



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS TRINDADE
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

ANDRÉ JOAQUIM BEZ BIROLO

**Lixo no ponto: um aplicativo para promover a colaboração comunitária na
coleta de resíduos nas áreas urbanas.**

PROPOSTA PARA DESENVOLVIMENTO DO TRABALHO DE CONCLUSÃO DE
CURSO

Florianópolis - SC

2024

André Joaquim Bez Birolo

Lixo no ponto: um aplicativo para promover a colaboração comunitária na coleta de resíduos nas áreas urbanas.

Proposta para desenvolvimento do trabalho de conclusão de curso como requisito para obtenção de aprovação na disciplina INE5638 - Introdução a Projetos, submetido ao curso de Sistemas de Informação do Campus Trindade da Universidade Federal de Santa Catarina.

Orientador(a): Prof^a. Dr^a. Fabiane Barreto Vavassori Benitti

Florianópolis - SC

2024

Sumário

Sumário.....	3
Lista de Figuras.....	5
Agradecimentos.....	6
Resumo.....	7
1. Introdução.....	8
1.1 Contexto.....	8
1.2 Problema.....	9
1.3 Solução.....	10
1.4 Objetivos.....	11
1.4.1 Objetivo geral.....	11
1.4.2 Objetivos específicos.....	11
1.5 Metodologia.....	11
1.5.1 Métodos de Pesquisa.....	12
1.5.2 Procedimentos metodológicos.....	12
Etapa 1: Fundamentação teórica.....	12
Etapa 2: Desenvolvimento.....	12
Etapa 3: Avaliação.....	13
2. Revisão Bibliográfica.....	13
2.1 Reciclagem.....	13
2.1.1 Órgãos Ambientais.....	13
2.1.2 Sustentabilidade.....	14
2.1.3 Tecnologia aliada da reciclagem.....	14
2.2 Progressive Web Application.....	14
2.2.1 Tecnologias para desenvolvimento do Front End.....	15
2.2.2 Tecnologias para desenvolvimento do Back End.....	15
2.2.3 Escolha de um Design System para a aplicação.....	16
2.2.4 UI & UX.....	16
2.2.5 Servidor Web.....	16
2.2.6 Banco de Dados.....	16
2.3 Soluções similares.....	17
2.3.1 Planejamento.....	17
2.3.2 Execução.....	18
2.3.3 Resultado.....	19
2.3.3.1 QP1: Quais são as soluções tecnológicas existentes que auxiliam na reciclagem ou que envolvem a existência de um aplicativo que auxilia na reciclagem?.....	20
2.3.3.2 QP2: Quais tecnologias são utilizadas nessas soluções?.....	20
2.3.3.3 QP3: Em que contexto essas soluções foram desenvolvidas (acadêmico ou empresarial)?.....	21
2.3.3.4 QP4: Como essas soluções foram avaliadas?.....	21

2.3.4 Considerações finais.....	21
3. Projeto.....	22
3.1 Requisitos.....	22
3.1.1 Requisitos Funcionais.....	22
3.1.2 Requisitos não Funcionais.....	23
3.1.3 Regra de Negócio.....	24
3.2 Casos de uso e Protótipos.....	24
3.3 Diagramas de Entidade e Relacionamento.....	28
4. Operacionalidade.....	29
4.1 Funcionalidades Comuns.....	30
4.2 Funcionalidades do Perfil Usuário.....	31
4.3 Funcionalidades do Perfil Administrador.....	35
4.4 Funcionalidades Invisíveis.....	37
4.5 Flexibilidade.....	37
5. Avaliação.....	39
5.1. Planejamento.....	39
5.2. Execução.....	41
5.3. Análise dos Resultados.....	42
5.3.1. Quão efetivo é o sistema em informar sobre os pontos de descarte e rotas de coleta de lixo? (Q1.1).....	42
5.3.2. Qual o grau de compreensão das informações sobre os pontos de descarte e rotas de coleta de lixo? (Q1.2).....	44
5.3.3. Qual a pontuação na escala SUS obtida pelo sistema? (Q2.1).....	45
5.4. Ameaças ao Estudo.....	45
6. Conclusões.....	46
6.1 Trabalhos Futuros.....	46
Referências bibliográficas.....	48
Anexo(s) e Apêndice(s) (minha autoria):.....	53
Apêndice A - Protótipos da Aplicação:.....	53
Apêndice B - Introdução à Avaliação:.....	61
Apêndice C - Questionário de Avaliação:.....	63
Apêndice D - Código Fonte Front End:.....	71
Apêndice E - Código Fonte Back End:.....	261
Apêndice F - Artigo SBC:.....	337

Lista de Figuras

Figura 1 – Processo de execução da string de busca.....	19
Figura 2 – Diagrama de caso de uso.....	25
Figura 3 – Modelo Entidade-Relacionamento.....	29
Figura 4 - Tela de Login.....	30
Figura 5 - Tela de Registro.....	31
Figura 6 - Página Inicial do Usuário.....	32
Figura 7 - Modal de Informações do Ponto de Descarte.....	32
Figura 8 - Página de Perfil do Usuário.....	33
Figura 9 - Página de Empresas de Reciclagem.....	34
Figura 10 - Página de Rotas e Horários de Coleta de Lixo.....	35
Figura 11 - Página Ponto de Descarte do Administrador.....	36
Figura 12 - Página de Cadastro de Empresas de Reciclagem.....	36
Figura 13 - Página de Cadastro de Rotas e Horários de Coleta de Lixo.....	37
Figura 14 - (a) Página Inicial e (b) Rotas em dispositivos móveis.....	38
Figura 15 - Pop-up de aviso para instalar PWA.....	38
Figura 16 - Exemplo visual da estrutura do GQM.....	39
Figura 17 - Passo a passo da etapa de execução.....	41
Figura 18 - Relação entre Pontos de Descartes e Rotas de Coletas Conhecidos Antes e Depois do Uso da Aplicação.....	43
Figura 19 - Grau de Compreensão das Informações Sobre os Pontos de Descarte e Rotas de Coleta de Lixo.....	44
Figura 20 - Pontuação SUS da Avaliação.....	45
Figura 21 – Protótipo de Cadastro de Usuário.....	53
Figura 20 – Protótipo de Login de Usuário.....	54
Figura 22 – Protótipo de Visualizar Pontos de Descarte.....	54
Figura 23 – Protótipo de Traçar Rota até o Local de Descarte.....	55
Figura 24 – Protótipo de Adicionar Ponto de Descarte.....	56
Figura 25 – Protótipo de Verificar Ponto de Descarte.....	57
Figura 26 – Protótipo de Visualizar Empresas de Reciclagem.....	58
Figura 27 – Protótipo de Visualizar Rotas e Horários de Coleta.....	58
Figura 28 – Protótipo de Visualizar Perfil de Usuário.....	59
Figura 29 – Protótipo de Cadastro de Rotas de Coleta.....	59
Figura 30 – Protótipo de Cadastro de Empresas Patrocinadoras.....	60
Figura 31 – Protótipo de Excluir um Ponto de Descarte.....	61

Agradecimentos

Agradeço à toda minha família, irmã, pai e mãe por me apoiarem diretamente ou indiretamente, possibilitando finalizar este trabalho e dar mais um passo importante na minha vida.

