



Curso Superior de Desenvolvimento de Software Multiplataforma

Davy Oliveira Ribeiro – 3011392413004

Matheus Augusto Santos Gueff – 3011392413006

Matheus Nery de Camargo – 3011392413002

Pedro Silva Martins – 3011392413013

Ryan Carlo Negretti Pereira – 3011392413028

Disciplinas

Experiência do Usuário

Integração e Entrega Contínua

Laboratório de Desenvolvimento Web

SaneaSP

Orientadores:

Prof. Felipe Santos Medeiros

Prof.^a Maria Janaina da Silva Ferreira

Prof. Tiago Vanderlei de Arruda

Votorantim
Novembro de 2025.

RESUMO

O projeto SaneaSP tem como objetivo criar um site para fortalecer a comunicação entre a população, a prefeitura e os responsáveis pela área de saneamento em sua cidade, contribuindo para duas ODS: Água Potável e Saneamento (ODS6) e Saúde e Bem-Estar (ODS3). O projeto permitirá que os cidadãos relatem problemas de forma estruturada e fácil e acompanhem as soluções para esses problemas, além de disponibilizar um dashboard para controle e geração de relatórios para a prefeitura. Dessa forma, buscamos contribuir para a eficiência e transparência das iniciativas já existentes.

Sumário

1. DESCRIÇÃO DO PROJETO	4
1.1. Proposta do Software.....	4
1.2. Justificativa	4
1.3. Mapa do Site	5
1.4. Logomarca.....	6
2. REQUISITOS DO PROJETO.....	6
2.1. Levantamento de Requisitos	6
2.2. Requisitos Funcionais	7
2.3. Diagrama de Caso de Uso.....	8
2.4. Descrição do Caso de Uso	9
2.5. Requisitos Não Funcionais	15
3. PROJETO DO SOFTWARE	16
3.1. Tecnologias Utilizadas	16
3.2. Modelo de Dados	17
3.2.1 Modelo Conceitual.....	17
3.2.2 Modelo Lógico	18
3.3. Diagrama de Classes.....	19
3.4. Diagrama de Atividades.....	20
3.5. Diagrama de Sequência.....	22
3.6. Interfaces com o Usuário	22
3.7. Metodologia Utilizada.....	39
4. ESTRATÉGIA DE TESTES	39
5. IMPLANTAÇÃO	40

LISTA DE FIGURAS

Figura 1 - Mapa mental do Site	2
Figura 2 - Logomarca do projeto	3
Figura 3 - Caso de Uso	5
Figura 4 - Diagrama de Entidade Relacionamento (DER).....	14
Figura 5 - Modelo Lógico	15
Figura 6 - Diagrama de Classes	16
Figura 7 - Diagrama de Atividade "Adicionar Reclamação"	17
Figura 8 - Diagrama de Atividades "Comentários".....	18
Figura 9 - Diagrama de Sequência "Reclamação"	18
Figura 10 – Cabeçalho e barra de navegação	19
Figura 11 – Rodapé do site.....	19
Figura 12 – Página Inicial dos Usuários	19
Figura 13 – Continuação (Página Inicial dos Usuários).....	20
Figura 14 – Página de Login.....	20
Figura 15 – Página de Cadastro.....	21
Figura 16 – Continuação (Página de Cadastro).....	21
Figura 17 - Aviso de Cadastro realizado com sucesso.....	22
Figura 18 - Aviso de login realizado com sucesso	22
Figura 19 – Barra de navegação após o login	22
Figura 20 - Barra de navegação após o login	23
Figura 21 – Editar Perfil	23
Figura 22 – Continuação (Editar Perfil)	23
Figura 23 – Página Inicial do Administrador.....	24
Figura 24 – Página de Doenças	24
Figura 25 – Continuação (Página de Doenças)	25

Figura 26 – Card para nenhum resultado encontrado	25
Figura 27 – Página de Descrição da Doença	25
Figura 28 – Continuação (Página da Descrição da Doença)	26
Figura 29 – Página de Notícias	26
Figura 30 - Continuação (Página de Notícias)	27
Figura 31 – Página de Reclamações	27
Figura 32 – Continuação (Página de Reclamações)	28
Figura 33 – Continuação (Página de Reclamações)	28
Figura 34 – Página de Descrição da Reclamação	29
Figura 35 – Página de Comentários	29
Figura 36 - Continuação (Página de Comentários)	30
Figura 37 - Formulário de Cadastro de Reclamação	30
Figura 38 - Continuação (Formulário de Cadastro de Reclamação)	31
Figura 39 - Pré Visualização de Imagem	31
Figura 40 - Formulário de Cadastro de Notícia	32
Figura 41 - Continuação (Formulário de Cadastro de Notícias)	32
Figura 42 - Inserção de tag e fonte	32
Figura 43 - Formulário de Cadastro de Doenças	33
Figura 44 – Continuação (Formulário de Cadastro de Doenças)	33
Figura 45 - Continuação (Formulário de Cadastro de Doenças)	33
Figura 46 – Página Sobre Nós	34
Figura 47 - Continuação (Página Sobre Nós)	34
Figura 48 – Página dos Responsáveis	34
Figura 49 – Continuação (Página dos Responsáveis)	35

LISTA DE QUADROS

Quadro 1 - Requisitos funcionais.....	4
Quadro 2 - Caso de uso "Cadastro"	5
Quadro 3 - Caso de uso "Login"	6
Quadro 4 - Caso de uso "Login com redes sociais"	7
Quadro 5 - Caso de uso "Reclamação"	7
Quadro 6 - Caso de uso "Comentário"	8
Quadro 7 - Caso de uso "Gerar log"	9
Quadro 8 - Caso de uso "Informações do site"	9
Quadro 9 - Caso de Uso "Administrar Doenças"	10
Quadro 10 - Caso de uso "Administrar Notícias"	10
Quadro 11 - Caso de uso "Configurar perfil"	11
Quadro 12 - Requisitos não funcionais	12

1. DESCRIÇÃO DO PROJETO

1.1. Proposta do Software

O presente projeto tem como objetivo abranger duas ¹ODSs: Água Potável e Saneamento (ODS6) e Saúde e Bem-Estar (ODS3), por meio de um website. A função do site SaneaSP é auxiliar a prefeitura ou o órgão responsável pelo saneamento básico e administração da água da cidade, disponibilizando de forma rápida e organizada uma série de comentários e denúncias de problemas que estão ocorrendo na região, servindo como um centro de informações para os responsáveis.

A plataforma também contará com um dashboard para controle e geração de relatórios para os administradores, prefeitura e órgãos responsáveis pelo saneamento.

Dessa forma, o projeto prevê reduzir a invisibilidade dos problemas de saneamento na sua cidade, contribuindo tanto para os responsáveis quanto para os cidadãos do local, tendo um layout agradável, simples e objetivo, acessibilidade para libras e futuramente daltonismo.

1.2. Justificativa

Foram escolhidas as ODSs seis e três (Água Potável e Saneamento e Saúde e Bem-Estar, respectivamente), pois operam uma relação de causa e consequência entre si, já que populações com saneamento básico e água potável estão protegidas de uma ampla gama de doenças.

Todavia, problemas de saneamento e esgoto, como as fossas a céu aberto e canos de água estourados, não recebem a devida atenção dos órgãos responsáveis. Mesmo com a infraestrutura para resolver tais problemas, muitas vezes não há a percepção de que esses problemas sequer estão acontecendo.

¹ ODS

Além disso, a desinformação da população em relação às consequências da falta de saneamento básico, como as doenças que ele dissemina, dificulta a comunicação entre a governança e a população, pois ambos não sabem que existe um problema a ser resolvido.

Somado a isso, a má divulgação dificulta que a população saiba se a prefeitura está se mobilizando para resolver estes problemas, o que a desestimula a denunciá-los.

Portanto, haja vista esses obstáculos, foi analisada uma chance de colaborar para a redução desses problemas, por meio de um site focado justamente em tornar a manutenção do saneamento básico e as atitudes da governança mais visíveis e mostrar os problemas do cotidiano, informar, e divulgar notícias para a população.

