumpridas, como objetivos de reciclagem ou redução de consumo.
ENTRADAS	Metas (descrição e valor numérico)
ORIGEM	Usuário
SAÍDAS	Confirmação de inserção bem-sucedida
DESTINO	Banco de Dados do sistema
AÇÃO	Inserção de metas
REQUER	Nenhum
PRECONDIÇÃO	O usuário estar autenticado no sistema
PÓS-CONDIÇÃO	As metas definidas pelo usuário são armazenadas no Banco de Dados

EFEITOS COLATERAIS	Caso haja erro na inserção das metas, as informações não serão registradas corretamente no sistema, afetando a definição de objetivos.
--------------------	--

RF008	O sistema deve permitir que o usuário tenha recompensas com as metas batidas
FUNÇÃO	Recompensas por Metas Cumpridas
DESCRIÇÃO	O sistema deve permitir que o usuário receba recompensas ao atingir as metas definidas, como prêmios ou pontos de recompensa.
ENTRADAS	Metas cumpridas, informações de recompensas (prêmios ou pontos)
ORIGEM	Sistema e usuário
SAÍDAS	Confirmação de recompensa bem-sucedida
DESTINO	Perfil do usuário e sistema de recompensas
AÇÃO	Atribuição de recompensas
REQUER	Nenhum
PRECONDIÇÃO	O usuário ter metas previamente definidas e cumpridas
PÓS-CONDIÇÃO	O usuário recebe as recompensas definidas após o cumprimento das metas
EFEITOS COLATERAIS	Caso as informações de metas ou recompensas sejam imprecisas, o sistema pode atribuir incorretamente as recompensas.

RF009	O sistema deve conter um mapa de atividades a serem cumpridas
FUNÇÃO	Mapeamento das atividades
DESCRIÇÃO	Exibir um mapa com missões que devem ser realizadas

ENTRADAS	-
ORIGEM	Usuário
SAÍDAS	Informações do mapa
DESTINO	Interface do usuário
AÇÃO	Mostrar o progresso nas missões no mapa
REQUER	Progresso do usuário
PRECONDIÇÃO	O sistema está ativo e funcional
PÓS-CONDIÇÃO	O usuário tem acesso ao mapa
EFEITOS COLATERAIS	Nenhum

RF0010	O sistema deve permitir que o usuário altere/redefina seus dados de acesso
FUNÇÃO	Alteração/Redefinição de Dados de Acesso
DESCRIÇÃO	O sistema deve permitir que o usuário altere ou redefina seus dados
ENTRADAS	Novos dados de acesso (nome de usuário e senha)
ORIGEM	Usuário
SAÍDAS	Confirmação de alteração bem-sucedida
DESTINO	Banco de Dados do sistema
AÇÃO	Atualização de dados de acesso
REQUER	Nenhum
PRECONDIÇÃO	O usuário estar autenticado no sistema

PÓS-CONDIÇÃO	Os novos dados de acesso são registrados no Banco de Dados
EFEITOS COLATERAIS	Caso os novos dados de acesso não sejam válidos, o usuário pode ter problemas ao tentar fazer login no futuro.

RF0011	O sistema deve permitir a comunicação entre o admin e o usuário
FUNÇÃO	Permitir a comunicação entre as partes
DESCRIÇÃO	Permitir a troca de mensagens entre os usuários e os admins
ENTRADAS	Mensagem enviada pelo remetente
ORIGEM	Usuário/admin
SAÍDAS	Mensagem recebida pelo destinatário
DESTINO	Usuário/admin
AÇÃO	Enviar, Receber, Armazenar mensagens entre as partes
REQUER	-
PRECONDIÇÃO	Que o sistema esteja funcional
PÓS-CONDIÇÃO	Mensagens são trocadas
EFEITOS COLATERAIS	Nenhum

RF0012	O sistema deve dar um pequeno entendimento para clientes de como usa-lo e progredir nele
FUNÇÃO	Orientação para Usuários
DESCRIÇÃO	O sistema deve fornecer uma orientação clara e concisa para os clientes entenderem como usar o sistema e fazer progressos nele.

ENTRADAS	Nenhum
ORIGEM	Sistema
SAÍDAS	Instruções ou guias de uso
DESTINO	Interface do Usuário
AÇÃO	Exibição de instruções ou guias de uso
REQUER	Interface intuitiva, documentação de uso
PRECONDIÇÃO	O usuário estar autenticado no sistema
PÓS-CONDIÇÃO	O usuário compreende o uso do sistema e pode progredir nele
EFEITOS COLATERAIS	Se as instruções não forem claras, os usuários podem ficar confusos e ter dificuldade em progredir.

RF0013	O sistema deve permitir que o usuário gere dados para relatórios
FUNÇÃO	Gerar relatórios
DESCRIÇÃO	Permitir que o usuário crie relatórios
ENTRADAS	Dados recolhidos
ORIGEM	Usuário
SAÍDAS	Informações geradas
DESTINO	Interface do usuário
AÇÃO	Configurar, gerar e exibir relatórios
REQUER	-
PRECONDIÇÃO	O usuário está autenticado e conectado.

PÓS-CONDIÇÃO	O usuário visualiza o relatório
EFEITOS COLATERAIS	Nenhum

RF0014	O sistema deve permitir que o admin cadastre quantas empresas forem necessárias
FUNÇÃO	Cadastro de Empresas
DESCRIÇÃO	O sistema deve permitir que o administrador cadastre quantas empresas forem necessárias para interagir com o sistema.
ENTRADAS	Dados da empresa (nome, informações de contato, etc.)
ORIGEM	Usuário
SAÍDAS	Confirmação de cadastro bem-sucedido
DESTINO	Banco de Dados do sistema
AÇÃO	Inserção de dados da empresa
REQUER	Nenhum
PRECONDIÇÃO	O administrador estar autenticado no sistema e ter permissões para cadastro de empresas
PÓS-CONDIÇÃO	Os dados da empresa são armazenados no Banco de Dados
EFEITOS COLATERAIS	Caso os dados da empresa sejam inseridos incorretamente, as informações podem estar imprecisas no sistema.

RF0015	O sistema deve permitir que o usuário cadastre quantas contas forem necessárias
FUNÇÃO	Diversos cadastros do mesmo usuários

DESCRIÇÃO	O sistema não interrompe uma pessoa de criar varias contas para a mesma, sendo possível ter vários usuários na plataforma
ENTRADAS	Dados do usuário
ORIGEM	Usuário
SAÍDAS	Varios acesso a plataforma
DESTINO	Plataforma
AÇÃO	Cadastrar diversos usuários
REQUER	Informações para cadastro
PRECONDIÇÃO	Acesso à plataforma
PÓS-CONDIÇÃO	Diversos usuários para acesso
EFEITOS COLATERAIS	-

RF0016	O sistema deve permitir que o usuário veja suas progressões
FUNÇÃO	Visualizar progresso
DESCRIÇÃO	Permitir que o usuário veja seu progresso
ENTRADAS	-
ORIGEM	Usuário
SAÍDAS	Progressão do usuário
DESTINO	Interface do usuário
AÇÃO	Apresentar o progresso do usuário
REQUER	Sistema funciona

PRECONDIÇÃO	O usuário está autenticado e conectado.
PÓS-CONDIÇÃO	O usuário visualiza suas progressões.
EFEITOS COLATERAIS	Nenhum

RF0017	O sistema deve permitir que o usuário veja os dados inseridos
FUNÇÃO	Visualização de Dados Inseridos
DESCRIÇÃO	O sistema deve permitir que o usuário veja os dados que foram inseridos no sistema previamente.
ENTRADAS	Nenhum
ORIGEM	Usuário
SAÍDAS	Dados inseridos previamente
DESTINO	Interface do Usuário
AÇÃO	Exibição de dados
REQUER	Nenhum
PRECONDIÇÃO	O usuário estar autenticado no sistema
PÓS-CONDIÇÃO	O usuário visualiza os dados inseridos anteriormente
EFEITOS COLATERAIS	Se os dados não estiverem corretamente organizados ou apresentados, o usuário pode ter dificuldade em interpretá-los corretamente.

RF0018	O sistema deve permitir que o usuário tenha acesso a novas inserções de novas Empresas
FUNÇÃO	Acesso a Novas Inserções de Empresas

DESCRIÇÃO	O sistema deve permitir que o usuário tenha acesso às novas inserções de empresas feitas no sistema.
ENTRADAS	Nenhum
ORIGEM	Sistema
SAÍDAS	Lista de novas inserções de empresas
DESTINO	Interface do Usuário
AÇÃO	Exibição da lista de novas inserções
REQUER	Nenhum
PRECONDIÇÃO	O usuário estar autenticado no sistema
PÓS-CONDIÇÃO	O usuário pode visualizar as novas empresas inseridas no sistema
EFEITOS COLATERAIS	Se a lista de novas inserções não for atualizada corretamente, o usuário pode não ver as empresas mais recentes.