Obrigado à Franciele Rodrigues de Queiros, por estar comigo nos anos mais conturbados desse processo e ter me ajudado a me manter em pé para conseguir atingir esse objetivo que é a entrega do trabalho.

Agradeço à Rafaela Tasso e Beatriz Valio Weiss, que são mulheres poderosíssimas que me acompanharam durante toda a trajetória dessa etapa da minha vida, que nas matérias mais difíceis se propuseram a me ensinar e, além da faculdade, me tornaram uma pessoa melhor.

Agradeço à Fabiane Barreto Vavassori Benitti, minha orientadora, que me acompanhou, me instruiu e me colocou no caminho certo para a entrega deste projeto. Foi uma pessoa super atenciosa, entendeu minhas dores e também me colocou para trabalhar quando necessário.

Obrigado à banca por darem a oportunidade do meu trabalho ser avaliado.

Resumo

Este estudo contextualiza a situação da coleta seletiva no Brasil em 2019, destacando o aumento significativo na quantidade de resíduos sólidos produzidos e sua atual distribuição. Em resposta a esse problema, propõe-se um aplicativo que tem como finalidade disponibilizar informações visíveis à população sobre as rotas e horários das coletas seletivas, além de incluir marcadores no mapa que indicam a localização das estações de coleta seletiva e permitir a participação ativa da comunidade na sugestão de novos marcadores. A proposta objetiva contribuir para a promoção da colaboração comunitária na coleta de resíduos nas áreas urbanas e facilitar a comunicação entre doadores e receptores de resíduos, promovendo, assim, uma abordagem mais eficaz e colaborativa na gestão de resíduos sólidos.

Palavras chaves: reciclagem, tecnologia para reciclagem, educação ambiental

1. Introdução

1.1 Contexto

Na 1ª Conferência Internacional de Resíduos Sólidos (CIRSOL), o governador de Pernambuco, Paulo Câmara, cita em Milhomem (2022):

“A gestão dos resíduos sólidos não é preocupação recente da humanidade, muito menos apenas dos administradores públicos que se preocupam com a coletividade. Mesmo assim, o fenômeno é hoje um dos maiores desafios da nossa geração. Não só por sua grandeza, mas por sua ligação com a questão do clima”.

Segundo dados do Sistema Nacional de Informações sobre Saneamento (SNIS), em 2019, o Brasil produziu cerca de 79 milhões de toneladas de resíduos sólidos urbanos, sendo que apenas 59,5% foram coletados seletivamente e 40,5% foram destinados a lixões ou aterros controlados (SENADO FEDERAL, 2021), além disso para cada 10kg de resíduos disponibilizados para coleta, apenas 374 gramas são coletados de forma seletiva (ABRELPE, 2020).

O crescente número de resíduos sólidos revela a necessidade de implementar políticas públicas que promovam a gestão integrada de resíduos sólidos, envolvendo aspectos como a redução, a reutilização, a reciclagem, o tratamento e a disposição final adequada dos resíduos. Essa gestão integrada é prevista na Política Nacional de Resíduos Sólidos (PNRS) e busca abordar de forma abrangente e sustentável a problemática dos resíduos sólidos (SEMIL, 2019).

Nesse sentido, a Política Nacional de Resíduos Sólidos (PNRS), instituída pela Lei nº 12.305/2010 visa ao desenvolvimento sustentável, à prevenção e à redução da geração de resíduos, à reciclagem, à reutilização, ao tratamento e à disposição final ambientalmente adequada dos rejeitos. Além de visar também o estabelecimento de responsabilidade compartilhada pelo ciclo de vida dos produtos, envolvendo fabricantes, importadores, distribuidores, comerciantes, consumidores e poder público. Além disso, a lei prevê a elaboração de planos de resíduos sólidos em diferentes níveis (nacional, estadual, municipal, etc.) e a criação de metas para a eliminação dos lixões e a inclusão dos catadores de materiais recicláveis (BRASIL, 2010). Um dos instrumentos previstos pela PNRS é o Plano Municipal de Gestão Integrada de Resíduos Sólidos (PMGIRS), que deve ser elaborado pelos municípios como condição para o acesso a recursos da União destinados à limpeza urbana e ao manejo dos resíduos. O PMGIRS deve contemplar diagnóstico, prognóstico, metas, programas, projetos e ações voltados para a gestão dos resíduos sólidos no âmbito municipal.

Apesar da existência da Política Nacional de Resíduos Sólidos (PNRS) e do Plano Municipal de Gerenciamento Integrado de Resíduos Sólidos (PMGIRS), muitos municípios brasileiros ainda enfrentam dificuldades para implementar uma gestão integrada de resíduos sólidos eficiente e sustentável (ECYCLE, 2023). Entre os principais desafios estão:

- Falta de recursos financeiros e humanos para a implementação de políticas públicas eficientes de gestão integrada de resíduos sólidos (ECYCLE, 2023) (SZIGETHY, 2021).
- Ausência de educação ambiental e baixa adesão da população à coleta seletiva (ECYCLE, 2023) (NASCIMENTO et al., 2015).

- Precarização do trabalho dos catadores (SZIGETHY, 2021).
- Insuficiência de infraestrutura e tecnologia para o tratamento e a destinação final dos resíduos (SZIGETHY, 2021).

Esses fatores comprometem não apenas o cumprimento das metas estabelecidas pela PNRS, mas também a qualidade de vida das pessoas e o equilíbrio do meio ambiente. É fundamental que sejam adotadas medidas para enfrentar esses desafios e promover uma gestão integrada de resíduos sólidos eficiente e sustentável, como a implementação de programas de educação ambiental, a valorização dos catadores e a adoção de tecnologias que promovam o gerenciamento e a gestão integrada dos resíduos sólidos (SZIGETHY, 2021) (ECYCLE, 2023).

1.2 Problema

Em suma, os problemas mencionados podem ter várias consequências negativas na sociedade. Os impactos ambientais do descarte inadequado de resíduos podem afetar a qualidade de vida das pessoas, prejudicando a biodiversidade e contribuindo para as mudanças climáticas. Isso pode levar a problemas como a perda de habitats naturais, aumento do nível do mar, eventos climáticos extremos e prejuízos econômicos.

Os impactos na saúde pública são significativos. O aumento do risco de doenças transmitidas por vetores e problemas respiratórios e outras doenças relacionadas à poluição, assim como a poluição da água e do solo, podem afetar a saúde e o bem-estar das pessoas. Além disso, os custos associados ao tratamento dessas doenças podem ser elevados, afetando a economia e o sistema de saúde (ROCHA; MOURA JUNIOR; MAGALHÃES, 2012).