1.3. Mapa do Site

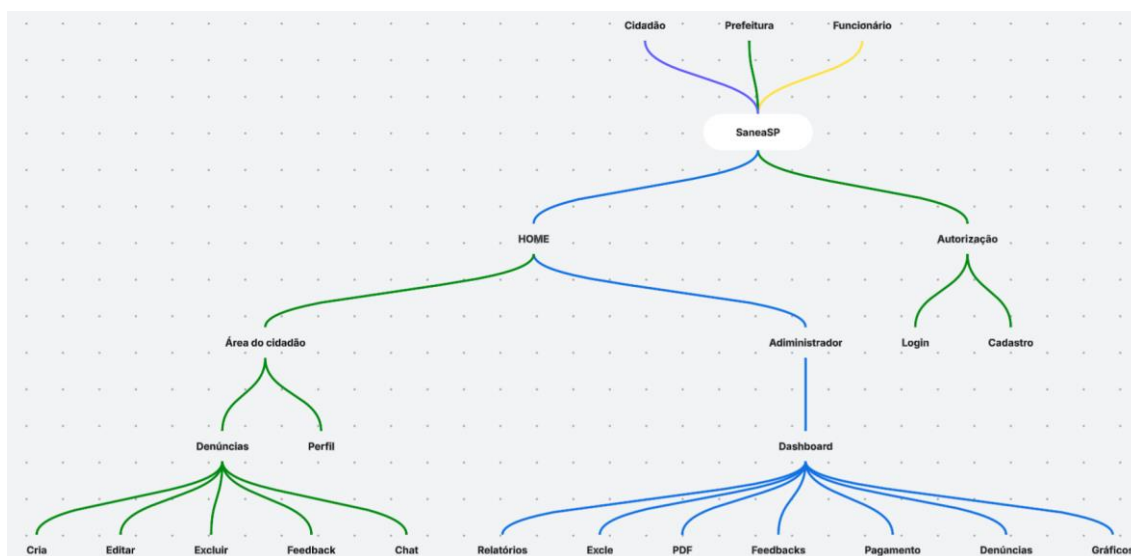


Figura 1 - Mapa mental do site

1.4. Logomarca



Figura 2 - Logomarca do projeto

2. REQUISITOS DO PROJETO

2.1. Levantamento de Requisitos

Para o levantamento de requisitos, foi observado como é o cotidiano do saneamento básico em Votorantim, que é a cidade-base do projeto, onde foi observada a necessidade da criação de um site que reúna informações e que torne a comunicação da população mais fácil.

Uma das ferramentas que foi utilizada como base é o site Reclame Aqui, que dispõe de uma comunicação entre o usuário e o serviço que ele avalia, assim como o site deste projeto, que busca criar essa facilidade para os responsáveis pelo saneamento da cidade, com o sistema de comentários e respostas.

2.2. Requisitos Funcionais

Quadro 1 - Requisitos funcionais

Nº Requisito	Nome	Descrição
RF001	Login e autenticação.	O usuário pode se cadastrar ou fazer o login para poder acessar o sistema. Pode ser feito pelo serviço do projeto ou a partir das redes sociais.
RF002	Gerenciamento e Cadastro de Reclamações.	O usuário deve ser capaz de criar reclamações, editar e remover, e a reclamação deve ser pública para visualização por outros usuários ou administradores.
RF003	Criação de comentários e chat de respostas.	Um administrador deve ser capaz de responder à reclamação do usuário e criar um chat, onde o usuário criador da reclamação pode se

		comunicar com quem o responder.
RF004	Gerenciamento e Cadastro de Doenças.	Um administrador deve ser capaz de criar, editar e excluir uma doença, e essa doença deve ser visualizada por todos os usuários.
RF005	Gerenciamento e Cadastro de Notícias.	Um administrador deve ser capaz de criar, editar e excluir uma notícia, e essa notícia deve ser visualizada por todos os usuários.
RF006	Filtragem de Reclamações.	Os dados da tabela de reclamações devem ser filtrados com base em pontuação, categorias e geograficamente.
RF007	Upload de Imagens	O sistema deve fazer upload de imagens enviadas relacionadas aos registros para exibição no site.
RF008	Logging de Comentários e Reclamações	As ações da tabela de Comentários e Reclamações devem ser registradas em log.
RF009	Geração de Pontuação para Reclamações.	As reclamações devem ser classificadas por pontuação para melhorar a confiabilidade. Realizada pelo nível de detalhamento da reclamação e do perfil do usuário criado.

2.3. Diagrama de Caso de Uso

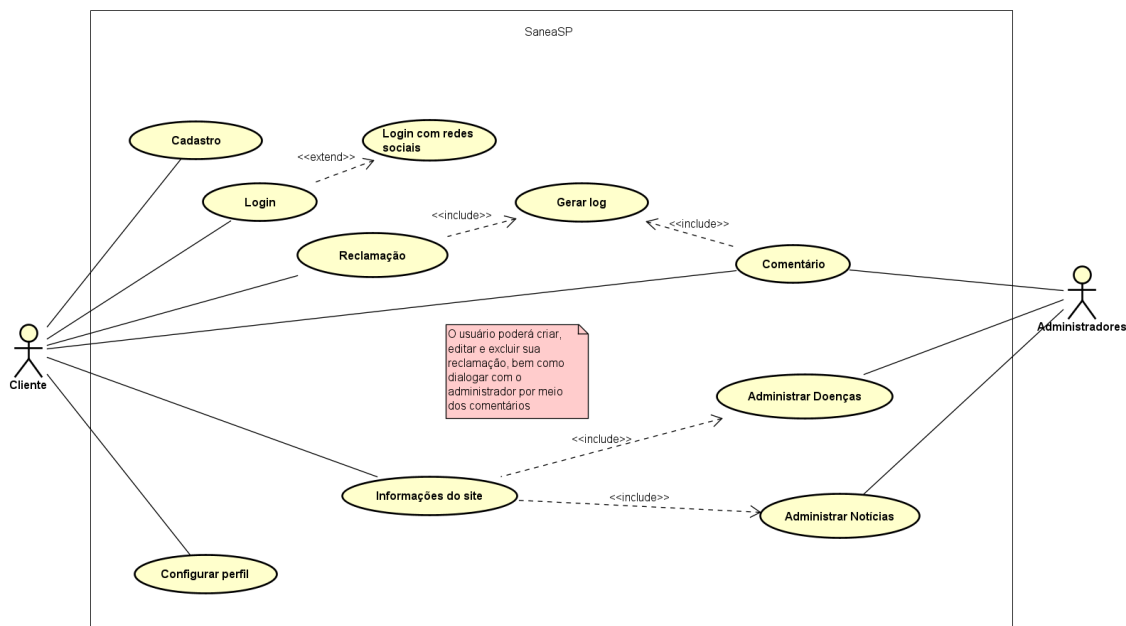


Figura 3 - Caso de Uso

2.4. Descrição do Caso de Uso

Quadro 2 - Caso de uso "Cadastro"

Caso de uso	Cadastrar Usuário.	
Ator Principal	Usuário.	
Ator Secundário	-	
Pré-condições	Usuário não possui login ou quer criar uma conta	
Pós-condições	O usuário possui seus dados cadastrados no sistema, possibilitando efetuar seu login.	
Fluxo Principal		
Ações do Ator		Ações do Sistema
CADASTRAR		
1. Acessar página de cadastro.		
		2. Sistema exibe o formulário.
3. Usuário preenche o formulário com seus dados (nome, e-mail, senha, telefone, CPF e endereço).		
		4. Sistema valida se campos obrigatórios estão preenchidos, se o e-mail já existe no sistema e se é um e-mail válido, se o CEP

	é válido, e se as senhas coincidem na confirmação da senha.
	5. Sistema guarda as informações no banco de dados.
	6. Retorna uma mensagem de cadastro realizado com sucesso.
	7. O usuário é redirecionado para a página de login.

Quadro 3 - Caso de uso "Login"

Caso de uso	Login de Usuário	
Ator Principal	Usuário.	
Ator Secundário	-	
Pré-condições	Usuário possui seus dados já cadastrados no sistema.	
Pós-condições	O usuário pode realizar reclamações no site.	
Fluxo Principal		
Ações do Ator		Ações do Sistema
LOGIN		
1. Acessar página de login.		
		2. Sistema exibe o formulário.
3. Usuário preenche o formulário com seus dados (e-mail e senha).		
		4. Sistema valida se todos os campos estão preenchidos, se o e-mail existe no sistema e se a senha é válida para esse usuário.
		5. Retorna uma mensagem se o login foi efetuado com sucesso.
		6. O usuário é redirecionado para a página principal do site.