RF0019	O sistema deve permitir que o usuário altere dados que o mesmo inseriu
FUNÇÃO	Alterar dados
DESCRIÇÃO	Permitir que o usuário possa alterar dados inseridos
ENTRADAS	Dados a serem alterados
ORIGEM	Usuário
SAÍDAS	Novos dados
DESTINO	Banco de dados
AÇÃO	Atualizar os dados cadastrados com alteração do usuário
REQUER	Dados para serem alterados

PRECONDIÇÃO	Usuário estar autenticado e conectado
PÓS-CONDIÇÃO	Dados alterados e atualizados
EFEITOS COLATERAIS	Nenhum

RF0020	O sistema deve permitir que o usuário possa “excluir” dados lançados
FUNÇÃO	Excluir dados que os usuário já tenha lançado
DESCRIÇÃO	Após o inserir alguns dados que possivelmente está errado ou incompleto, a plataforma permite que o usuário consiga excluir essa informação
ENTRADAS	Informações já lançadas
ORIGEM	Lançamentos
SAÍDAS	Exclusão do lançamento selecionado
DESTINO	Banco de dados
AÇÃO	Excluir dados
REQUER	Lançamentos cadastrados
PRECONDIÇÃO	-
PÓS-CONDIÇÃO	-
EFEITOS COLATERAIS	-

RF0021	O sistema deve permitir que o usuário cadastre seus dados para lançamento e não permita o lançamento sem que o mesmo tenha efetuado login
--------	---

FUNÇÃO	Cadastro e Lançamento de Dados com Autenticação
DESCRIÇÃO	O sistema deve permitir que o usuário cadastre seus dados para lançamento, garantindo que o lançamento só seja possível após o login.
ENTRADAS	Dados de lançamento
ORIGEM	Usuário
SAÍDAS	Confirmação de lançamento bem-sucedido
DESTINO	Banco de Dados do sistema
AÇÃO	Cadastro e lançamento de dados
REQUER	O usuário estar autenticado no sistema
PRECONDIÇÃO	O usuário estar autenticado no sistema
PÓS-CONDIÇÃO	Os dados de lançamento são armazenados no Banco de Dados
EFEITOS COLATERAIS	Caso o usuário não esteja autenticado, ele não pode lançar dados; sem autenticação, o usuário não terá acesso a suas informações de lançamento.

RF0022	O sistema deve possuir um avatar padrão ao se cadastrar no sistema
FUNÇÃO	Avatar Padrão no Cadastro
DESCRIÇÃO	O sistema deve atribuir um avatar padrão a um usuário no momento do cadastro no sistema.
ENTRADAS	Nenhum
ORIGEM	Sistema
SAÍDAS	Avatar padrão atribuído
DESTINO	Perfil do Usuário

AÇÃO	Atribuição de avatar
REQUER	Nenhum
PRECONDIÇÃO	Novo usuário se cadastrando no sistema
PÓS-CONDIÇÃO	Avatar padrão é associado à conta do usuário
EFEITOS COLATERAIS	Caso a atribuição do avatar padrão falhe, o usuário pode não ter um avatar associado a sua conta no cadastro.

RF0023	O Sistema deve possuir 9 avatares, baseados em cada integrante do grupo, conforme cumprir as missões desbloqueia os avatares
FUNÇÃO	Avatares Desbloqueáveis por Missões
DESCRIÇÃO	O sistema deve possuir 9 avatares, cada um baseado em um integrante do grupo. Os avatares são desbloqueados à medida que o usuário cumpre missões específicas.
ENTRADAS	Missões cumpridas, seleção de avatares
ORIGEM	Usuário
SAÍDAS	Avatares desbloqueados
DESTINO	Perfil do Usuário
AÇÃO	Desbloqueio de avatares
REQUER	Nenhum
PRECONDIÇÃO	O usuário estar autenticado e cumprir as missões necessárias
PÓS-CONDIÇÃO	O usuário obtém acesso a avatares específicos conforme cumpre as missões
EFEITOS COLATERAIS	Caso as missões não estejam bem definidas ou os avatares não sejam desbloqueados corretamente, o usuário pode não conseguir obter os

	avatares esperados.
--	---------------------

RF0024	O Sistema deve permitir que os usuários vejam seus progressos
FUNÇÃO	A plataforma mostra ao usuário seus progressos
DESCRIÇÃO	Atraves de uma barra e coonforme o usuario foi cumprindo seus objetivos e alcançando suas metas, está barra vai preenchendo e o usuário consegue acompanhar seu progresso mediante suas ações
ENTRADAS	Ações completas pelo usuário
ORIGEM	Usuário
SAÍDAS	Status do progresso
DESTINO	Plataforma
AÇÃO	Cumprir objetivos
REQUER	Progressão do usuário para cumprir os objetivos
PRECONDIÇÃO	Ter cadastro
PÓS-CONDIÇÃO	Acompanhar o progresso
EFEITOS COLATERAIS	Cumprir objetivo

RF0025	O Sistema deve permitir que o usuário tenha atualizações regulares do conteúdo
FUNÇÃO	Atualizar o conteúdo
DESCRIÇÃO	Permitir que o usuário tenha acesso a atualizações regulares do conteúdo

ENTRADAS	Informações a serem enviadas a todos usuários
ORIGEM	Administrador
SAÍDAS	Novas informações
DESTINO	Interface do Usuário
AÇÃO	Disponibilizar e exibir conteúdo atualizado.
REQUER	Nenhum
PRECONDIÇÃO	O sistema possui novas atualizações de conteúdo.
PÓS-CONDIÇÃO	O usuário visualiza o conteúdo atualizado.
EFEITOS COLATERAIS	Nenhum

RF0026	O Sistema deve permitir cooperação entre os usuários
FUNÇÃO	Cooperação entre Usuários
DESCRIÇÃO	O sistema deve permitir a cooperação e interação entre os usuários, facilitando a colaboração em tarefas ou projetos.
ENTRADAS	Comunicações, compartilhamento de recursos
ORIGEM	Usuários
SAÍDAS	Comunicações, compartilhamento de recursos
DESTINO	Outros usuários ou recursos compartilhados
AÇÃO	Troca de informações, colaboração
REQUER	Nenhum
PRECONDIÇÃO	Usuários estarem autenticados e autorizados para interagir

PÓS-CONDIÇÃO	Os usuários podem colaborar e compartilhar informações com outros usuários ou recursos, permitindo a cooperação
EFEITOS COLATERAIS	Se não houver mecanismos claros de comunicação ou compartilhamento, a cooperação entre os usuários pode ser dificultada.

RF0027	O Sistema deve permitir que o usuário tenha recompensas e conquistas em cada ação
FUNÇÃO	Recompensas e Conquistas
DESCRIÇÃO	O sistema deve permitir que o usuário receba recompensas e conquistas por suas ações e atividades realizadas no sistema.
ENTRADAS	Ações realizadas, progresso
ORIGEM	Usuário
SAÍDAS	Recompensas, conquistas desbloqueadas
DESTINO	Perfil do Usuário
AÇÃO	Atribuição de recompensas e conquistas
REQUER	Nenhum
PRECONDIÇÃO	O usuário estar autenticado e ter realizado ações que gerem recompensas ou conquistas
PÓS-CONDIÇÃO	O usuário recebe recompensas e desbloqueia conquistas conforme realiza ações
EFEITOS COLATERAIS	Caso as recompensas ou conquistas não sejam adequadamente atribuídas, o usuário pode não se sentir motivado a continuar usando o sistema de forma engajada.

RF0028	O Sistema deve permitir atualizações para manter o usuário engajados a
--------	--

	longo prazo
FUNÇÃO	Atualizações para Engajamento
DESCRIÇÃO	O sistema deve permitir atualizações contínuas para manter os usuários engajados a longo prazo, fornecendo novidades, recursos adicionais e melhorias.
ENTRADAS	Novos recursos, informações
ORIGEM	Desenvolvedores do Sistema
SAÍDAS	Novos recursos, informações atualizadas
DESTINO	Interface do Usuário
AÇÃO	Disponibilização de atualizações
REQUER	Nenhum
PRECONDIÇÃO	Nenhum
PÓS-CONDIÇÃO	Os usuários têm acesso a novidades e melhorias frequentes no sistema, o que mantém seu interesse a longo prazo
EFEITOS COLATERAIS	Caso as atualizações não sejam relevantes ou não sejam comunicadas adequadamente, os usuários podem perder o interesse no sistema.