O descarte inadequado de resíduos pode afetar a biodiversidade, levando à perda de habitats naturais e à extinção de espécies. Assim como também pode contribuir para o aumento do nível do mar, causando a erosão das praias, a perda de áreas costeiras e contribuir para as mudanças climáticas, causando eventos climáticos extremos, como enchentes e deslizamentos de terra. Esses eventos podem provocar mortes, ferimentos, doenças e prejuízos materiais (BAUMINAS, 2023) (ISOPOR, 2022).

Os impactos econômicos da má gestão de resíduos também podem ser significativos. O aumento dos custos associados à limpeza e remediação de áreas contaminadas pode afetar negativamente o orçamento dos municípios e afetar o turismo e outras atividades econômicas. Além disso, a má gestão de resíduos pode afetar negativamente a imagem da cidade, desencorajando investimentos e afetando o desenvolvimento econômico (ROCHA; MOURA JUNIOR; MAGALHÃES, 2012).

A gestão adequada de resíduos é uma questão fundamental para a preservação do meio ambiente e da saúde pública, pois evita a contaminação da água, do ar e do solo, bem como a proliferação de doenças e pragas. No entanto, muitas pessoas ainda desconhecem os métodos corretos de descarte e reciclagem dos resíduos que geram, o que resulta em comportamentos inadequados, como o despejo de lixo em locais impróprios (ECYCLE, 2023) (WILLICH, 2022). Para enfrentar esse desafio, é necessário educar a população sobre a importância e os benefícios da gestão de resíduos, bem como fornecer soluções tecnológicas inovadoras que facilitem e estimulem essa prática.

1.3 Solução

Nesse sentido, uma iniciativa possível é a criação de aplicações móveis e/ou online que informem os cidadãos sobre os tipos e as formas de descarte e reciclagem dos resíduos (VERTOWN, 2023). Essa solução tecnológica não só melhora a eficiência da gestão de resíduos, mas também visa engajar os cidadãos de forma mais efetiva na adoção de práticas sustentáveis, transformando o problema do descarte inadequado em uma oportunidade para a construção de comunidades mais conscientes e ecologicamente responsáveis.

Diante desse cenário, torna-se relevante buscar soluções inovadoras que possam contribuir para melhorar a gestão integrada de resíduos sólidos nos municípios brasileiros. Uma dessas soluções é o uso das tecnologias da informação e comunicação (TICs) como ferramentas para sensibilizar, informar e engajar os cidadãos na separação e no descarte correto dos resíduos. As TICs podem facilitar o acesso à informação sobre os serviços públicos de limpeza urbana e coleta seletiva, bem como sobre os locais adequados para a entrega voluntária ou a reciclagem dos resíduos. Além disso, as TICs podem estimular a participação social e o controle social na gestão dos resíduos sólidos, por meio de canais de comunicação entre os usuários, os gestores públicos e os agentes envolvidos na cadeia produtiva dos resíduos.

Nesse contexto, o presente trabalho tem como foco o desenvolvimento de um aplicativo *web* (PWA - *Progressive Web Application*) para promover a gestão integrada de resíduos em Florianópolis e nas demais áreas urbanas. O aplicativo *web* terá como objetivo conscientizar a população sobre a importância da gestão adequada de resíduos e facilitar o descarte adequado de resíduos. Para isso, o aplicativo *web* irá apresentar informações sobre as rotas de cada caminhão de lixo, seja ele comum, orgânico, de coleta de vidro e etc, além de mostrar no mapa vários *pins*¹ que indicam onde a população pode realizar determinado descarte. Além disso, o aplicativo *web* também fornecerá informações sobre como reciclar e desenvolver a reciclagem na cidade.

Com a finalidade de aperfeiçoar a aplicação, prevê-se que haja colaboração da comunidade através da possibilidade do usuário poder informar pontos de coleta seletiva que não estão disponíveis.

Através dessas funcionalidades, espera-se que o aplicativo *web* contribua para a promoção da gestão integrada de resíduos em Florianópolis e em outros municípios interessados, abordando alguns dos problemas identificados anteriormente. Por exemplo, ao fornecer informações sobre as rotas dos caminhões de lixo e os locais de descarte adequado, o aplicativo *web* pode ajudar a aumentar a conscientização da população sobre a importância da gestão adequada de resíduos. Além disso, ao fornecer informações sobre como reciclar e desenvolver a reciclagem na cidade, o aplicativo *web* pode ajudar a promover a reciclagem e reduzir os custos associados à gestão de resíduos.

Florianópolis é uma cidade que possui uma grande diversidade ambiental, cultural e socioeconômica, sendo reconhecida como um polo de inovação e turismo no país. No entanto, a cidade também enfrenta problemas relacionados à gestão dos resíduos sólidos, como a geração excessiva de lixo, a baixa cobertura da coleta seletiva, a destinação inadequada de resíduos especiais e a falta de integração entre os diferentes atores envolvidos na gestão dos resíduos (KALFELTZ; SANTOS, 2021). Esses problemas afetam

¹ Uma marca ou ícone usado nos mapas digitais para representar um ponto específico na superfície da Terra

negativamente a qualidade ambiental e a saúde pública da cidade, além de comprometerem sua imagem e seu potencial turístico.

A solução proposta está alinhada com as diretrizes estabelecidas pela Política Nacional de Resíduos Sólidos (PNRS) e pelo Plano Municipal de Gestão Integrada de Resíduos Sólidos de Florianópolis, que visam promover a gestão integrada de resíduos através da coleta seletiva, reciclagem e destinação adequada. Além disso, essa solução pode trazer benefícios sociais, econômicos e ambientais significativos para a cidade de Florianópolis.

Com a implementação dessa solução, várias entidades podem ser beneficiadas. Por exemplo, os catadores de materiais recicláveis podem ter mais oportunidades de emprego e renda com a coleta, separação e venda de materiais recicláveis. A indústria da reciclagem também pode se beneficiar com o fornecimento de materiais recicláveis de alta qualidade para a fabricação de novos produtos.

Além disso, o município pode economizar com os custos associados à gestão de resíduos, reduzindo o volume de resíduos enviados para aterros sanitários ou incinerados. O meio ambiente também pode ser beneficiado com a redução da poluição do solo, da água e do ar, além da economia de recursos naturais e redução das emissões de gases de efeito estufa. Portanto, essa solução pode ter um impacto positivo em vários aspectos da sociedade.

1.4 Objetivos

Essa seção visa esclarecer os atuais objetivos deste projeto.

1.4.1 Objetivo geral

O objetivo geral do presente trabalho é desenvolver um sistema que contribua para uma gestão integrada da coleta de resíduos sólidos urbanos.

1.4.2 Objetivos específicos

Com o objetivo de atingir o objetivo geral deste projeto, têm-se os seguintes objetivos específicos:

- Analisar produtos semelhantes existentes.
- Projetar e desenvolver um sistema que auxilie a comunidade quanto ao descarte correto de resíduos sólidos urbanos.
- Avaliar o sistema.

1.5 Metodologia

Essa seção visa esclarecer a metodologia desenvolvida durante o estudo deste projeto.