Quadro 4 - Caso de uso "Login com redes sociais"

Caso de uso	Login com redes sociais.
Ator Principal	Usuário.
Ator Secundário	-
Pré-condições	O usuário deve possuir um cadastro em alguma rede

	social.
Pós-condições	O usuário pode realizar reclamações no site.
Fluxo Principal	
Ações do Ator	Ações do Sistema
LOGIN COM REDES SOCIAIS	
1. Acessar página de login.	
	2. Sistema exibe o formulário.
3. Usuário escolhe a rede social para efetuar login.	
	4. Sistema redireciona o usuário para a página de login da rede social.
	5. Retorna uma mensagem se o login foi efetuado com sucesso.
	6. Redireciona o usuário para a página principal.

Quadro 5 - Caso de uso "Reclamação"

Caso de uso	Reclamações.
Ator Principal	Usuário.
Ator Secundário	-
Pré-condições	O usuário deve estar com seu login.
Pós-condições	-
Fluxo Principal	
Ações do Ator	Ações do Sistema
RECLAMAÇÃO	
1. Acessar página de adicionar reclamações.	
	2. Sistema exibe o formulário.
3. O usuário preenche os dados do formulário com os campos necessários.	
	4. Sistema valida se os campos obrigatórios foram preenchidos e se há um endereço válido.
	5. Sistema valida se o usuário enviou uma imagem, e se houver, valida o tamanho e tipo do arquivo enviado.

	6. O sistema gera pontuação baseada na informação fornecida e na confiabilidade da reclamação.
	7. Sistema guarda as informações no banco de dados.
	8. Retorna uma mensagem se a reclamação foi cadastrada com sucesso.
	9. Redireciona o usuário para a página de reclamações.

Quadro 6 - Caso de uso "Comentário"

Caso de uso	Comentário.
Ator Principal	Usuário.
Ator Secundário	-
Pré-condições	Necessita de uma reclamação.
Pós-condições	-
Fluxo Principal	
Ações do Ator	Ações do Sistema
RESPOSTA	
1. Acessar página de comentários.	
	2. Sistema exibe o histórico das mensagens e uma caixa de texto para digitar seu comentário.
3. O usuário preenche a caixa de texto com seu comentário.	
	4. Sistema valida se o usuário preencheu a caixa de texto e se foram digitadas palavras ofensivas.
	5. Sistema grava as informações no banco de dados.
	6. Sistema atualiza a página, com os dados do comentário do usuário preenchidos.

Quadro 7 - Caso de uso "Gerar log"

Caso de uso	Gerar log.
Ator Principal	Usuário.
Ator Secundário	-
Pré-condições	Reclamação ou comentário a ser editado.
Pós-condições	-
Fluxo Principal	
Ações do Ator	Ações do Sistema
GERAR LOG	
1. O ator edita um comentário ou uma reclamação criada.	
	2. O sistema utiliza um trigger para registrar o conteúdo antigo e o novo em linhas da tabela de logs.

Quadro 8 - Caso de uso "Informações do site"

Caso de uso	Informações do site.
Ator Principal	Usuário.
Ator Secundário	-
Pré-condições	Acessar a página de "Sobre nós"
Pós-condições	
Fluxo Principal	
Ações do Ator	Ações do Sistema
INFORMAÇÕES DO SITE	
1. Acessar páginas informativas.	
	2. Sistema exibe as informações em ordem de atualidade.
3. O usuário pesquisa por tags para filtragem das informações ou por nome.	
	4. O sistema valida se existem informações com a tag especificada ou se existe alguma informação com o nome especificado.

Quadro 9 - Caso de Uso "Administrar Doenças"

Caso de uso	Administrar doenças.
Ator Principal	Administrador.

Ator Secundário	-
Pré-condições	Possuir login de administrador.
Pós-condições	-
Fluxo Principal	
Ações do Ator	Ações do Sistema
ADMINISTRAR DOENÇAS	
1. Acessar página de cadastro de informações.	
	2. Sistema exibe o formulário para o cadastro da informação.
3. O administrador preenche os campos do formulário.	
	4. O sistema valida se os dados das informações são válidos e se não são duplicados. Se houver imagem, válida se o tamanho e tipo são válidos.
	5. Sistema guarda os dados no banco de dados.
	6. Sistema informa se foi efetuado com sucesso e retorna uma mensagem.

Quadro 10 - Caso de uso "Administrar Notícias"

Caso de uso	Administrar notícias.
Ator Principal	Administrador.
Ator Secundário	-
Pré-condições	Possuir login de administrador.
Pós-condições	-
Fluxo Principal	
Ações do Ator	Ações do Sistema
ADMINISTRAR NOTÍCIAS	
1. Acessar página de cadastro de informações.	
	2. Sistema exibe o formulário para o cadastro da informação.
3. O administrador preenche os campos do formulário.	
	4. O sistema valida se os dados das

	informações são válidos e se não são duplicados, valida se o tamanho e tipo são válidos.
	5. Sistema guarda as informações no banco de dados.
	6. Sistema informa se foi efetuado com sucesso e retorna uma mensagem.

Quadro 11 - Caso de uso "Configurar perfil"

Caso de uso	Configurar perfil.	
Ator Principal	Usuário.	
Ator Secundário	-	
Pré-condições	Possuir um login.	
Pós-condições	-	
Fluxo Principal		
Ações do Ator		Ações do Sistema
CONFIGURAR PERFIL		
1. Acessar página de alteração de perfil.		
		2. Sistema exibe o formulário com valores preenchidos pelo banco.
3. O usuário preenche o formulário e clica em salvar.		
		4. O sistema valida se houve alguma alteração no perfil e se o e-mail é válido e não duplicado. Caso o preenchimento tenha sucesso, o sistema exibe um formulário para verificação de senha do usuário.
5. O usuário digita sua senha.		
		6. O sistema valida se a senha digitada corresponde à senha registrada para aquele usuário.
		7. Sistema guarda as informações no banco de dados.
		8. Sistema informa que foi alterado com sucesso e retorna uma mensagem.
		9. O sistema redireciona o usuário

	para a tela principal.
--	------------------------

2.5. Requisitos Não Funcionais

Quadro 12 - Requisitos não funcionais

Nº Requisito	Nome	Descrição
RNF001	Acesso à Internet.	O usuário deve ter acesso à internet para acessar o site.
RNF002	Banco de Dados	Banco de dados no qual os registros serão armazenados para utilização do site.
RNF003	Responsividade.	O site terá responsividade para ter um bom acesso em diferentes plataformas.
RNF004	Hardwares mínimos	Será necessário que o usuário tenha pelo menos um computador ou dispositivo que acesse a internet e abra os navegadores.
RNF005	Hospedagem.	É necessário para o funcionamento do site que esteja hospedado num servidor com acesso a banco de dados e todos os recursos web que o site necessite.
RNF006	Acessibilidade.	O site deverá ter acessibilidade para daltônicos, cegos, pessoas com visão debilitada e gerais.
RNF007	Notificações.	Notificar quando acontecer um erro na operação, ou quando um processo for concluído corretamente.
RNF008	Backend e API.	O sistema deve contar com um back-end robusto para processamento das requisições, expor uma API RESTful e integrar-se ao banco de dados em nuvem. Deve garantir segurança e performance adequadas.
RNF009	Documentação.	O projeto deve conter documentação seguindo as

3. PROJETO DO SOFTWARE

3.1. Tecnologias Utilizadas

Bootstrap² – É um framework Front-End que fornece estruturas de CSS para a criação de sites e aplicações responsivas de forma rápida e simples. Além disso, pode lidar com sites de desktop e páginas de dispositivos móveis da mesma forma.

Astah³ – Ferramenta de diagramação para indivíduos e equipes de todos os tamanhos. As ferramentas do Astah permitem criar uma grande variedade de diagramas e oferecem preços flexíveis adequados às suas necessidades.

Angular⁴ – Angular é um framework de código aberto desenvolvido pelo Google para a criação de aplicativos dinâmicos e interativos da web. Ele se baseia em TypeScript, uma linguagem superset do JavaScript, e utiliza uma abordagem orientada a componentes para a construção de interfaces que também foi utilizada para o desenvolvimento do back-end. O Angular oferece uma estrutura sólida para desenvolvimento, incluindo suporte a gerenciamento de estado, roteamento, validação de formulários e muito mais.

SweetAlert⁵ – O SweetAlert é uma API (Application Programming Interface), utilizada para estilizar e adicionar funcionalidades às caixas de diálogos em aplicações web. Ele oferece uma série de recursos que tornam a comunicação entre a página e a pessoa usuária muito mais funcional e elegante.