RF0029	O sistema deve ser capaz de salvar o avanço de seus usuários
FUNÇÃO	O sistema deve permitir que o progresso individual de cada usuário seja salvo, permitindo que eles continuem de onde pararam em sessões futuras. Isso é especialmente relevante para aplicativos, jogos ou sistemas com fluxos de trabalho longos.
DESCRIÇÃO	Progresso do Usuário (por exemplo, marcos alcançados, etapas completas)
ENTRADAS	Progresso do Usuário (por exemplo, marcos alcançados, etapas

	completas)
ORIGEM	Usuários, Sistema
SAÍDAS	Dados de Progresso Salvo
DESTINO	Banco de Dados ou Sistema de Armazenamento
AÇÃO	Capturar e Armazenar o Progresso do Usuário
REQUER	Identificação do Usuário (login ou identificação única)
PRECONDIÇÃO	O sistema deve ter uma maneira de autenticar usuários.
PÓS-CONDIÇÃO	O progresso do usuário é registrado e salvo para uso futuro.
EFEITOS COLATERAIS	Usuários podem retomar suas atividades de onde pararam, independentemente do dispositivo ou sessão.

RF0030	O sistema deve ser capaz de enviar emails para seus usuários
FUNÇÃO	Envio de Emails para Usuários
DESCRIÇÃO	O sistema deve ser capaz de enviar emails aos usuários para fornecer informações, notificações, atualizações ou qualquer outra comunicação relevante. Isso é útil para manter os usuários informados sobre suas atividades ou mudanças no sistema.
ENTRADAS	Conteúdo do Email, Lista de Destinatários, Assunto, Anexos (opcional)
ORIGEM	Sistema, Administradores
SAÍDAS	Emails Enviados
DESTINO	Servidor de Email ou Serviço de Email em Nuvem
AÇÃO	Criar, Formatar e Enviar Emails
REQUER	Configurações de Email (servidor, credenciais)

PRECONDIÇÃO	Disponibilidade do Servidor de Email ou Serviço em Nuvem.
PÓS-CONDIÇÃO	Os emails são enviados com sucesso aos destinatários.
EFEITOS COLATERAIS	Possíveis atrasos na entrega dos emails devido a problemas de conectividade ou configuração inadequada. - Pode haver necessidade de monitoramento para evitar que os emails sejam marcados como spam. - Potencial aumento na comunicação eficaz com os usuários.

3.4.1.2 REQUISITOS NÃO FUNCIONAIS

RNF001	O sistema deve ser hospedado na nuvem
FUNÇÃO	Hospedagem na Nuvem
DESCRIÇÃO	Nenhum
ENTRADAS	Equipe de desenvolvimento.
ORIGEM	Ambiente de hospedagem na nuvem.
SAÍDAS	Ambiente de Hospedagem na Nuvem
DESTINO	Infraestrutura de Nuvem (por exemplo, Amazon Web Services, Microsoft Azure, Google Cloud Platform)
AÇÃO	Implantar e configurar o sistema em um ambiente de hospedagem na nuvem.
REQUER	Conexão com a Internet
PRECONDIÇÃO	Seleção de um provedor de serviços em nuvem.
PÓS-CONDIÇÃO	Implantação bem-sucedida do sistema na nuvem.
EFEITOS COLATERAIS	Acesso a recursos de escalabilidade e elasticidade, permitindo ajustes conforme o tráfego varia. - Potencial aumento de segurança e redundância, dependendo das configurações da nuvem. - Dependência contínua do provedor de serviços em nuvem para manter a disponibilidade e o desempenho.

RNF002	O sistema deve ter seu banco construído em SQL
FUNÇÃO	Armazenar dados dos usuários
DESCRIÇÃO	O sistema deve ser capaz de armazenar e exibir as informações dos usuários utilizando SQL

ENTRADAS	Dados
ORIGEM	Usuário
SAÍDAS	Dados tratados
DESTINO	Usuário
AÇÃO	O sistema deve armazenar os dados de seus usuários, onde os mesmos poderão visualizá-los na tela se assim for solicitado.
REQUER	Um banco de dados SQL e Scripts
PRECONDIÇÃO	O usuário deve preencher suas informações e salvá-las
PÓS-CONDIÇÃO	Os dados devem ser exibidos de forma harmônica aos usuários
EFEITOS COLATERAIS	Nenhum

RNF003	O sistema deve ser construído utilizando Node.js
FUNÇÃO	Alinhar a linguagem com as aprendidas em sala de aula
DESCRIÇÃO	O sistema será desenvolvido em Node.js, possuindo recursos já vistos e novos recursos
ENTRADAS	Linhas de código escritas em Node.js
ORIGEM	Desenvolvedores
SAÍDAS	Telas do sistema
DESTINO	Usuário
AÇÃO	O sistema deve ser capaz de responder rapidamente aos comandos do usuário, além de ser funcional, bem indentado e seguro
REQUER	Linhas de código desenvolvidas em Node.js

PRECONDIÇÃO	Obtenção de conhecimento aprofundado em Node.js
PÓS-CONDIÇÃO	Telas funcionais
EFEITOS COLATERAIS	Nenhum

RNF004	O sistema deve ter seu front-end criado em JavaScript e AngularJS
FUNÇÃO	Possuir animações mais práticas e fluídas
DESCRIÇÃO	O sistema deve usar JavaScript e Angular, pois podem ser mais rápidos
ENTRADAS	Linhas de código escritas em JavaScript e AngularJS
ORIGEM	Desenvolvedores
SAÍDAS	Telas do sistema
DESTINO	Usuário
AÇÃO	Tornar as telas mais interativas e agradáveis aos usuários
REQUER	Linhas de código escritas em JavaScript e AngularJS
PRECONDIÇÃO	Obtenção de conhecimento aprofundado em AngularJS
PÓS-CONDIÇÃO	Telas interativas e que prendam os usuários
EFEITOS COLATERAIS	Nenhum

RNF005	O sistema pode ser desenvolvido para mobile utilizando Flutter/Dart
FUNÇÃO	Desenvolvimento Mobile com Flutter/Dart

DESCRIÇÃO	O sistema pode ser implementado para dispositivos móveis (como smartphones e tablets) utilizando a estrutura de desenvolvimento Flutter e a linguagem de programação Dart. Isso permite uma abordagem de desenvolvimento multiplataforma eficaz para criar aplicativos móveis consistentes e de alta qualidade.
ENTRADAS	Design de Interface, Lógica de Negócios, Componentes Flutter
ORIGEM	Equipe de Desenvolvimento
SAÍDAS	Aplicativo Móvel Implementado
DESTINO	Dispositivos Móveis (iOS, Android)
AÇÃO	Desenvolver e Implementar o Aplicativo com Flutter/Dart
REQUER	Conhecimento em Flutter e Dart por parte da equipe de desenvolvimento.
PRECONDIÇÃO	Decisão de usar a tecnologia Flutter/Dart para o desenvolvimento mobile.
PÓS-CONDIÇÃO	Aplicativo móvel funcional implementado usando Flutter/Dart.
EFEITOS COLATERAIS	Maior eficiência de desenvolvimento ao criar um único código-base para múltiplas plataformas (iOS e Android). - Necessidade de treinamento ou contratação de desenvolvedores com conhecimento em Flutter/Dart. - Potencial otimização do desempenho e da experiência do usuário devido à natureza de alto desempenho do Flutter.

RNF006	O sistema deve possuir uma aplicação de Cross Site Scripting (XSS)
FUNÇÃO	Evitar possíveis malwares
DESCRIÇÃO	O sistema deve possuir métodos de proteção contra Cross Site Scripting
ENTRADAS	Códigos com medidas de segurança para a prevenção de ataques XSS

ORIGEM	Desenvolvedores
SAÍDAS	Bloqueios de caracteres e dados que podem causar danos aos usuários
DESTINO	Usuários perigosos
AÇÃO	O sistema deve bloquear malwares que são escritos em XSS que podem roubar dados e realizar fishing
REQUER	Conhecimento aprofundado em Cyber Segurança
PRECONDIÇÃO	Usuário tentando executar malwares
PÓS-CONDIÇÃO	Bloqueio do recebimento de dados daquele usuário
EFEITOS COLATERAIS	O usuário não conseguirá mais enviar dados

RNF007	O sistema deve ser desenvolvido em plataforma web
FUNÇÃO	Desenvolvimento em Plataforma Web
DESCRIÇÃO	O sistema deve ser projetado, desenvolvido e implantado como uma aplicação web, acessível por meio de navegadores web. Isso permite que os usuários acessem o sistema através da Internet, independentemente do dispositivo ou sistema operacional.
ENTRADAS	Requisitos de Negócios, Design de Interface, Lógica de Negócios
ORIGEM	Equipe de Desenvolvimento
SAÍDAS	Aplicação Web Funcional
DESTINO	Navegadores da Web
AÇÃO	Projetar, Desenvolver e Implantar uma Aplicação Web
REQUER	Conhecimento em tecnologias web (HTML, CSS, JavaScript, etc.).