1.5.1 Métodos de Pesquisa

Do ponto de vista da sua natureza foi classificado como pesquisa aplicada, visto que tem como objetivo gerar conhecimento para aplicação prática e à solução de problemas específicos cita Silva e Menezes (2005), neste caso o conhecimento para o desenvolvimento da aplicação e a tentativa da solução quanto o assunto descarte correto de resíduos sólidos urbanos.

Do ponto de vista da forma de abordagem do problema foi identificado como pesquisa quantitativa considerando que o projeto é traduzido em números, a partir disso, classificá-los e analisá-los e também como qualitativo uma vez que o estudo irá buscar através de pesquisas com os usuários resultados que resultem em uma análise e interpretação (SILVA; MENEZES, 2005).

Por fim, quanto do ponto de vista dos objetivos foi abordado uma pesquisa exploratória já que visa proporcionar maior familiaridade com o problema com o objetivo de torná-lo explícito ou a construir hipóteses, cita Silva e Menezes (2005). Dessa forma, é necessário envolver levantamento bibliográfico.

1.5.2 Procedimentos metodológicos

O progresso deste projeto está previsto para seguir o plano de trabalho a seguir:

Etapa 1: Fundamentação teórica:

Passo em que é realizada a revisão bibliográfica sobre os seguintes assuntos:

- a) Revisão bibliográfica sobre reciclagem: pesquisa teórica envolvendo os principais conceitos relacionados à reciclagem focando no potencial da tecnologia como maior aliado.
- b) Revisão bibliográfica de tecnologias de desenvolvimento: conceitos teóricos e práticos para implementação das duas tecnologias em conjunto.
- c) Mapeamento das soluções similares: buscar identificar soluções tecnológicas que tenham como objetivo facilitar a gestão de resíduos sólidos nas cidades. Esta atividade observará alguns aspectos de uma abordagem sistemática conforme Kitchenham e Charters (2007).

Etapa 2: Desenvolvimento

Etapa na qual é planejada, desenvolvida e testada a aplicação conforme os seguintes passos:

- a) Análise e projeto: levantamento dos requisitos e características técnicas que a aplicação terá, além da sua modelagem de classes utilizando a *UML (Unified Modeling Language)* e a prototipação de fidelidade alta através da ferramenta Figma.

- b) Implementação: maior etapa do desenvolvimento a qual será destinada para o desenvolvimento da aplicação utilizando as tecnologias posteriormente escolhidas.
- c) Teste: etapa destinada ao teste de integração das funcionalidades da aplicação.

Etapa 3: Avaliação

Nessa fase, a abordagem GQM (*Goal-Question-Metric*) vai ser utilizada como um arcabouço metodológico fundamental no seguimento desta pesquisa. Essa abordagem se destaca por sua estrutura clara e eficaz para o planejamento, execução e análise do estudo. Durante o desenvolvimento deste trabalho, as três etapas fundamentais da abordagem GQM serão observadas e aplicadas, com o intuito de estabelecer metas e objetivos claros (a fase de planejamento), elaborar questões de pesquisa (a fase de execução) e, por fim, analisar os resultados obtidos através das métricas definidas (a fase de análise) (BASILI; CALDIERA; ROMBACH, 1994).

- a) Planejamento: é estabelecido o plano para a implementação do processo avaliativo.
- b) Execução: os usuários finais avaliarão o sistema e fornecerão sugestões de aprimoramento.
- c) Análise de resultados: examinaremos os resultados, a fim de chegar em conclusões apropriadas.

2. Revisão Bibliográfica

Nesta seção, são abordados conceitos e tecnologias predominantes relacionadas ao projeto.

2.1 Reciclagem

Segundo Wojnowska-Baryla (2022), a reciclagem de resíduos orgânicos é um processo de gestão no qual os resíduos são reciclados ou convertidos em matéria útil por diferentes métodos de reciclagem. A reciclagem orgânica pode oferecer ao material um novo ciclo de vida, permitindo que ele seja reaproveitado de maneira sustentável e benéfica para o meio ambiente.

2.1.1 Órgãos Ambientais

Os órgãos ambientais, conforme estabelecido pelo artigo 6º da Lei nº 6.938/81 (BRASIL, 1981), são estruturas fundamentais do Sistema Nacional do Meio Ambiente (SISNAMA). Eles têm a função de promover a proteção e a recuperação da qualidade ambiental, objetivando um ambiente ecologicamente equilibrado e essencial à qualidade de vida. Esses órgãos atuam na formulação e aplicação de políticas ambientais, controle e

fiscalização de atividades que impactam o meio ambiente, garantindo assim a sustentabilidade dos recursos naturais para as presentes e futuras gerações.

2.1.2 Sustentabilidade

Elkington (1994), ao cunhar o termo “Triple Bottom Line”, destacou a sustentabilidade como o equilíbrio entre os pilares ambiental, econômico e social. Ele enfatizou que o compromisso das empresas com a sustentabilidade é fundamental, reconhecendo que negócios sustentáveis dependem de mercados estáveis e da capacidade de gerenciar recursos tecnológicos e financeiros para fomentar o desenvolvimento sustentável. Elkington (1994) argumenta que o desenvolvimento sustentável só se realiza quando essas três dimensões operam em conjunto, de forma equilibrada e integrada.

2.1.3 Tecnologia aliada da reciclagem

Na atualidade, a tecnologia tem se mostrado uma parceira essencial no processo de reciclagem, introduzindo inovações e soluções que aumentam a eficiência e promovem a sustentabilidade. A implementação de novas tecnologias têm melhorado significativamente a reciclagem de materiais como alumínio, plástico e papel, ajudando a diminuir o volume de resíduos e a fortalecer a preservação ambiental. Um exemplo disso é a aplicação do Desengate Térmico (TDT) na reciclagem do alumínio, que, segundo o GRUPO INTERAÇÃO (2024), tem facilitado a reutilização desse material, gerando novos produtos e abrindo portas para métodos de reciclagem mais eficientes.

De acordo com Convale (2023), a Tecnologia da Informação (TI) tem se mostrado uma aliada essencial no processo de reciclagem, introduzindo inovações e soluções que aumentam a eficiência e promovem a sustentabilidade. A implementação de novas tecnologias têm melhorado significativamente a reciclagem de materiais como alumínio, plástico e papel, ajudando a diminuir o volume de resíduos e a fortalecer a preservação ambiental. Um exemplo disso é a aplicação de sistemas inteligentes na gestão de resíduos, que facilitam a coleta, separação e reutilização de materiais, gerando novos produtos e abrindo portas para métodos de reciclagem mais eficientes. Além disso, a TI pode ser utilizada para desenvolver aplicativos que incentivam a reciclagem e a conscientização ambiental, contribuindo para a construção de uma sociedade mais sustentável.