² <https://www.alura.com.br/artigos/bootstrap>

³ <https://astah.net/pt/>

⁴ <https://www.alura.com.br/artigos/angular-js>

⁵ <https://blog.betrybe.com/desenvolvimento-web/sweetalert/>

Github, DockerHub e Actions – tais foram utilizados para o controle e versionamento automático e semântico dos repositórios deste trabalho.

BetterStack para monitoramento de logs.

3.2. Usabilidade e Acessibilidade

A plataforma consiste em uma paleta de cores simples, que futuramente irá atender às necessidades das pessoas com daltonismo. Nossa plataforma possui um leitor de libras para os usuários com deficiência auditiva.

3.3. Modelo de Dados

3.2.2 Modelo Lógico

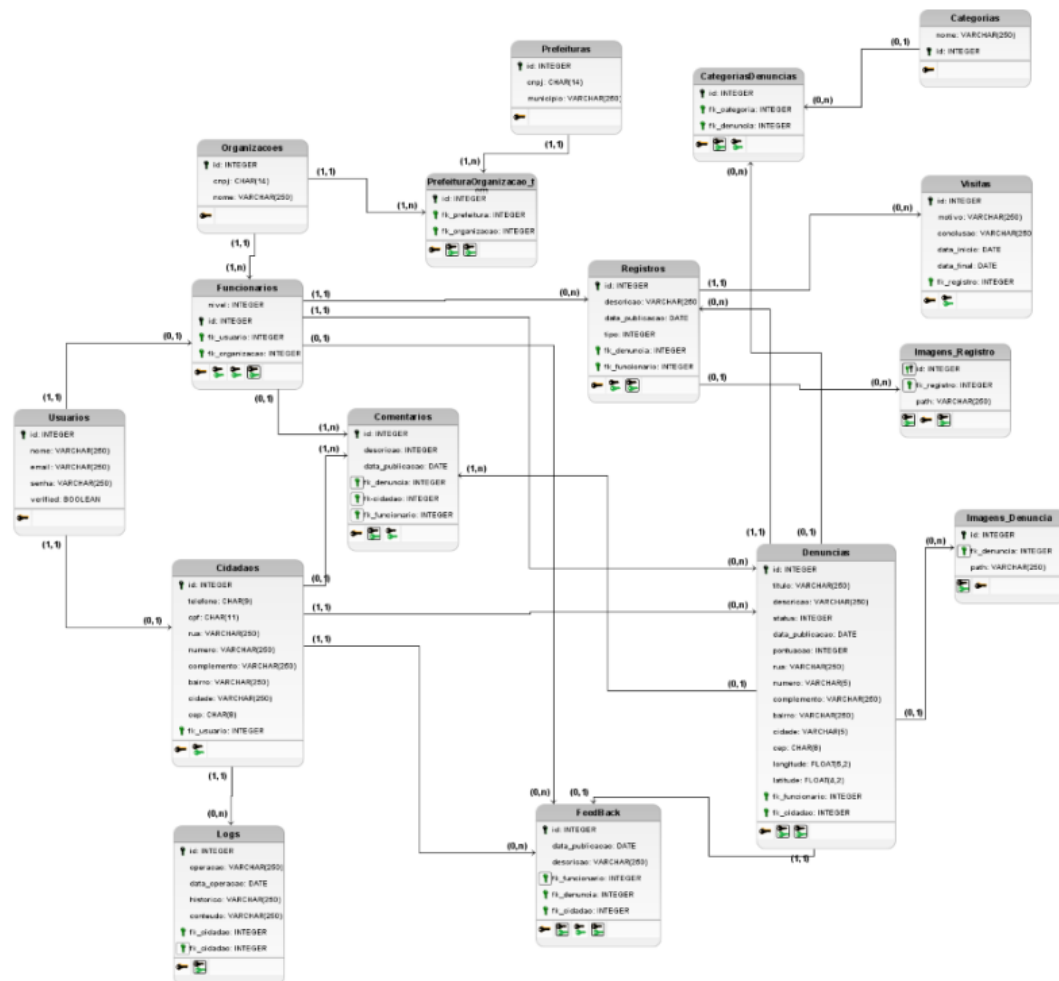