PRECONDIÇÃO	Decisão de desenvolver o sistema como uma aplicação web.
PÓS-CONDIÇÃO	O sistema web é implantado e acessível através de navegadores
EFEITOS COLATERAIS	Acesso amplo a partir de diversos dispositivos e sistemas operacionais. - Necessidade de considerar a responsividade e a experiência do usuário em diferentes tamanhos de tela. - Requisitos de segurança adicionais para proteger contra ameaças da web.

RNF008	O sistema deve ser desenvolvido seguindo as leis gerais de proteção de dados (LGPD)
FUNÇÃO	Desenvolvimento do sistema de acordo com as regulamentações da LGPD.
DESCRIÇÃO	Garantir que o sistema atenda aos requisitos de proteção de dados da LGPD.
ENTRADAS	Regulamentações e diretrizes da LGPD.
ORIGEM	Legislação e regulamentações de proteção de dados.
SAÍDAS	Sistema desenvolvido em conformidade com a LGPD.
DESTINO	Sistema e seus usuários.
AÇÃO	Desenvolver o sistema de acordo com as diretrizes da LGPD.
REQUER	Conhecimento detalhado das regulamentações da LGPD, implementação de medidas técnicas e organizacionais de proteção de dados.
PRECONDIÇÃO	Equipe de desenvolvimento treinada sobre a LGPD, entendimento das regulamentações aplicáveis.
PÓS-CONDIÇÃO	Sistema implementado em conformidade com a LGPD, tratamento adequado dos dados pessoais.
EFEITOS	Possível aumento no tempo de desenvolvimento devido à

COLATERAIS	implementação de medidas de conformidade.
------------	---

RNF009	O sistema deve conter relatórios dos lançamentos
FUNÇÃO	Geração de relatórios dos lançamentos no sistema.
DESCRIÇÃO	Capacidade de criar relatórios detalhados sobre os lançamentos registrados.
ENTRADAS	Dados dos lançamentos (data, valor, categoria, etc.).
ORIGEM	Dados registrados no sistema.
SAÍDAS	Relatórios formatados dos lançamentos.
DESTINO	Usuários autorizados, impressão ou exportação.
AÇÃO	Gerar relatórios com base nos dados dos lançamentos.
REQUER	Estrutura de dados organizada para armazenar informações dos lançamentos, funcionalidade de filtragem e formatação de relatórios.
PRECONDIÇÃO	Lançamentos devem estar registrados no sistema.
PÓS-CONDIÇÃO	Relatórios precisos e atualizados disponíveis para os usuários.
EFEITOS COLATERAIS	Possível aumento na carga do sistema ao gerar relatórios complexos.

RNF0010	O sistema deverá conter o home atrativo e auto explicativo para atrair mais usuários
FUNÇÃO	Home Atrativo e Auto Explicativo
DESCRIÇÃO	O sistema deve apresentar uma página inicial (home) que seja visualmente atrativa e ofereça informações claras e auto explicativas sobre os recursos e benefícios do sistema. Isso visa atrair e engajar os

	usuários, incentivando-os a explorar mais o sistema.
ENTRADAS	Design da Página Inicial, Textos Explicativos, Mídia (imagens, vídeos)
ORIGEM	Equipe de Design e Desenvolvimento
SAÍDAS	Página Inicial Atrativa e Auto Explicativa
DESTINO	Navegadores da Web dos Usuários
AÇÃO	Projetar, Desenvolver e Implantar uma Página Inicial Atrativa
REQUER	Habilidades de Design e Desenvolvimento Web.
PRECONDIÇÃO	Compreensão das características e público-alvo do sistema
PÓS-CONDIÇÃO	A página inicial é implantada e acessível aos usuários
EFEITOS COLATERAIS	Maior probabilidade de retenção de usuários devido a uma primeira impressão positiva. - Maior engajamento inicial dos usuários com o sistema. - Possível aumento na taxa de conversão de visitantes em usuários registrados.

RNF0011	O sistema deve ser desenvolvido em interface intuitiva e fácil de se usar
FUNÇÃO	Interface Intuitiva e Usabilidade
DESCRIÇÃO	Garantir que o sistema seja desenvolvido com uma interface intuitiva e fácil de usar, proporcionando aos usuários uma experiência amigável e eficiente.
ENTRADAS	Nenhum
ORIGEM	Desenvolvedores de Interface
SAÍDAS	Interface do Usuário com elementos intuitivos, navegação clara e interações simples

DESTINO	Usuário
AÇÃO	Projeto e desenvolvimento da interface
REQUER	Design, Usabilidade
PRECONDIÇÃO	Nenhum
PÓS-CONDIÇÃO	Nenhum
EFEITOS COLATERAIS	Os usuários interagem facilmente com o sistema, conseguindo realizar tarefas de forma intuitiva e eficiente

RNF0012	O sistema deve ter aba saiba mais com links dos parceiros
FUNÇÃO	Aba "Saiba Mais" com Links de Parceiros
DESCRIÇÃO	O sistema deve ter uma aba "Saiba Mais" dedicada que contenha links para informações fornecidas pelos parceiros do sistema.
ENTRADAS	Nenhum
ORIGEM	Desenvolvedores de Interface
SAÍDAS	Aba "Saiba Mais" com links para informações dos parceiros
DESTINO	Interface do Usuário
AÇÃO	Inclusão da aba "Saiba Mais" e links de parceiros
REQUER	Nenhum
PRECONDIÇÃO	Nenhum
PÓS-CONDIÇÃO	Nenhum
EFEITOS COLATERAIS	Os usuários podem acessar informações detalhadas fornecidas pelos parceiros, aumentando o conhecimento e a utilidade do sistema

RNF0013	O sistema deve ser capaz de reconhecer senhas incorretas
FUNÇÃO	Reconhecimento de Senhas Incorretas
DESCRIÇÃO	O sistema deve ser capaz de identificar quando um usuário insere uma senha incorreta durante o processo de autenticação. Isso é fundamental para proteger a segurança do sistema e fornecer feedback aos usuários sobre as tentativas de login mal-sucedidas.
ENTRADAS	Senha Inserida pelo Usuário
ORIGEM	Usuário
SAÍDAS	Mensagem de Erro ou Alerta
DESTINO	Interface do Usuário (página de login)
AÇÃO	Verificar a Senha Inserida em relação à senha armazenada no sistema
REQUER	Armazenamento seguro de senhas (hashing, salting).
PRECONDIÇÃO	O sistema deve ter um processo de autenticação implementado.
PÓS-CONDIÇÃO	Feedback é fornecido ao usuário sobre a tentativa de login com senha incorreta.
EFEITOS COLATERAIS	Melhoria na segurança do sistema, protegendo contra tentativas de login não autorizadas. - Melhoria na experiência do usuário, fornecendo feedback imediato sobre erros de autenticação. - Possíveis bloqueios temporários de contas após várias tentativas de login incorretas.

RNF0014	O sistema deve ser capaz de ter uma interface responsiva, se adaptando automaticamente a diferentes tamanhos de tela referente a web
FUNÇÃO	Interface Responsiva
DESCRIÇÃO	O sistema deve ser desenvolvido com uma interface responsiva que se adapte automaticamente a diferentes tamanhos de tela, garantindo uma experiência de usuário otimizada independentemente do dispositivo

	utilizado para acessar o sistema. Isso abrange desde monitores de desktop até dispositivos móveis.
ENTRADAS	Tamanhos de Tela, Elementos de Interface
ORIGEM	Design de Interface, Equipe de Desenvolvimento
SAÍDAS	Interface Adaptada a Tamanhos de Tela
DESTINO	Diferentes Dispositivos (desktop, tablet, smartphone)
AÇÃO	Criar uma Interface Flexível e Adaptação de Elementos
REQUER	Habilidades de Design e Desenvolvimento Front-end.
PRECONDIÇÃO	Conhecimento dos tamanhos de tela alvo e das diretrizes de design responsivo.
PÓS-CONDIÇÃO	O sistema apresenta uma interface adaptada a diferentes tamanhos de tela.
EFEITOS COLATERAIS	Melhoria na experiência do usuário, independentemente do dispositivo utilizado. - Potencial aumento no engajamento do usuário devido à interface otimizada. - Maior satisfação dos usuários ao acessar o sistema de dispositivos variados.