2.2 Progressive Web Application

Compreendendo a importância da tecnologia no processo de reciclagem, é possível vislumbrar a aplicação de tendências emergentes, como as *Progressive Web Applications* (PWAs), para aprimorar ainda mais esses processos. As PWAs, construídas com tecnologias da web como HTML, CSS e JavaScript, oferecem uma experiência semelhante a aplicativos nativos em dispositivos móveis. Essas aplicações podem ser desenvolvidas para fornecer informações sobre pontos de coleta de reciclagem, horários de coleta e tipos de materiais recicláveis, facilitando a participação dos cidadãos no processo de reciclagem. Além disso, podem incluir funcionalidades que incentivam a reciclagem, como recompensas por comportamentos sustentáveis (KINGHOST, 2024). Portanto, a combinação da tecnologia com a reciclagem através de PWAs pode contribuir significativamente para a promoção da sustentabilidade e a preservação do meio ambiente.

De acordo com o Tecnoblog (2023), *PWAs (Progressive Web Apps)* são uma nova tendência que representa uma revolução no uso de dispositivos móveis, combinando as vantagens dos aplicativos nativos com as da web. Eles são construídos usando tecnologias da web, como HTML, CSS e JavaScript, mas oferecem uma experiência semelhante a aplicativos nativos em dispositivos móveis, como o X, que pode ser instalado na tela inicial do celular e funciona exatamente como um aplicativo nativo.

Em resumo, as *PWAs* são uma solução híbrida entre aplicativos nativos e web, que oferecem uma experiência semelhante a aplicativos nativos em dispositivos móveis, com a facilidade de desenvolvimento e manutenção de aplicativos web.

2.2.1 Tecnologias para desenvolvimento do Front End

Segundo Alura (2024), Angular é uma plataforma de desenvolvimento de aplicativos web que oferece uma estrutura para criar aplicativos escaláveis e de alto desempenho. A plataforma é baseada em TypeScript e oferece uma variedade de recursos para ajudar os desenvolvedores a criar aplicativos robustos e de fácil manutenção, como a arquitetura de componentes, a injeção de dependência e a detecção automática de alterações. Além disso, Angular tem uma forte comunidade de desenvolvedores e uma vasta gama de recursos de aprendizado disponíveis, como cursos, tutoriais e livros.

Baseado nisso, para o desenvolvimento do Front End será utilizado o *framework* Angular.

2.2.2 Tecnologias para desenvolvimento do Back End

O back end é a parte do desenvolvimento de software que é responsável por processar dados e gerenciar as interações entre a aplicação e o banco de dados. É a parte do sistema que não é visível para o usuário final, mas é fundamental para o funcionamento da aplicação. De acordo com SOUTO, M (2024), o back-end pode ser desenvolvido utilizando diferentes linguagens de programação, como Java, Python, Ruby, entre outras.

No contexto do desenvolvimento web, o back-end é responsável por processar as requisições feitas pelo front-end e retornar uma resposta adequada. Isso inclui a autenticação de usuários, a manipulação de dados no banco de dados, a geração de relatórios e a implementação de regras de negócio. Além disso, o back-end pode ser responsável por fornecer serviços de *API* para outras aplicações ou sistemas.

Um exemplo de *framework* popular para o desenvolvimento de back-end em Java é o Spring. O Spring é uma plataforma de desenvolvimento de aplicativos Java *open source* que fornece uma estrutura para construir aplicativos robustos e escaláveis. Ele oferece uma ampla gama de recursos, como injeção de dependência, gerenciamento de transações, segurança e integração com outras tecnologias. Além disso, o Spring tem uma grande comunidade de desenvolvedores e uma vasta gama de recursos de aprendizado disponíveis, como cursos, tutoriais e livros (BROADCOM, 2024).

Em resumo, o back-end é uma parte fundamental do desenvolvimento de *software* e é responsável por processar dados e gerenciar as interações entre a aplicação e o banco de dados. O Spring é um *framework* popular para o desenvolvimento de back-end em Java, oferecendo uma ampla gama de recursos e uma grande comunidade de desenvolvedores e será utilizado neste projeto.

2.2.3 Escolha de um Design System para a aplicação

Para o *Design System*, será utilizado o Angular Material. Segundo a Google LLC (2024), o Angular Material é uma biblioteca de componentes de interface de usuário baseados no Material Design, um conjunto de diretrizes de design desenvolvido pelo Google. Esses componentes são especialmente otimizados para funcionar de forma integrada com o Angular, proporcionando uma experiência de desenvolvimento consistente e eficiente. Ao utilizar Angular Material, os desenvolvedores têm acesso a uma ampla variedade de componentes prontos para uso, como botões, barras de navegação, caixas de diálogo, tabelas, entre outros, todos seguindo as diretrizes de design do Material Design. Isso não apenas agiliza o processo de desenvolvimento, mas também garante uma interface de usuário coesa e intuitiva, melhorando a usabilidade e a experiência do usuário final.

2.2.4 UI & UX

UI (*User Interface*) e UX (*User Experience*) são conceitos essenciais no desenvolvimento de produtos digitais, onde a UI se refere ao design visual da interface e a UX à experiência geral do usuário ao interagir com o sistema. Em relação às *PWAs* (*Progressive Web Apps*), a qualidade da UI e da UX desempenha um papel crucial, pois as *PWAs* são aplicativos web que buscam oferecer uma experiência semelhante a aplicativos nativos em dispositivos móveis, segundo a Rockcontent (2023). Portanto, ao criar uma *PWA*, é fundamental garantir uma interface intuitiva, atraente e funcional, juntamente com uma experiência de usuário satisfatória, considerando aspectos como usabilidade, acessibilidade e estética.

2.2.5 Servidor Web

O conceito de Servidor Web refere-se a um programa de computador que hospeda e disponibiliza conteúdo na *World Wide Web* (*WWW*). Ele é responsável por receber solicitações de clientes, como navegadores web, e enviar respostas contendo o conteúdo solicitado. O servidor web pode ser implementado usando diferentes tecnologias, como Apache, Nginx ou Microsoft IIS.

Em relação às *Progressive Web Apps* (*PWA*), o servidor web desempenha um papel fundamental, pois é o responsável por hospedar e disponibilizar o aplicativo web progressivo na *WWW* (GRONER, 2018). Além disso, o servidor web pode fornecer recursos adicionais, como autenticação de usuário, segurança e gerenciamento de conteúdo, que podem ser integrados a uma *PWA* (CASA DO DESENVOLVEDOR, 2023).

Em resumo, o servidor web é um componente fundamental na infraestrutura de uma *PWA*, pois é responsável por hospedar e disponibilizar o aplicativo na *WWW* e fornecer recursos adicionais que podem ser integrados à *PWA* (GRONER, 2018) (CASA DO DESENVOLVEDOR, 2023).

2.2.6 Banco de Dados

Bancos de dados são sistemas que permitem armazenar, recuperar e manipular dados de forma estruturada, sendo essenciais para gerenciar o conteúdo dinâmico de uma *PWA*, como informações de usuários e dados de aplicativos (ROCKCONTENT, 2024). Para o banco de dados do projeto, será utilizado o PostgreSQL, um sistema de banco de dados

objeto-relacional avançado e de código aberto, conhecido por sua confiabilidade, robustez de recursos e desempenho (COODESH, 2024).