Figura 5 - Modelo Lógico

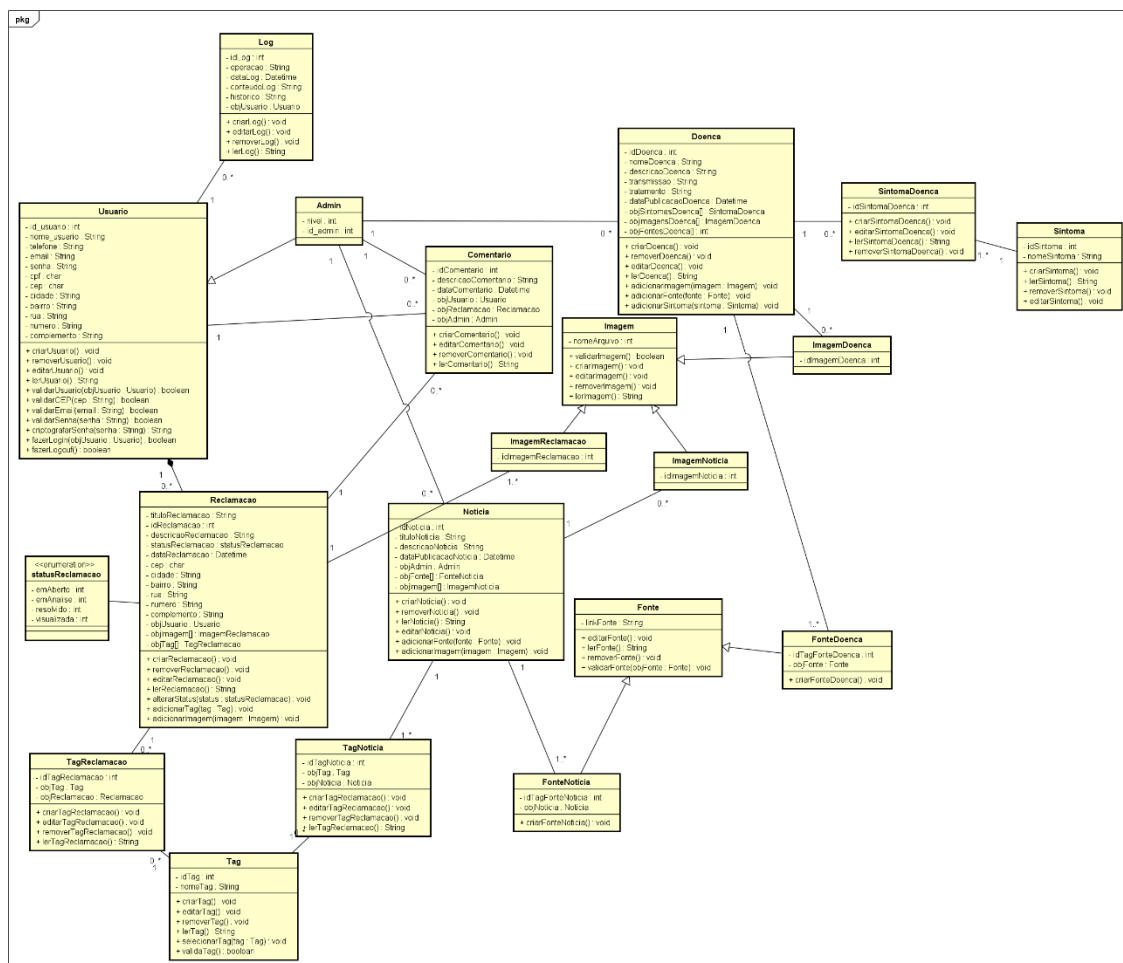


Figura 6 - Diagrama de Classes

3.5. Diagrama de Atividades

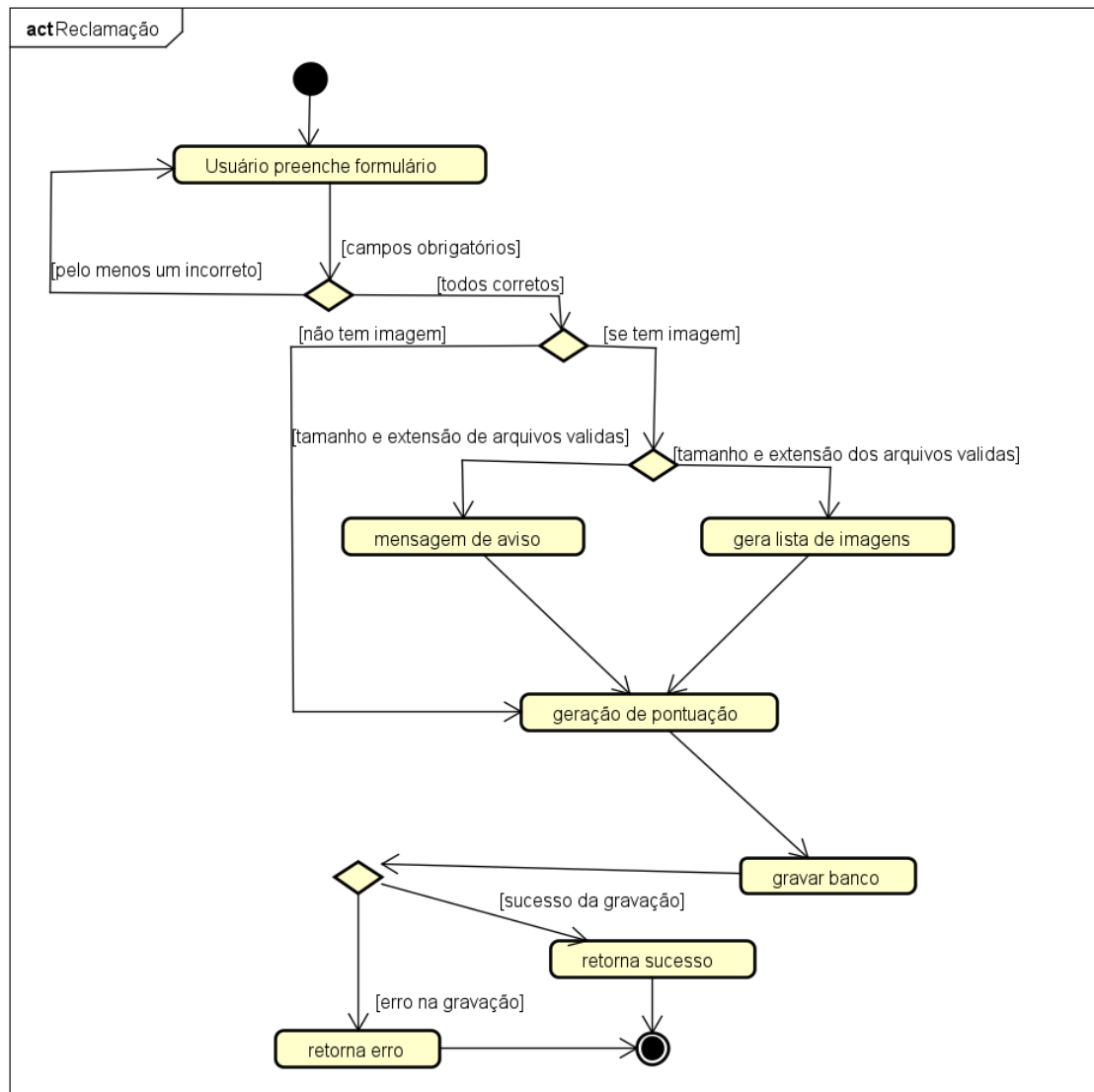


Figura 7 - Diagrama de Atividade "Adicionar Reclamação"

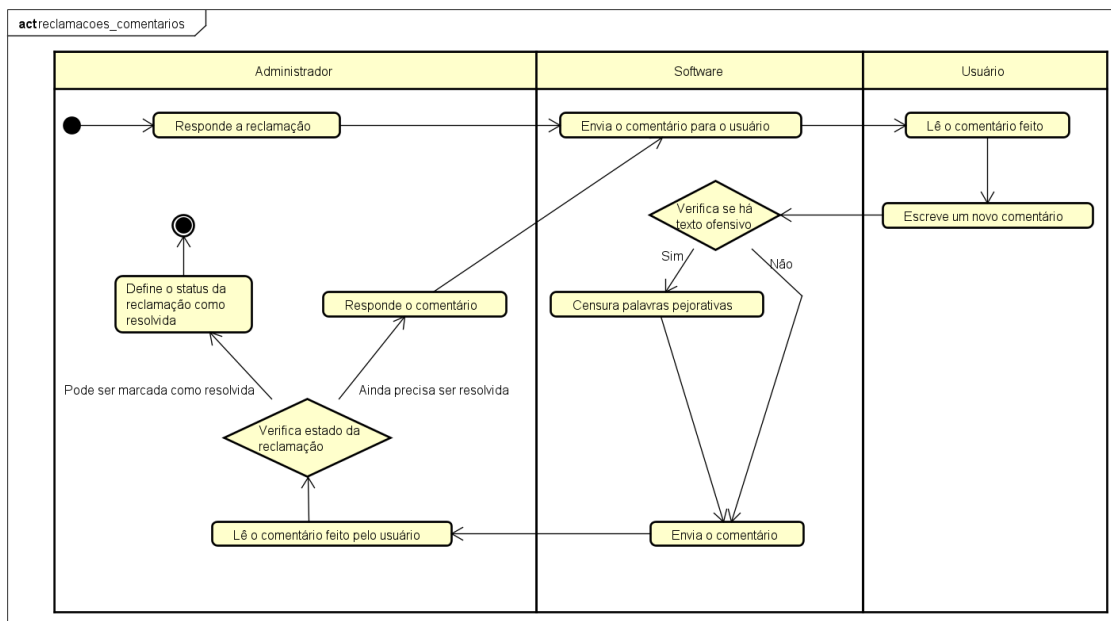


Figura 8 - Diagrama de Atividades "Comentários"

3.6. Diagrama de Sequência

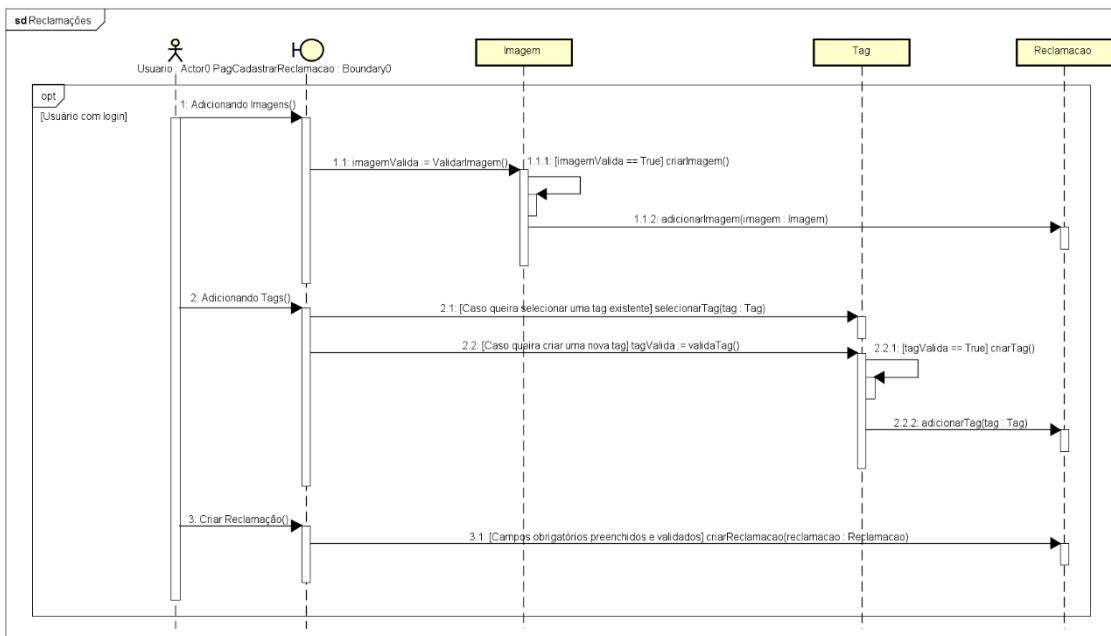


Figura 9 - Diagrama de Sequência "Reclamação"

3.7. Interfaces com o Usuário



Figura 10 – Barra de Navegação Comum



Figura 11 – Barra de Navegação Administrativa

Barra de Navegação: caso o usuário seja um cidadão, sua barra de navegação será como a Figura 10 ou, caso seja um administrador, será como a Figura 11.