RNF015	O sistema deve ter um código fonte bem documentado e seguir boas práticas de programação
FUNÇÃO	Código Fonte Documentado e Boas Práticas
DESCRIÇÃO	Assegurar que o sistema tenha um código fonte bem documentado e siga as boas práticas de programação, visando a legibilidade, manutenibilidade e qualidade do código.
ENTRADAS	Código fonte, documentação
ORIGEM	Desenvolvedores

SAÍDAS	Código fonte documentado e aderente às boas práticas
DESTINO	Repositório de Código
AÇÃO	Desenvolvimento, revisão e documentação
REQUER	Conhecimento de boas práticas de programação
PRECONDIÇÃO	Nenhum
PÓS-CONDIÇÃO	Nenhum
EFEITOS COLATERAIS	O código fonte é compreensível, organizado e aderente a padrões de qualidade, facilitando a manutenção e evolução do sistema

RNF016	O sistema deve ser capaz de lidar com erros de forma adequada, exibindo mensagens claras para os usuários
FUNÇÃO	Tratamento de erros e exibição de mensagens claras para os usuários.
DESCRIÇÃO	Assegurar que o sistema reaja apropriadamente a erros e forneça mensagens compreensíveis para os usuários.
ENTRADAS	Dados incorretos, solicitações inválidas, falhas inesperadas.
ORIGEM	Entradas incorretas ou eventos inesperados.
SAÍDAS	Mensagens de erro claras e informativas.
DESTINO	Usuários do sistema.
AÇÃO	Identificar erros, gerar mensagens de erro apropriadas e apresentá-las aos usuários.
REQUER	Mecanismo de detecção de erros.
PRECONDIÇÃO	O sistema deve estar em execução.
PÓS-CONDIÇÃO	Mensagens de erro compreensíveis são exibidas aos usuários,

	ajudando-os a entender e resolver problemas.
EFEITOS COLATERAIS	Possibilidade de aumento na complexidade do código para tratar diversos tipos de erros.

RNF0017	O sistema deve ser capaz de carregar páginas em X segundos para garantir uma experiência ágil para os usuários
FUNÇÃO	Tempo de carregamento ágil das páginas do sistema.
DESCRIÇÃO	Garantir que as páginas do sistema sejam carregadas dentro do limite de tempo especificado para proporcionar uma experiência de usuário rápida.
ENTRADAS	Solicitações de páginas pelo usuário.
ORIGEM	Usuário ou ações do sistema que resultam na solicitação de páginas.
SAÍDAS	Páginas carregadas dentro do limite de tempo.
DESTINO	Navegador ou aplicativo do usuário.
AÇÃO	Carregar as páginas do sistema de forma eficiente, otimizando o tempo de carregamento.
REQUER	Otimização do código e recursos do sistema para reduzir o tempo de carregamento.
PRECONDIÇÃO	Conexão estável com a internet, recursos disponíveis para carregamento da página.
PÓS-CONDIÇÃO	Páginas carregadas dentro do limite de tempo especificado, proporcionando uma experiência ágil ao usuário.
EFEITOS COLATERAIS	Possíveis concessões na qualidade de imagens ou recursos para atender ao limite de tempo.

RNF0018	O sistema deve ser capaz de ter proteção contra ataques
FUNÇÃO	Proteção contra Ataques
DESCRIÇÃO	O sistema deve ser capaz de ter proteção contra ataques, garantindo a segurança das informações e prevenindo vulnerabilidades.
ENTRADAS	Dados, interações
ORIGEM	Usuários, elementos externos
SAÍDAS	Interações seguras, dados não comprometidos
DESTINO	Sistema
AÇÃO	Implementação de medidas de segurança
REQUER	Conhecimento em segurança cibernética
PRECONDIÇÃO	Nenhum
PÓS-CONDIÇÃO	Nenhum
EFEITOS COLATERAIS	O sistema resiste a tentativas de invasão, mantendo a integridade dos dados e a funcionalidade

3.5 CONTEÚDO DA FORMAÇÃO PARA A VIDA: ENFRENTANDO ESTEREÓTIPOS

Para além do universo acadêmico, o estudo de conteúdo de formação para a vida se mostra com importância, a fim de aprimorar o desenvolvimento de cada estudante. Durante este semestre, o material estudado foi voltado para cidadania.

Nos tópicos a seguir, será relatado uma síntese do material estudado, referente ao subtema “Enfrentando Estereótipos”.

3.5.1 ENFRENTANDO ESTEREÓTIPOS

O mundo hoje é formado por padrões sociais que vão desde vestimentas, modo de comportamento, gírias e crenças. Porém, ao mesmo tempo que existe essa diversidade, nasce em conjunto um preconceito que precisa ser combatido.

3.5.1.1 Tema 1: Estereótipos e vida social

Este tópico explica como os estereótipos afetam as interações sociais. Um estereótipo é um preconceito ou generalização simplista sobre um grupo de pessoas com base em características como raça, sexo ou idade.

Eles podem influenciar a forma como as pessoas são tratadas e percebidas. Um exemplo é a crença generalizada de que todas as pessoas idosas são frágeis, o que pode levar à discriminação nos cuidados de saúde e nas oportunidades de emprego.

3.5.1.2 Tema 2: Estereótipos e representações

Neste tópico, é descrito como os estereótipos influenciam a representação de grupos na mídia e na cultura popular. Por exemplo, os estereótipos raciais podem levar a representações negativas de grupos raciais em filmes, programas de televisão ou anúncios, reforçando o preconceito e limitando a compreensão de diferentes culturas.

3.5.1.3 Tópico 3: Troco Likes - A Idealização da Vida na Internet

Este tópico se concentra na idealização da vida nas redes sociais, onde as pessoas muitas vezes compartilham apenas os momentos mais felizes e positivos, criando uma imagem irreal de suas vidas. Por exemplo, alguém pode postar constantemente fotos de viagens incríveis e momentos de diversão, deixando de fora os desafios e as dificuldades que

enfrentam. Isso pode levar a uma sensação de inadequação em quem compara sua própria vida com essas representações idealizadas.

3.5.1.4 Tópico 4: Convivendo com a Diferença

Neste tópico, é abordado a importância de aceitar e conviver com a diversidade de culturas, ideias e identidades. Exemplos práticos incluem promover a inclusão de pessoas com deficiências no local de trabalho, respeitar as diferentes orientações sexuais e identidades de gênero, e valorizar a diversidade étnica em comunidades e instituições. Isso contribui para uma sociedade mais justa e igualitária.

Por tanto, a compreensão dos estereótipos, sua influência na representação e interações sociais, a percepção distorcida da vida nas redes sociais e a promoção da convivência com a diferença são questões cruciais para uma sociedade mais inclusiva e justa. É essencial reconhecer como esses tópicos afetam nosso dia-a-dia e buscar maneiras de desafiar estereótipos e promover a diversidade.

3.5.2 ESTUDANTES NA PRÁTICA

Imagem 18: Explorando Estereótipos: Além do Rótulo “Redneck”



Fonte: Autoria própria

A equipe trabalhou para criar material provocativo e informativo que investiga as complexidades da identidade regional brasileira. O projeto visa desafiar estereótipos prejudiciais sobre a população rural, que muitas vezes é simplesmente rotulada de “caipiras”.

Cada região do Brasil tem seu próprio mundo, com cultura, tradições e realidades únicas. No entanto, essas nuances tendem a desaparecer quando vistas de fora. O estereótipo do “caipira” muitas vezes se baseia em generalizações muito distantes da realidade, principalmente quando relacionadas ao agronegócio.

A nossa abordagem explora esta dicotomia, destacando a diferença entre viver dentro e fora do campo. Ao discutir o termo “agrobóy” tentamos obter uma compreensão mais profunda das complexidades da identidade regional e desmistificar noções preconcebidas.

O banner criado busca incentivar a reflexão e a compreensão das questões culturais do Brasil. Cada integrante da equipe contribuiu com perspectivas. Espera-se que ele ressoe com as pessoas de diferentes comunidades, permitindo que se identifiquem com suas origens regionais de forma autêntica e se sintam valorizadas por sua diversidade cultural.

4. CONCLUSÃO

No Projeto Integrado (PI), foi possível destacar os principais pontos abordados ao longo da idealização da ideia da equipe. Inicialmente, foram examinadas algumas ideias para elaboração deste projeto, visando compreender o intuito real do projeto. Durante essa jornada, foram identificados e analisados aspectos cruciais.