O banco de dados é uma parte fundamental da arquitetura de uma *PWA*, pois é responsável por armazenar e gerenciar os dados do aplicativo, como informações de usuários, preferências e configurações (ROCKCONTENT, 2024). O PostgreSQL é uma ótima opção para bancos de dados de *PWAs*, pois oferece recursos avançados e de alto desempenho, garantindo a confiabilidade e robustez do aplicativo (COODESH, 2024).

2.3 Soluções similares

Neste capítulo, é apresentado algumas práticas de um mapeamento sistemático desenvolvido para obter uma visão abrangente sobre as soluções tecnológicas existentes que auxiliam na reciclagem ou que envolvem a existência de um aplicativo que auxilia na reciclagem. Em geral, a execução de um mapeamento sistemático, em vez de uma revisão bibliográfica, é preferível para investigar uma área de pesquisa mais ampla ou quando pouca ou nenhuma evidência acerca do tema é conhecida (KITCHENHAM; CHARTERS, 2007). O interesse aqui é identificar a quantidade de soluções que tratam sobre a mesma temática, servindo como base de estudo tanto para esta pesquisa quanto para futuras investigações.

O mapeamento será realizado em três etapas: planejamento, execução e análise de resultados. Cada etapa será detalhada em suas respectivas seções neste capítulo.

2.3.1 Planejamento

O planejamento da pesquisa envolve a definição das questões de pesquisa, a identificação das bases de dados que serão utilizadas, a formulação da string de busca e a definição dos critérios de inclusão e exclusão dos estudos (KITCHENHAM; CHARTERS, 2007).

As questões de pesquisa orientam a busca por estudos relevantes e ajudam a definir os critérios de inclusão e exclusão. As questões de pesquisa para este estudo são:

QP1: Quais são as soluções tecnológicas existentes que auxiliam na reciclagem ou que envolvem a existência de um aplicativo que auxilia na reciclagem?

QP2: Quais tecnologias são utilizadas nessas soluções?

QP3: Em que contexto essas soluções foram desenvolvidas (acadêmico ou empresarial)?

QP4: Como essas soluções foram avaliadas?

As bases de dados a serem utilizadas para a busca de estudos são:

- ACM Digital Library
- SCOPUS
- Google Scholar
- SOL (<https://sol.sbc.org.br>).

A string de busca é uma combinação de palavras-chave e operadores booleanos que é usada para buscar estudos nas bases de dados. Ela é uma ferramenta essencial na

realização de um mapeamento sistemático, pois permite filtrar os estudos relevantes para a pesquisa de forma eficiente e precisa. A string de busca da proposta para este estudo é:

("recycling app" OR "app for recycling") AND (solution OR software OR tool OR methodology)

Esta string de busca foi elaborada com o objetivo de selecionar artigos que tratam de tecnologia que auxilia na reciclagem ou de aplicativos que auxiliam na reciclagem. As palavras-chave *"recycling app"* e *"app for recycling"* foram escolhidas para capturar a maior quantidade possível de estudos que se enquadram nesse critério. O operador booleano *OR* é usado para incluir estudos que contêm qualquer uma dessas palavras-chave.

A segunda parte da string de busca, *"(solution OR software OR tool OR methodology)"*, foi elaborada para selecionar entre esses artigos aqueles que mencionam explicitamente como a solução foi implementada, usando algum tipo de software, ferramenta ou metodologia. O operador booleano *AND* é usado para garantir que apenas os estudos que atendem a ambos os critérios sejam selecionados.

Os critérios de inclusão são usados para selecionar os estudos relevantes para a pesquisa, para este estudo são:

- **CI1:** O estudo deve ter a data de publicação posterior ao ano de 2020.
- **CI2:** O estudo deve ser sobre uma solução tecnológica que auxilia na reciclagem ou que envolve a existência de um aplicativo que auxilia na reciclagem.
- **CI3:** Deve ser possível visualizar a interface da solução ou conhecer suas funcionalidades.

Também foi selecionado o seguinte critério de exclusão para a seleção dos estudos:

- **CE1:** O estudo não deve ter seu acesso público nem mediante autorização de acessos universitários.

2.3.2 Execução

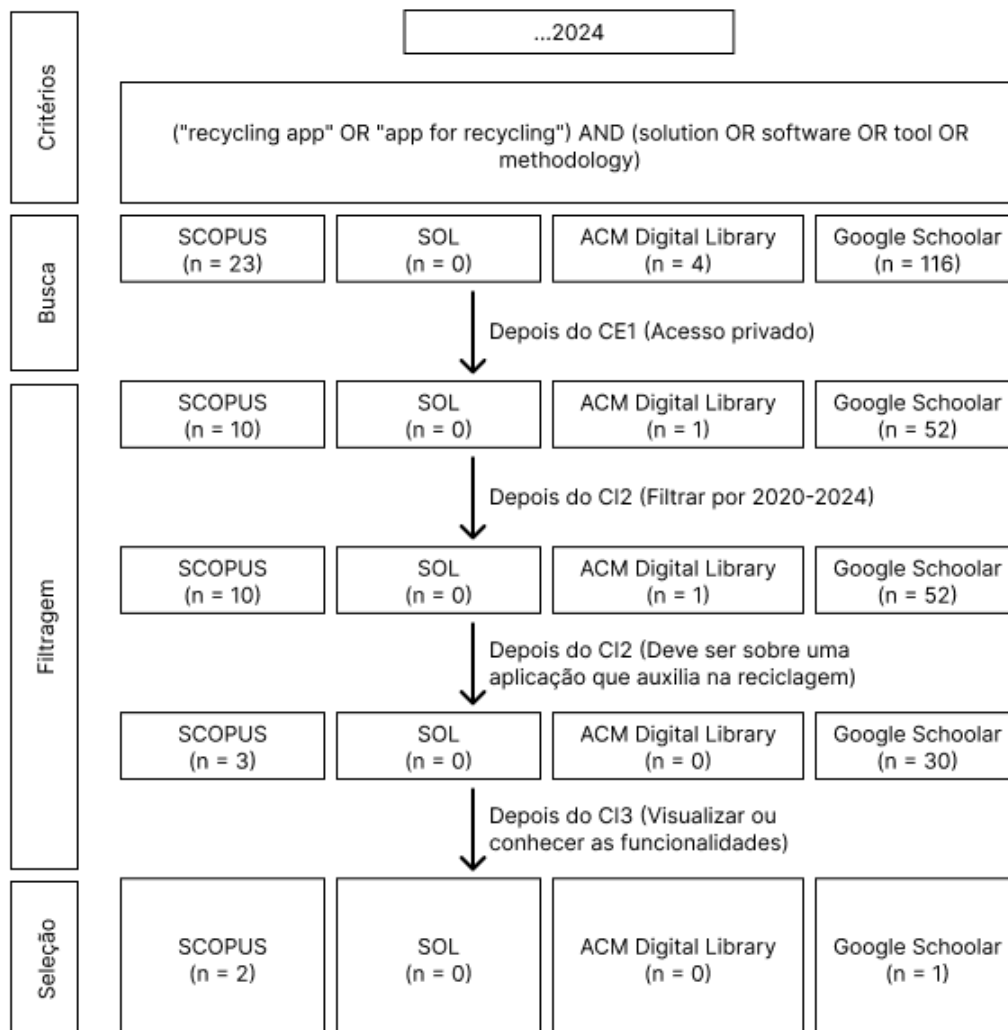
Para a execução da pesquisa, foram utilizadas as seguintes bases de dados:

- Google Scholar
- ACM Digital Library
- SOL
- Scopus

A string de busca foi executada em 10/06/2024, resultando em um total de 143 estudos. Após a aplicação dos critérios de inclusão e exclusão, 3 estudos foram incluídos para a análise final.