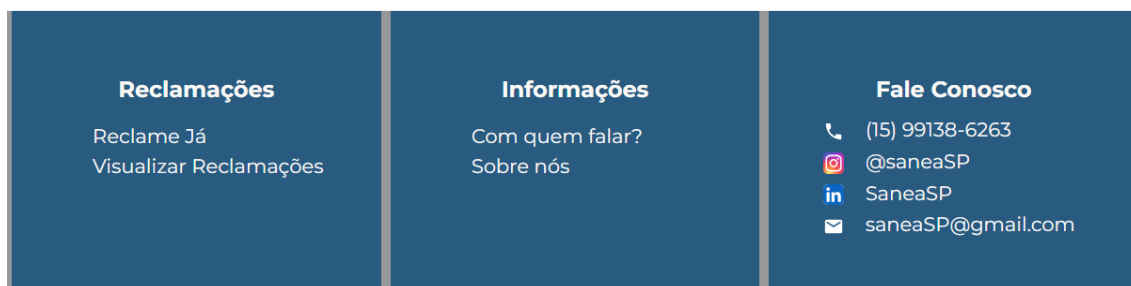


Figura 12 – Rodapé



Figura 13 – Página Inicial – 1

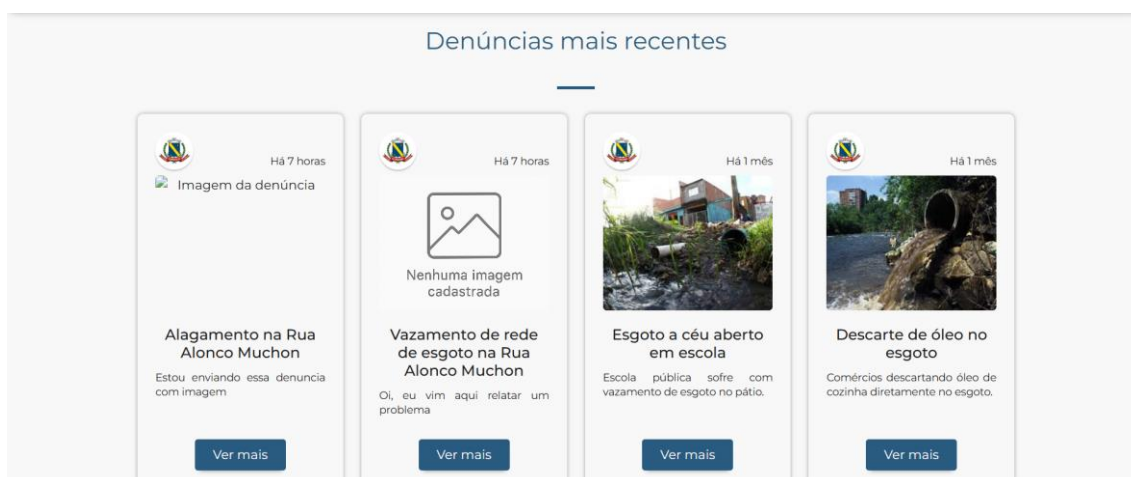


Figura 14 – Página Inicial – 2

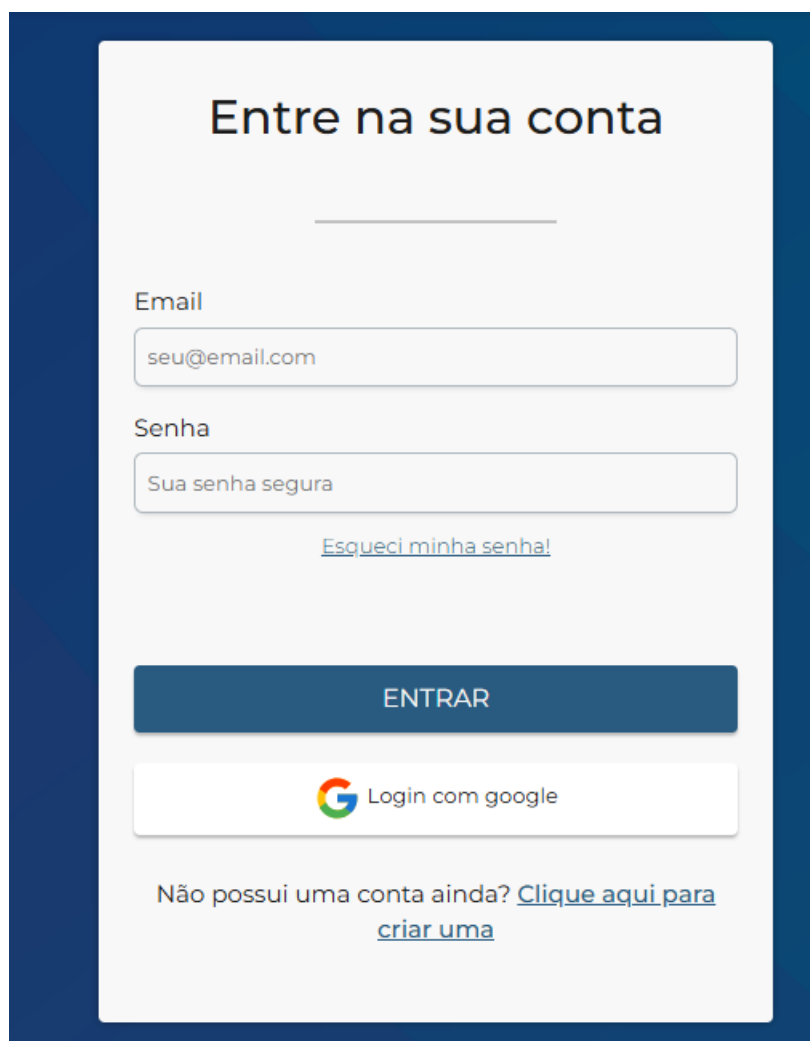


Figura 15 – Página Inicial – 3



Figura 16 – Página Inicial – 4

Página inicial: será a mesma para todos os usuários, porém, administradores não poderão criar as denúncias.

The image shows a login page with a dark blue border. At the top, the text "Entre na sua conta" is centered. Below it is a horizontal line. The "Email" section has a label and a text input field containing "seu@email.com". The "Senha" section has a label and a text input field containing "Sua senha segura". Below the password field is a link that says "Esqueci minha senha!". There is a dark blue button labeled "ENTRAR". Below the button is a white button with the Google logo and the text "Login com google". At the bottom, there is a link that says "Não possui uma conta ainda? Clique aqui para criar uma".


Entre na sua conta

Email

Senha

[Esqueci minha senha!](#)

ENTRAR

 Login com google

Não possui uma conta ainda? [Clique aqui para criar uma](#)

Figura 17 – Página de Login.

Página de login: caso o usuário esqueça sua senha, será enviado um e-mail para a recuperação.

Crie sua conta

Digite as informações necessárias para o seu cadastro



Nome

Seu nome

Email

seu@email.com

Senha

Sua senha

Confirme sua senha


Confirme sua senha

CRIAR

Já possui uma conta? [Clique aqui para entrar](#)

Figura 18 – Página de Cadastro.

Página de Cadastro: após o preenchimento para se cadastrar, o usuário irá receber um e-mail para validar seu cadastro.

Área dos cidadãosPainel administrativo

Sua conta

Início

Meus dados

Notificações

Endereços salvos

Informações de acesso

Email
gueffmatheus@gmail.com

Senha

Editar

Informações de identificação

Nome
Matheus Augusto Santos Gueff

CPF
123.456.789-00

Telefone
(15) 99261-5827

Editar

Seu endereço

CEP
18035-100

Cidade
Sorocaba

Bairro
Centro

Rua
Rua do Sol

Número
123

Complemento
Apto 45

Editar

Reclamações

Reclame Já
Visualizar Reclamações

Informações

Com quem falar?
Sobre nós

Fale Conosco

(15) 99138-6263
@saneaSP
SaneaSP
saneaSP@gmail.com

Figura 19 – Editar Perfil.