Ao longo da execução, enfrentou-se algumas dificuldades, das quais se destacam desentendimentos entre alguns colegas, dificuldade em compreender algumas ideias, entre outras. No entanto, essas adversidades foram abordadas com diligência e resiliência, resultando em valiosas lições aprendidas e contribuindo para o desenvolvimento das habilidades necessárias para superar futuros desafios semelhantes.

É fundamental ressaltar que, ao longo do processo de pesquisa e elaboração deste PI, a equipe teve a constante busca por aprimoramento e aprofundamento nos temas tratados. As fontes utilizadas e os métodos empregados permitiram uma análise abrangente e embasada, contribuindo para a robustez das conclusões apresentadas.

Em suma, este projeto proporcionou uma visão aprofundada. As lições aprendidas, a superação de desafios e as descobertas realizadas ao longo desta jornada acadêmica contribuem significativamente para o enriquecimento do conhecimento e para o desenvolvimento de competências críticas para a formação profissional. Este trabalho representa não apenas o encerramento de uma etapa, mas também o ponto de partida para novas oportunidades de pesquisa e aprendizado.

REFERÊNCIAS

A ONU e o meio ambiente | As Nações Unidas no Brasil. Disponível em: <<https://brasil.un.org/pt-br/91223-onu-e-o-meio-ambiente>>.

ALSAQQA, S.; SAWALHA, S.; ABDEL-NABI, H. Agile Software Development: Methodologies and Trends. International Journal of Interactive Mobile Technologies, v. 14, n. 11, p. 246–270, 1 jul. 2020.

BCRYPT. Disponível em: <<https://www.npmjs.com/package/bcrypt>>.

CANADAY, R. H. et al. A back-end computer for database management. v. 17, n. 10, p. 575–582, 1 out. 1974.

CRYAH. O que é Low Poly? Disponível em: <<https://cryah.com.br/2015/04/27/o-que-e-low-poly/>>. Acesso em: 22 nov. 2023.

DATE, C. J. Introdução a sistemas de bancos de dados. 8. ed. Rio de Janeiro: Elsevier, 2003.

DURÃES, G. O que é React: o que é, como funciona, onde usar e o por que usar React! Disponível em: <<https://blog.tecnospeed.com.br/o-que-e-react/>>.

GÓMEZ GIL, C. Objetivos de Desarrollo Sostenible (ODS): una revisión crítica . , 2017. Acesso em: 19 out. 2023

Histórico ODS. Disponível em: <<https://www.gov.br/mma/pt-br/acao-a-informacao/informacoes-ambientais/historico-ods>>.

INFINITECH. Sustentabilidade. Google Forms. <https://forms.gle/iVtttScAwQvcTLrZ7>. Acessado em 21 de outubro de 2023.

JSONWEBTOKEN. Disponível em: <<https://www.npmjs.com/package/jsonwebtoken>>.

LIMA, Geycy Dyany Oliveira. ARQDEP: arquitetura de computação em nuvem com dependabilidade. 2014. 172 f. Dissertação (Mestrado em Ciências Exatas e da Terra) - Universidade Federal de Uberlândia, Uberlândia, 2014. DOI <https://doi.org/10.14393/ufu.di.2014.19>

MANTERO SUÁREZ, G. Sistema de recompensa del cerebro y neuronas del placer. idus.us.es, 1 jul. 2018.

OPENJS FOUNDATION. Express - Node.js web application framework. Disponível em: <<https://expressjs.com/>>.

Raw Queries | Sequelize. Disponível em: <<https://sequelize.org/docs/v6/core-concepts/raw-queries/>>.

REIS, K. et al. 5 Arquiteturas e linguagens. [s.l.: s.n.]. Disponível em: <<https://www.dpi.inpe.br/livros/bdados/cap5.pdf>>. Acesso em: 22 nov. 2023.

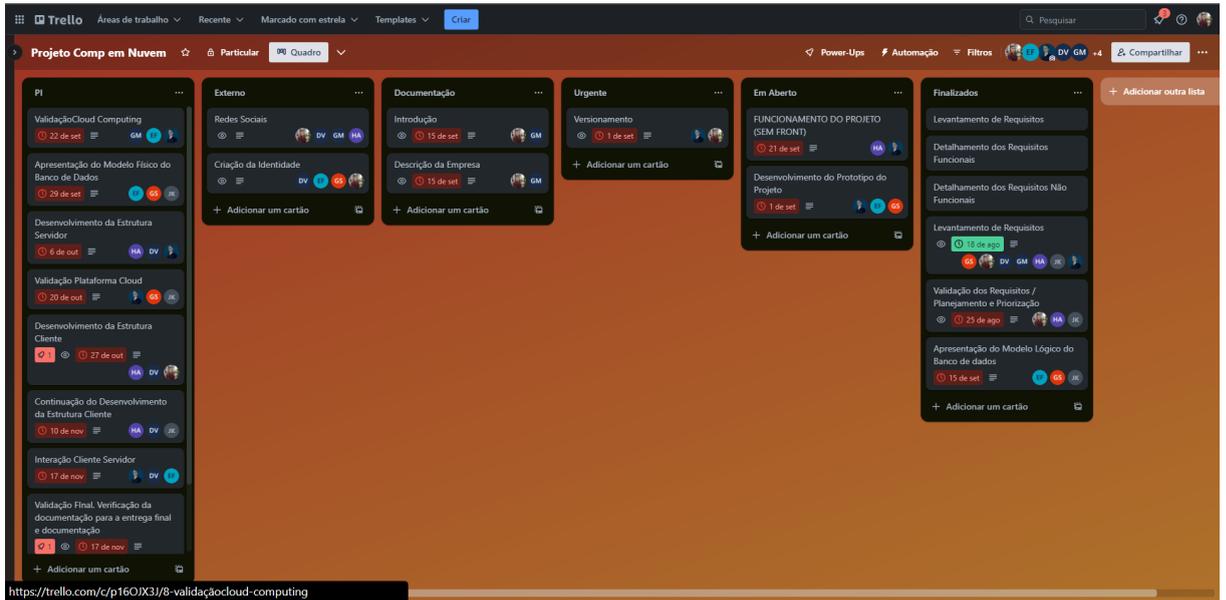
SOUSA, F. R. C. et al. Gerenciamento de Dados em Nuvem: Conceitos, Sistemas e Desafios. , dez. 2011. Acesso em: 3 nov. 2023

Trecho da Declaração da Conferência da ONU sobre o Meio Ambiente (Estocolmo, 1972), parágrafo 6. ONUBR Nações Unidas no Brasil. Disponível em: <https://nacoesunidas.org/acao/meio-ambiente/>. Acesso em 18 de outubro de 2023.

VELTE, Anthony T.; VELTE, Toby J.; ELSENPETER, Robert. Cloud Computing: Computação em Nuvem - Uma Abordagem Prática. Tradução de Gabriela Mei. Rio de Janeiro: Alta Books. 2010.

ANEXOS

Trello: Infinitech



Imagens de prototipação feita na plataforma Figma:

Gráfico da Plataforma



Tela Inicial



Tela de Cadastro

BUILDING MY WORLD

CADASTRO

DADOS PESSOAIS

Nome

Sobrenome

Data Nascimento

CPF

Google Facebook Apple

The registration screen features a white central card with rounded corners. At the top left of the card is the 'BUILDING MY WORLD' logo. The title 'CADASTRO' is prominently displayed in teal. Below it, the section 'DADOS PESSOAIS' contains four input fields: 'NOME', 'SOBRENOME', 'DATA NASCIMENTO', and 'CPF'. Each field has a small icon to its left. At the bottom of the card, there are three social media icons (Google, Facebook, Apple) and a navigation arrow.

Tela Login

BUILDING MY WORLD

LOGIN

USUÁRIO, CNPJ OU CPF

SENHA

ENTRAR

NÃO TEM UMA CONTA? CRIAR UMA

Google Facebook Apple

The login screen features a white central card with rounded corners. At the top left of the card is the 'BUILDING MY WORLD' logo. The title 'LOGIN' is prominently displayed in teal. Below it, there are two input fields: 'USUÁRIO, CNPJ OU CPF' and 'SENHA'. A teal 'ENTRAR' button is positioned below the fields. At the bottom of the card, there is a link 'NÃO TEM UMA CONTA? CRIAR UMA' and three social media icons (Google, Facebook, Apple).

Cadastro Empresa

BUILDING MY WORLD

BMW PARA EMPRESA

DADOS BASICOS

CNPJ

RAZÃO SOCIAL

Google Facebook Apple

The company registration screen features a white central card with rounded corners. At the top left of the card is the 'BUILDING MY WORLD' logo. The title 'BMW PARA EMPRESA' is prominently displayed in teal. Below it, the section 'DADOS BASICOS' contains two input fields: 'CNPJ' and 'RAZÃO SOCIAL'. At the bottom of the card, there are three social media icons (Google, Facebook, Apple) and a navigation arrow.