A Figura 1 abaixo destaca o número de estudos incluídos de cada base de dados.

Figura 1 – Processo de execução da string de busca



Fonte: Autor (2024)

Os resultados obtidos foram analisados de acordo com os critérios de inclusão estabelecidos. No Google Scholar, foi encontrado 1 trabalho relevante para a pesquisa. Na base de dados SCOPUS, foram encontrados 2 trabalhos relevantes. No entanto, nas bases de dados ACM e SOL, não foram encontrados trabalhos que atendessem aos critérios de inclusão.

Esses artigos fornecem insights valiosos sobre como a tecnologia pode ser usada para facilitar a reciclagem e a gestão de resíduos. Destaca-se também a importância de considerar fatores como a facilidade de uso e a utilidade percebida ao desenvolver essas soluções.

2.3.3 Resultado

Segue a análise detalhada dos resultados obtidos após a execução da pesquisa e a aplicação dos critérios de inclusão e exclusão, organizada conforme as questões de pesquisa definidas na fase de planejamento.

2.3.3.1 QP1: Quais são as soluções tecnológicas existentes que auxiliam na reciclagem ou que envolvem a existência de um aplicativo que auxilia na reciclagem?

As soluções tecnológicas existentes que auxiliam na reciclagem ou que envolvem a existência de um aplicativo que auxilia na reciclagem se diferem por artigo:

1. **"Novel Mobile Application System for Implementation of an Eco-Incentive Scheme"**, de Huang et al. (2022), apresenta um sistema de aplicativo móvel desenvolvido para implementar um esquema de eco-incentivo. O sistema premia os consumidores com eco-créditos quando reciclam e reutilizam produtos no fim de vida, que podem ser usados para descontos em compras, trocados por ingressos de museu/teatro ou doados para o plantio de árvores. Um estudo de caso demonstrou que o aplicativo móvel para eco-incentivo pode apoiar eficientemente os consumidores na reciclagem de produtos e no gerenciamento de seus registros de recompensas.
2. **"CIUDAD LIMPIA VALDIVIA: A MOBILE AND WEB BASED SMART SOLUTION BASED ON FOSS TECHNOLOGY TO SUPPORT MUNICIPAL AND HOUSEHOLD WASTE COLLECTION"**, de Sierra et al. (2021), descreve uma aplicação Web e Móvel chamada Ciudad Limpia Valdivia, desenvolvida para abordar alguns dos problemas de coleta de resíduos do Município de Valdivia no Chile. A ferramenta, baseada em Software Livre e de Código Aberto, visa melhorar a comunicação entre os usuários e o provedor de serviços, além de fornecer materiais educativos sobre gestão de resíduos e outras questões ambientais relacionadas. As principais características da solução incluem: um visualizador de mapa dinâmico mostrando a posição dos caminhões de coleta de resíduos do município e trajetórias futuras estimadas, um formulário para relatar locais de descarte ilegal, notificações do município sobre o status do serviço, conteúdo educacional e lista de contato de outras iniciativas independentes relacionadas à coleta de resíduos.
3. **"Smart Recycle – Development of a waste collection application"**, de Silva e Branco (2021), aborda o avanço descontrolado de resíduos gerados por indivíduos e empresas, as formas como são descartados na natureza, as ações que o governo tem tomado e como os autores pretendem ajudar a melhorar esses números, aproximando o país dos objetivos estabelecidos pela União Europeia. Como resultado do estudo, sistemas móveis e aplicativos, dotados de uma variedade de recursos de controle e assistentes intuitivos, serão disponibilizados. O propósito dessas ferramentas é facilitar e aprimorar a forma como os resíduos são caracterizados, descartados e reutilizados por diversas entidades organizacionais. O objetivo é diminuir substancialmente o descarte impróprio e minimizar o impacto ambiental, contribuindo para um planeta menos tóxico e mais saudável para as gerações atuais e futuras.

2.3.3.2 QP2: Quais tecnologias são utilizadas nessas soluções?

As tecnologias empregadas nessas soluções são diversas. "CIUDAD LIMPIA VALDIVIA", desenvolvido por Sierra et al. (2021), é baseado em Software Livre e de Código Aberto e utiliza tecnologias de geoespacial e aprendizado de máquina. "Smart Recycle", criado por Silva e Branco (2021), possui uma arquitetura baseada em "computação em

nuvem", com versões de aplicativos para Android e IOS e um sistema WEB fundamentado na linguagem de programação Angular, além de uma "API RestFul". "Novel Mobile Application System for Implementation of an Eco-Incentive Scheme", desenvolvido por Huang et al. (2022), é composto por um servidor de banco de dados, comunicação e integração, e quatro módulos funcionais, empregando diversas técnicas de desenvolvimento móvel.

2.3.3.3 QP3: Em que contexto essas soluções foram desenvolvidas (acadêmico ou empresarial)?

Todas as soluções foram concebidas em um contexto acadêmico. "CIUDAD LIMPIA VALDIVIA", desenvolvida por Sierra et al. (2021), foi criada para tratar problemas de coleta de resíduos no Município de Valdivia no Chile. "Smart Recycle", desenvolvida por Silva e Branco (2021), foi parte de um projeto de mestrado. E "Novel Mobile Application System for Implementation of an Eco-Incentive Scheme", desenvolvido por Huang et al. (2022), foi criado como parte do projeto CIRC4Life, apoiado pelo programa Horizon 2020 da Comissão Europeia.

2.3.3.4 QP4: Como essas soluções foram avaliadas?

A efetividade dessas soluções foi avaliada por meio de estudos de caso e estudos de usabilidade. "CIUDAD LIMPIA VALDIVIA", conforme Sierra et al. (2021), foi avaliada por meio de um estudo de caso realizado no setor de teste piloto da cidade. "Smart Recycle", de acordo com Silva e Branco (2021), foi avaliada por meio de um estudo de usabilidade com 7 usuários com diferentes níveis de experiência. "Novel Mobile Application System for Implementation of an Eco-Incentive Scheme", segundo Huang et al. (2022), foi avaliada por meio de um estudo de caso, demonstrando que o aplicativo móvel para eco-incentivo pode apoiar eficientemente os consumidores na reciclagem de produtos e no gerenciamento de seus registros de recompensas.