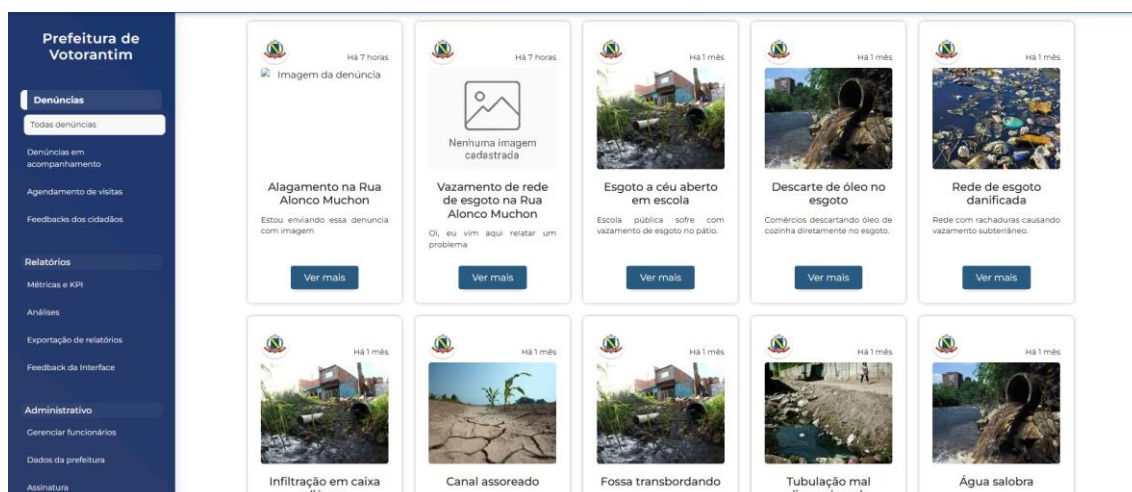


Figura 20 – Página Inicial do Administrador.

Página Administrador: nela, os administradores podem visualizar informações para geração de relatórios, como feedback das denúncias concluídas e feedbacks avaliando a interface da plataforma. Também é possível visualizar de forma gráfica algumas informações, como a quantidade de denúncias feitas, por exemplo, em um determinado bairro de uma cidade. Também é possível fazer exportações de PDF e Excel. Porém, para que um administrador tenha seu acesso, ele deve fazer um pagamento, caso contrário, será considerado um cidadão.

3.8. Better Stack

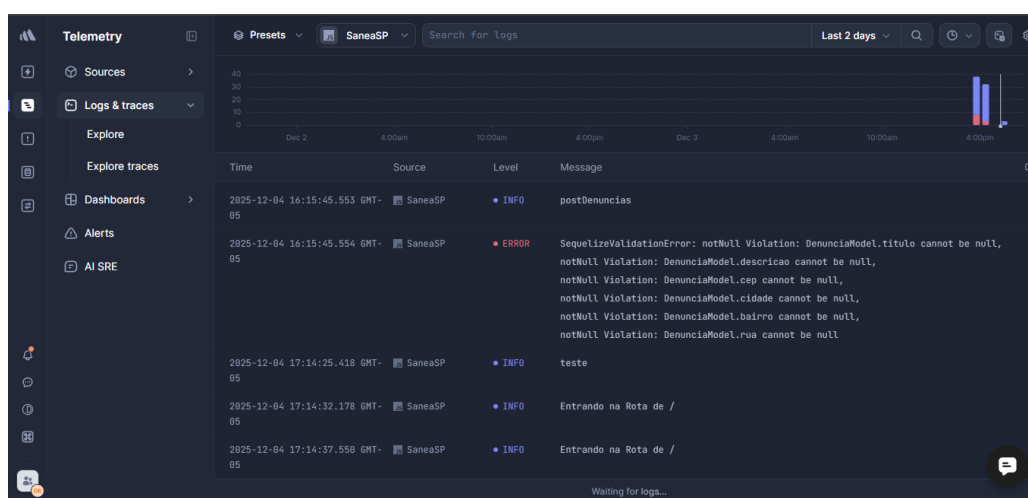


Figura 21 – Imagem Better Stack

3.9. Outras funcionalidades

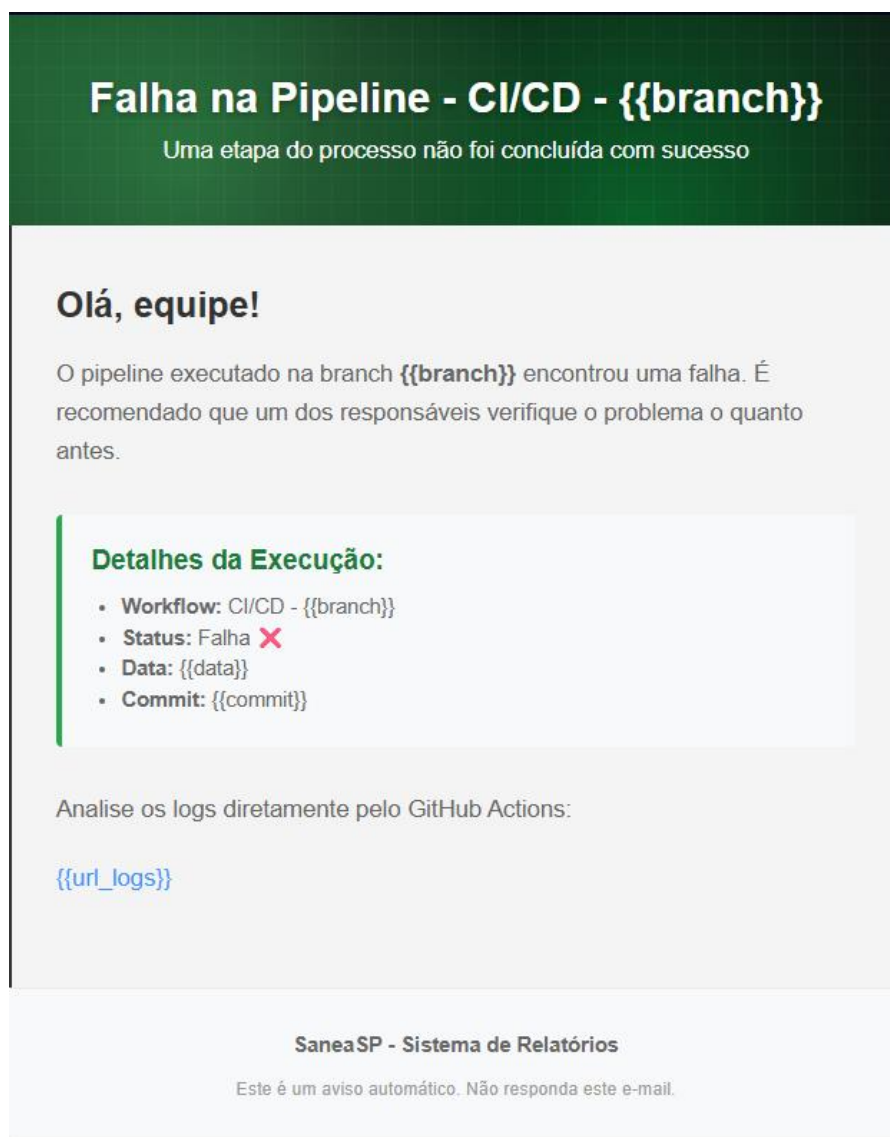


Figura 22 – E-mail do Actions

Actions: em caso de falha durante a execução de algum pipeline dos workflows, um e-mail é enviado para toda a equipe deste projeto.