Tela Empresa

 BUILDING MY WORLD

DADOS EMPRESARIAIS

CENTRO UNIVERSITÁRIO FUNDAÇÃO DE ENSINO OCTÁVIO BASTOS - UNIFEOB

59.764.555/0001-52

(19)3634-3300 UNIFEOB@UNIFEOB.EDU.BR

ENDEREÇO

13.874-149 AV DOUTOR OCTÁVIO DA SILVA BASTOS, 2439

SÃO JOÃO DA BOA VISTA JARDIM NOVA SÃO JOÃO



Home

 BUILDING MY WORLD



VER MEU MUNDO

- 
- 
- 
- 
- 
- 
- 

Home Menu

 BUILDING MY WORLD

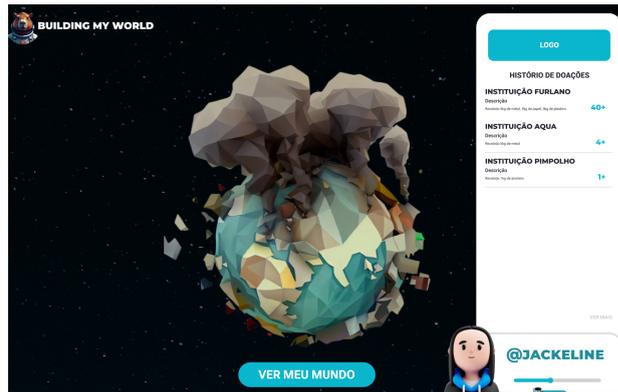


VER MEU MUNDO

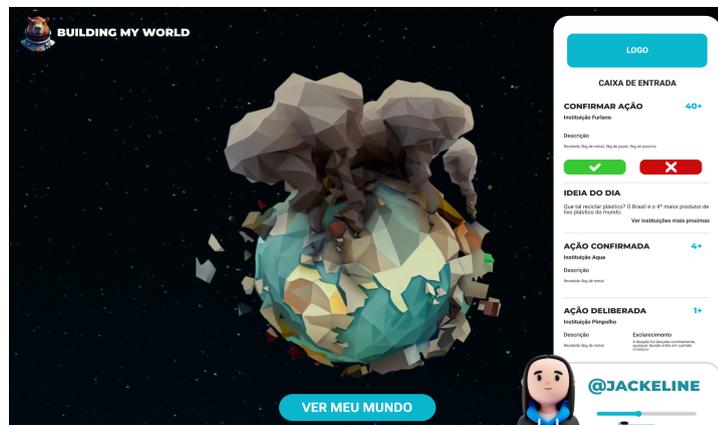
- LOGO
- PERFIL
- CAIXA DE ENTRADA 2
- HISTÓRICO DE AÇÕES
- SOM
- CONFIGURAÇÕES
- LOG OUT

 @JACKELINE

Home Histórico



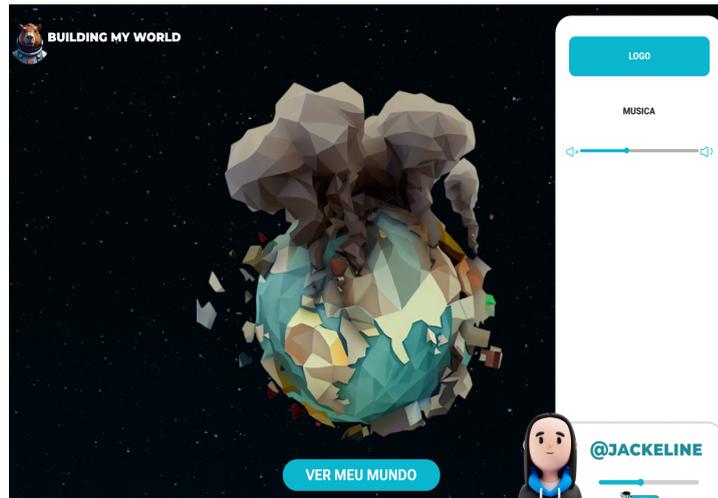
Home Menu Caixa de Entrada



Home Menu Contestar



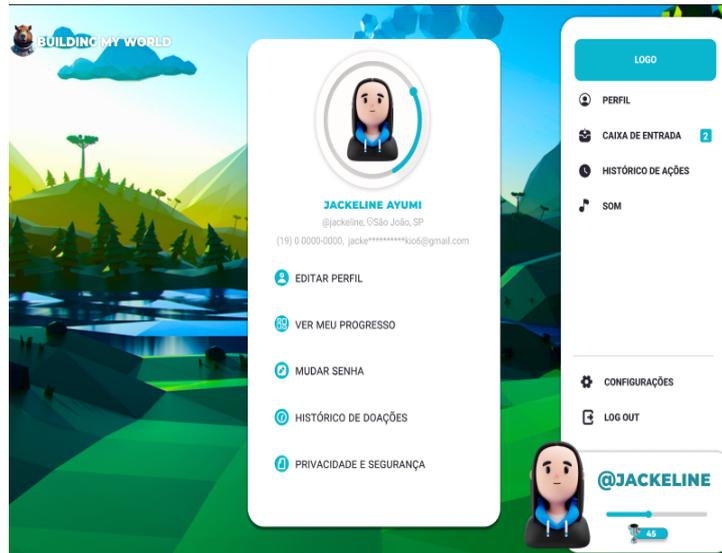
Home Menu Som



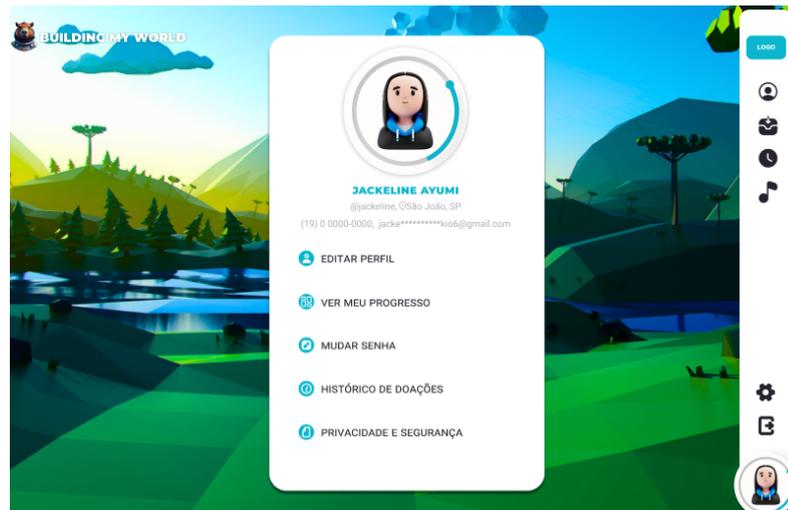
Perfil do usuário caixa de entrada



Perfil do usuário Open Menu



Perfil Usuário



Editar Perfil Usuário

BUILDING MY WORLD

LOGO

DADOS PESSOAIS

JACKELINE KANEKIO AYUMI

00/00/00 000.000.000-00

LOCALIZAÇÃO

00000-00 RUA A 56

CIDADE A BAIRRO A

jack*****9@GMAIL.COM

ENVIAR

Progresso Perfil Usuário

BUILDING MY WORLD

LOGO

45

3 DOAÇÕES VERIFIQUE TABELA DE PONTUAÇÃO

INSTITUIÇÃO	DATA	PONTOS
INSTITUIÇÃO FURLANO	24/09/2023	60+
INSTITUIÇÃO AGUA	28/09/2023	4+
INSTITUIÇÃO MARCOLHO	01/09/2023	1+

Editar Senha de Perfil e Usuário

BUILDING MY WORLD

LOGO

SENHA ATUAL

DIGITE SUA SENHA

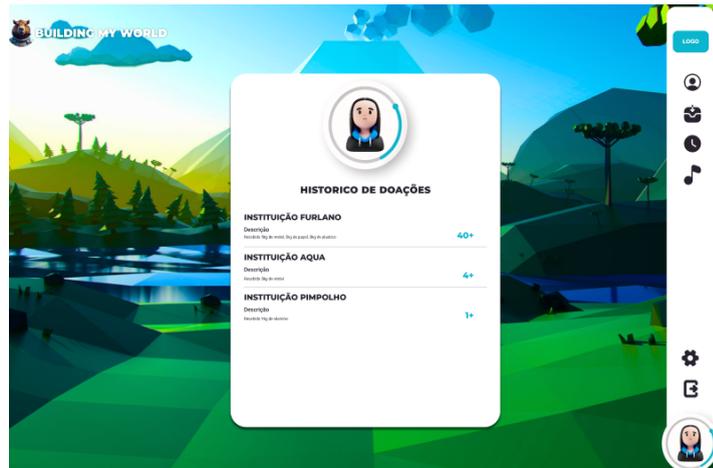
NOVA SENHA

DIGITE SUA NOVA SENHA

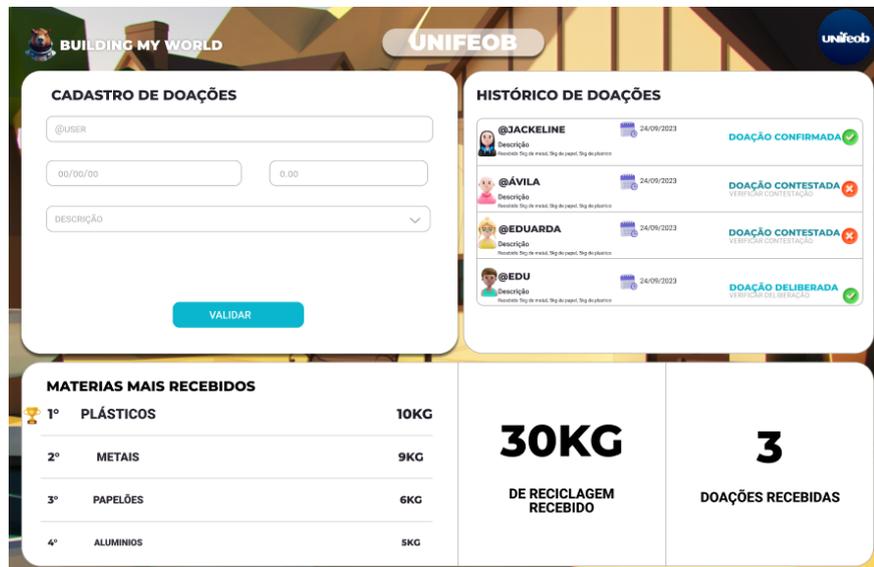
CONFIRME SUA NOVA SENHA

ENVIAR

Histórico Perfil e Usuário



Empresa



Termos de segurança

Termo de Compromisso, Segurança e Responsabilidade do Usuário

Segurança da Informação e Acesso

11 O Usuário deve adotar medidas razoáveis de segurança para proteger suas credenciais de acesso, senhas, informações pessoais e quaisquer outros dados sensíveis relacionados ao uso dos serviços fornecidos pelo Fornecedor.

12 É proibido ao Usuário divulgar, compartilhar, emprestar ou vender suas credenciais de acesso a terceiros sem a devida autorização expressa do Fornecedor.

13 O Usuário concorda em notificar imediatamente o Fornecedor sobre qualquer uso não autorizado de sua conta ou qualquer suspeita de comprometimento da segurança.

Uso Responsável e Ético

21 O Usuário concorda em utilizar os serviços fornecidos pelo Fornecedor de forma ética, legal e responsável, não se envolvendo em atividades fraudulentas, difamatórias, legais, ofensivas, prejudiciais, abusivas ou discriminatórias que possam afetar negativamente o Fornecedor, outros usuários ou terceiros.

22 O Usuário se compromete a cumprir todas as leis, regulamentos e diretrizes aplicáveis relacionadas ao uso dos serviços, incluindo, mas não se limitando a, leis de direitos autorais, propriedade intelectual, privacidade, segurança cibernética e proteção de dados.

Privacidade e Dados Pessoais

31 O Fornecedor compromete-se a proteger a privacidade do Usuário e a processar os dados pessoais de acordo com as leis de proteção de dados aplicáveis.

32 O Usuário concorda em fornecer informações verdadeiras, precisas e atualizadas ao Fornecedor, e compreende que o uso de informações falsas ou enganosas pode resultar na suspensão ou rescisão de sua conta.

33 O Usuário tem o direito de acessar, retificar ou excluir seus dados pessoais, conforme permitido pela lei aplicável, e pode entrar em contato com o Fornecedor para exercer esses direitos.

Atualização de Software e Segurança de Dispositivos

41 O Usuário é responsável por manter seus dispositivos e sistemas atualizados com as últimas atualizações de segurança e patches de software fornecidos pelo fabricante ou pelo Fornecedor, quando aplicável.

42 O Usuário reconhece que a falta de atualizações de segurança pode resultar em vulnerabilidades que podem comprometer a segurança de seus dados e do sistema em geral.

Este Termo de Compromisso, Segurança e Responsabilidade do Usuário ("Termo") estabelece as diretrizes e obrigações essenciais para garantir a segurança, a confidencialidade e a integridade de todas as partes envolvidas no uso de sistemas, serviços, informações, dispositivos ou recursos relacionados a qualquer plataforma, aplicativo, website ou sistema fornecido por qualquer entidade, doravante denominada "Fornecedor", e o usuário, doravante denominado "Usuário".

1. Comunicação e Notificações

2.51 O Usuário concorda em receber comunicações e notificações do Fornecedor relacionadas aos serviços, atualizações de segurança, promoções, mudanças nos termos de serviço e outras informações importantes, por meio dos canais de comunicação disponibilizados pelo Fornecedor.

3. Responsabilidade e Indenização

4.61 O Usuário reconhece que é o único responsável por qualquer uso indevido de sua conta e concorda em indenizar e isentar o Fornecedor de qualquer responsabilidade decorrente de atividades ilegais, fraudulentas ou não autorizadas em sua conta.

5.62 O Usuário concorda em indenizar o Fornecedor por quaisquer reclamações, perdas, danos, custos e despesas, incluindo honorários advocatícios razoáveis, decorrentes de sua violação deste Termo.

6. Termo de Rescisão e Encerramento da Conta

7.71 Este Termo permanecerá em vigor enquanto o Usuário utilizar os serviços do Fornecedor e poderá ser rescindido por qualquer das partes mediante notificação por escrito.

8.72 O Fornecedor reserva-se o direito de suspender ou encerrar a conta do Usuário a qualquer momento, com ou sem aviso prévio, se considerar que o Usuário violou este Termo ou de outra forma representou um risco para a segurança ou a integridade dos serviços.

9. Alterações no Termo

10.81 O Fornecedor reserva-se o direito de alterar este Termo a qualquer momento, notificando o Usuário por meio de comunicação adequada. O uso contínuo dos serviços após a notificação constituirá aceitação das alterações.

11. Lei Aplicável e Jurisdição

12.91 Este Termo será regido pelas leis do país ou jurisdição aplicável e quaisquer disputas relacionadas a este Termo serão resolvidas nos tribunais competentes dessa jurisdição.

13. Disposições Gerais

LOGO



Para visualizar o modelo:

<https://www.figma.com/file/zAgitpFcLzY1yQX1bWks8D?locale=en&type=design>

acesso em: 23 de outubro de 2023.

Curso de Análise e Desenvolvimento de Sistemas e Ciência da Computação

Módulo de Computação em Nuvem

Cronograma de Validação - Projeto Integrado

Unidade Estudo	Participação no Projeto	Validação
Tópicos Avançados de Banco de Dados	Modelo Lógico do Banco de Dados.	15/09/2023
	Modelagem física, estruturação e criação do banco de dados com técnicas e tópicos avançados.	29/09/2023
Linguagem e Técnicas de Programação	Protótipo do Sistema Web	16/08/2023
	Sistema Front-End integrado Back-End disponibilizado na web	08/11/2023
Estrutura de Dados	Levantamento de Requisitos	14/08/2023
	Validação dos requisitos	21/08/2023
Computação em Nuvem	Esta unidade é de extrema importância para o entendimento e a aplicação dos serviços em nuvem no sistema que será construído, mantendo-o com alta disponibilidade e escalabilidade. Os estudantes deverão realizar um estudo de viabilidade para implantação de aplicações em nuvem. Será inicialmente validado o escopo do projeto através do levantamento de requisitos realizado na unidade de Estrutura de Dados. Os estudantes deverão validar e apresentar uma proposta de crescimento de curto, médio e longo prazo a partir da arquitetura básica definida anteriormente. Como resultado final, os	

	estudantes deverão definir em equipe, cenários de implantação.	
Descrição do Projeto: desenvolver uma plataforma de gerenciamento de dados em nuvem que permita aos usuários armazenar, acessar e manipular informações de forma segura e eficiente. A plataforma será baseada em uma arquitetura de computação em nuvem e utilizará técnicas avançadas de banco de dados para garantir a escalabilidade e a disponibilidade dos dados.		