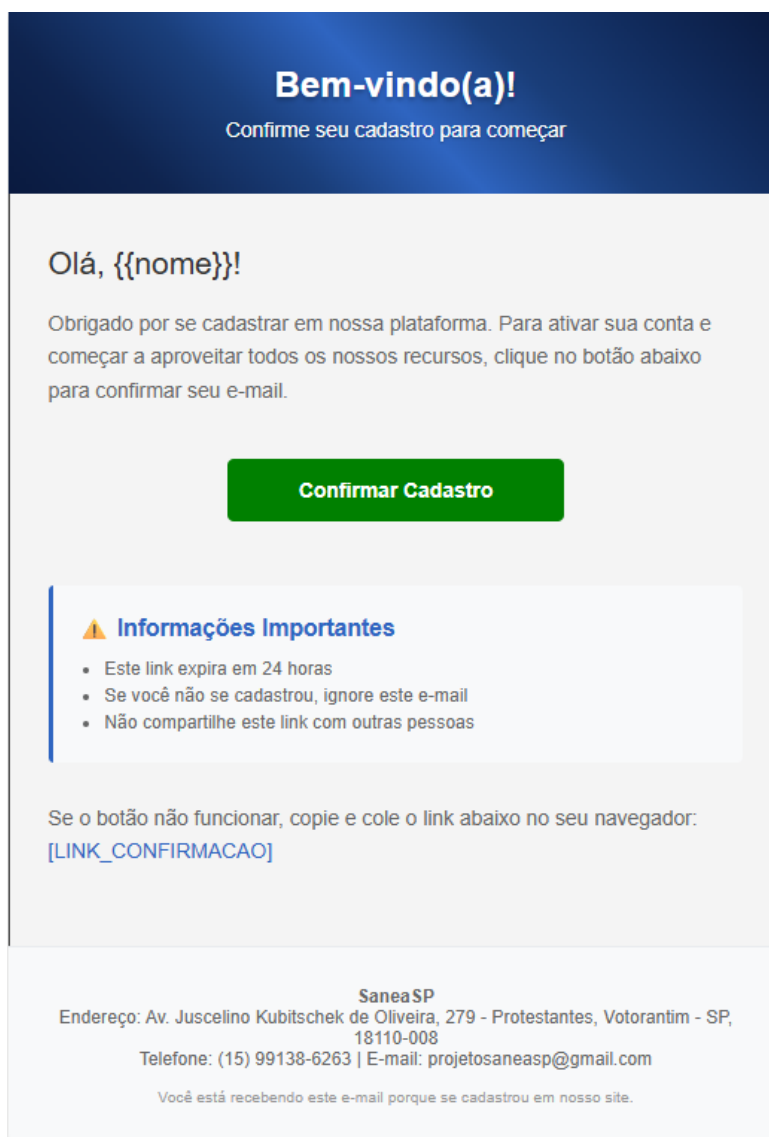


Figura 23 – E-mail enviado para validar o cadastro

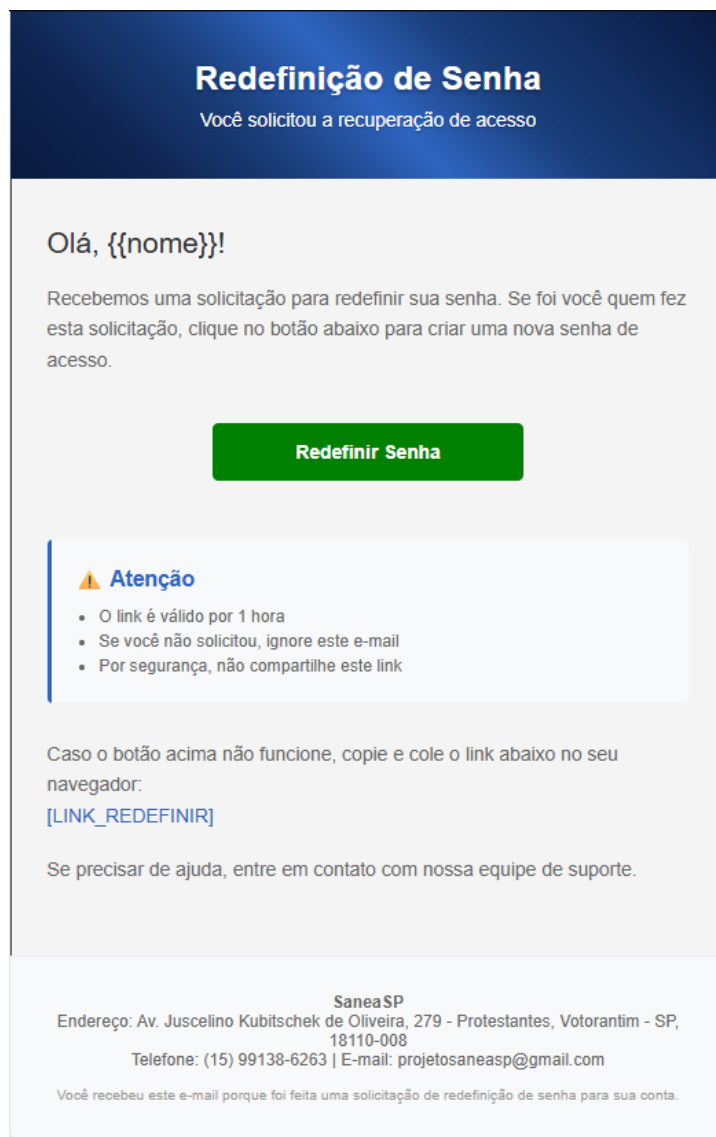


Figura 24 – E-mail enviado para recuperação de senha

E-mail: os e-mails são enviados de forma automática e futuramente serão implementados no dashboard dos administradores. Biblioteca: Nodemailer.

Pagamento: uma API de pagamento foi integrada com a plataforma, mas apenas para os administradores, uma forma de manter a hospedagem do site e o banco de dados. Enquanto o pagamento for efetuado, os administradores podem utilizar o dashboard. Biblioteca: Stripe.

Excel: utilizado para exportar todas as denúncias feitas em formato de tabela. Biblioteca: ExcelJS.

PDF: para os administradores, cada denúncia pode ser exportada com um PDF e futuramente será implementada para exportar mais dados no dashboard dos administradores. Bibliotecas: jsPDF e html2canvas.

Gráfico: serão utilizados para visualizar uma grande quantidade de dados já processados e contados para o dashboard dos administradores, além de serem interativos. Biblioteca: Chart.js.

Upload: os cidadãos podem carregar imagens para suas denúncias, até um limite de 10 imagens por denúncia. Biblioteca: Multer.

Geolocalização e Mapa: os cidadãos, para melhorar a descrição de suas denúncias, podem utilizar uma API para informar de forma mais eficaz a localização do problema. Os usuários também podem utilizar este recurso para complementar suas informações de endereço pessoal. Biblioteca: Leaflet.

Sistema de Avaliação: os cidadãos, após suas denúncias estarem concluídas, podem avaliar o serviço realizado pelo órgão responsável. Eles também podem avaliar a interface da plataforma. No dashboard dos administradores é possível visualizar os feedbacks das denúncias e interfaces.

Notificação em Tempo Real: foi implementada nos comentários das denúncias que são feitas entre um funcionário designado e o denunciante. Bibliotecas: WebSockets e Socket.IO.

Calendário: foi implementado no dashboard dos administradores, em que eles podem agendar as visitas em relação às denúncias por meio de um calendário dinâmico. Biblioteca: fullCalendar.

Pipeline para Deploy: usamos o render para hospedar o front e back. A pipeline cria imagens, atualiza elas nos repositórios do Dockerhub e faz o deploy da aplicação automaticamente.

3.10. Metodologia Utilizada

Para o desenvolvimento deste projeto, adotaram-se metodologias ágeis, especificamente o Scrum e o Kanban, a fim de garantir uma organização.

eficiente das tarefas e entregas contínuas. O progresso foi estruturado por meio de sprints e cartões (cards) com prazos definidos, alinhados aos requisitos estabelecidos. Cada entrega foi subdividida em pequenas tarefas, facilitando o acompanhamento e a execução gradual do projeto. Para gerenciar essas atividades, utilizou-se a plataforma Trello, que serviu como ferramenta principal de planejamento e controle.

Para o ambiente de desenvolvimento, utilizamos boas práticas de commit e criação de ramificações, separando o ambiente de desenvolvimento e o de produção, main e develop, padronizando ramificações como as iniciais feature, hotfix ou refactor e commits como feat, feat!, docs, refactor, fix e chore.

4. ESTRATÉGIA DE TESTES

Os testes a serem realizados no site consistem no conteúdo, na experiência dos usuários, na compatibilidade com diferentes navegadores no nosso sistema. Os testes realizados estão descritos a seguir:

Teste unitário: toda nova codificação foi brevemente testada para analisar se apontava algum defeito. Usa-se a ferramenta Visual Studio Code para mostrar erros de sintaxe no código.

Teste de conteúdo: todo texto e imagem foram revisados, sendo analisados a coerência e coesão do conteúdo.

Teste de interface: analisou-se se o site contém clareza na interatividade com o usuário, observando se houve omissão e/ou ambiguidade dos botões e na apresentação dos cadastros.

Teste de navegabilidade: analisou-se se o usuário conseguia acessar cada página e a navegação dele pelo site de forma rápida.

Teste de compatibilidade: avaliado o sistema em outros navegadores, sendo analisado se as mesmas funções foram executadas em diferentes navegadores.

5. IMPLEMENTAÇÃO

Links dos repositórios do GitHub e DockerHub:

- Front-end: <https://github.com/MathGueff/FrontEnd-SaneaSP>
- Back-end: <https://github.com/RyanCNP/Backend-SaneaSP>
- Trello: <https://trello.com/b/aogXELuz/saneasp>
- Imagem Docker front-end: <https://hub.docker.com/r/saneasp/front-sanea-prod/tags>
- Imagem Docker back-end: <https://hub.docker.com/r/saneasp/back-sanea/tags>

Links do projeto hospedado:

- Front-end: <https://saneasp.onrender.com>
- Back-end: <https://back-saneasp.onrender.com>
- Database: <https://dashboard.render.com/d/dpg-d4if91ogjchc73ek43f0-a>