2.3.4 Considerações finais

Este capítulo teve como objetivo identificar e analisar soluções tecnológicas existentes que auxiliam na reciclagem ou que envolvem a existência de um aplicativo que auxilia na reciclagem. Foram identificadas três soluções com foco em diferentes aspectos da gestão de resíduos: "CIUDAD LIMPIA VALDIVIA", "Smart Recycle" e "Novel Mobile Application System for Implementation of an Eco-Incentive Scheme". Cada uma dessas soluções apresenta abordagens distintas e utiliza diferentes tecnologias, demonstrando a diversidade de possibilidades para o uso da tecnologia na promoção da reciclagem. Dado essas soluções, este projeto de conclusão de curso propõe o desenvolvimento de um aplicativo que promove a gestão integrada de resíduos urbanos. O aplicativo fornecerá informações sobre as rotas e horários das coletas seletivas, contribuindo para a eficiência e eficácia da coleta de resíduos. Além disso, o aplicativo incluirá marcadores no mapa para indicar a localização das estações de coleta seletiva, facilitando o acesso dos usuários a esses pontos. O diferencial deste projeto é a participação ativa da comunidade, que poderá sugerir novos marcadores, promovendo assim a colaboração e o engajamento dos usuários na gestão de resíduos.

3. Projeto

Esta seção tem como objetivo detalhar todas as decisões tomadas durante o desenvolvimento da aplicação, descrevendo os requisitos funcionais e não funcionais identificados, os casos de uso e seus respectivos detalhamentos, protótipos de algumas interfaces da aplicação, bem como o diagrama de classes e outros artefatos que visam facilitar a compreensão geral da proposta.

3.1 Requisitos

Os requisitos para o projeto em questão serão coletados e analisados com base em aspectos funcionais (RF), não funcionais (RNF) e de negócio (RN), que são detalhados nas Seções 3.1.1, 3.1.2 e 3.1.3, respectivamente.

Ambos os tipos de requisitos são essenciais para o sucesso de qualquer projeto de software. Os requisitos funcionais definem o que o sistema deve fazer, enquanto os requisitos não funcionais descrevem como o sistema deve realizar essas funções. A combinação de ambos é fundamental para garantir que as necessidades e expectativas do projeto sejam atendidas (CASA DO DESENVOLVEDOR, 2023).

Para a elicitação desses requisitos, será empregado uma combinação de técnicas que incluem:

- Estudos Similares: Esta técnica foi empregada para a coleta dos requisitos do projeto, fundamentando-se nas soluções tecnológicas existentes que contribuem para a reciclagem ou que envolvem a presença de um aplicativo que facilita a reciclagem, conforme discutido na Seção 2.3 Soluções Similares.
- Prototipação: Através da prototipação, é possível validar as funcionalidades da aplicação e prever a organização das informações apresentadas. Isso permite a iteração e aprimoramento do design e da usabilidade do sistema antes de sua implementação final.
- Análise de Casos de Uso: Através da análise de casos de uso, torna-se possível compreender a interação dos usuários finais com o sistema em diferentes cenários. Isso permite o desenvolvimento de um sistema que atenda às suas necessidades de forma eficaz. A combinação dessas técnicas possibilita a coleta de um conjunto abrangente de requisitos, podendo fazer com que o sistema final atenda às necessidades dos usuários e aos objetivos do negócio.

3.1.1 Requisitos Funcionais

Para os requisitos funcionais foram priorizadas as ações do usuário final, das quais o objetivo principal é buscar e visualizar informações através do aplicativo. Como mencionado por Sommerville (2011, p.59), "São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer." Com base nisso, segue os requisitos levantados.

RF01: O sistema deve possibilitar que o usuário faça um cadastro na aplicação.

RF02: O sistema deve possibilitar que o usuário faça o login na aplicação.

RF03: O sistema deve possibilitar que os usuários visualizem os pontos de descarte no mapa na página inicial.

RF04: O sistema deve possibilitar que os usuários escolham um ponto marcado no mapa e que através do ponto selecionado seja possível traçar uma rota até o local via Google Maps.

RF05: O sistema deve mostrar as categorias de descarte e os meios de contato de um determinado ponto quando o mesmo for selecionado pelo usuário.

RF06: O sistema deve possibilitar que o usuário adicione um ponto de descarte no mapa com o selo de não verificado.

RF07: O sistema deve permitir ao administrador e aos usuários verificar um ponto de descarte não verificado.

RF08: O sistema deve permitir ao usuário visualizar uma página que mostre as empresas de reciclagem patrocinadoras da aplicação, juntamente com seus meios de contato e os tipos de descarte que compram.

RF09: O sistema deve permitir ao usuário visualizar uma página que mostre os horários em que a coleta normal e seletiva passa em um determinado ponto.

RF10: O sistema deve disponibilizar ao usuário uma página de perfil onde o mesmo possa ver a data de início na aplicação e quando pontos foram verificados pelo usuário.

RF11: O sistema deve permitir ao administrador o cadastro das rotas de coleta normal e seletiva.

RF12: O sistema deve permitir ao administrador o cadastro das empresas patrocinadoras.

RF13: O sistema deve permitir ao administrador excluir um ponto de descarte.

RF14: O sistema irá expurgar os pontos de descarte que passarem de 2 meses após a criação sem 3 verificações através de um temporizador.

3.1.2 Requisitos não Funcionais

Os requisitos não funcionais listados levam em consideração os recursos necessários para a concepção do aplicativo. Como Sommerville (2011, p.59) define, "São restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de timing, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo."

RNF01: O aplicativo deve ser disponibilizado como um site de acesso público.

RNF02: O aplicativo deve ser desenvolvido utilizando Angular para o front-end e Java para o back-end, considerando o tempo hábil para o desenvolvimento do projeto.

RNF03: O aplicativo deve ser compatível com os navegadores atuais e ser disponibilizado como um Progressive Web App (PWA).

RNF04: Os dados do aplicativo devem ser persistidos em um banco de dados PostgreSQL.

RNF05: O aplicativo deve incluir um botão de compartilhamento que permita aos usuários compartilhar a localização de um ponto de descarte através de um link.

3.1.3 Regra de Negócio

Com base em Amoasei (2023), as regras de negócio são essenciais para definir e limitar a estrutura e o comportamento de um negócio. No contexto do aplicativo em questão, essas regras estão intimamente ligadas à interação do usuário com o aplicativo e à maneira como as informações são apresentadas e gerenciadas.

RN01: O cadastro deve ser bem sucedido apenas se preencher todos os campos de cadastro.

RN02: As categorias para os pontos de descarte são: Material Orgânico, Papel e Papelão, Plásticos, Vidros, Metais, Eletrônicos, Móveis Grandes, Óleos, Roupas.

RN03: Os meios de contato com as empresas de reciclagem podem ser: e-mail, telefone, whatsapp.

RN04: Um ponto de descarte se torna 'verificado' quando três usuários diferentes confirmam a localização do ponto de descarte ou quando um administrador verifica o ponto.

RN05: O usuário precisa estar conectado em sua conta para verificar um ponto de descarte.

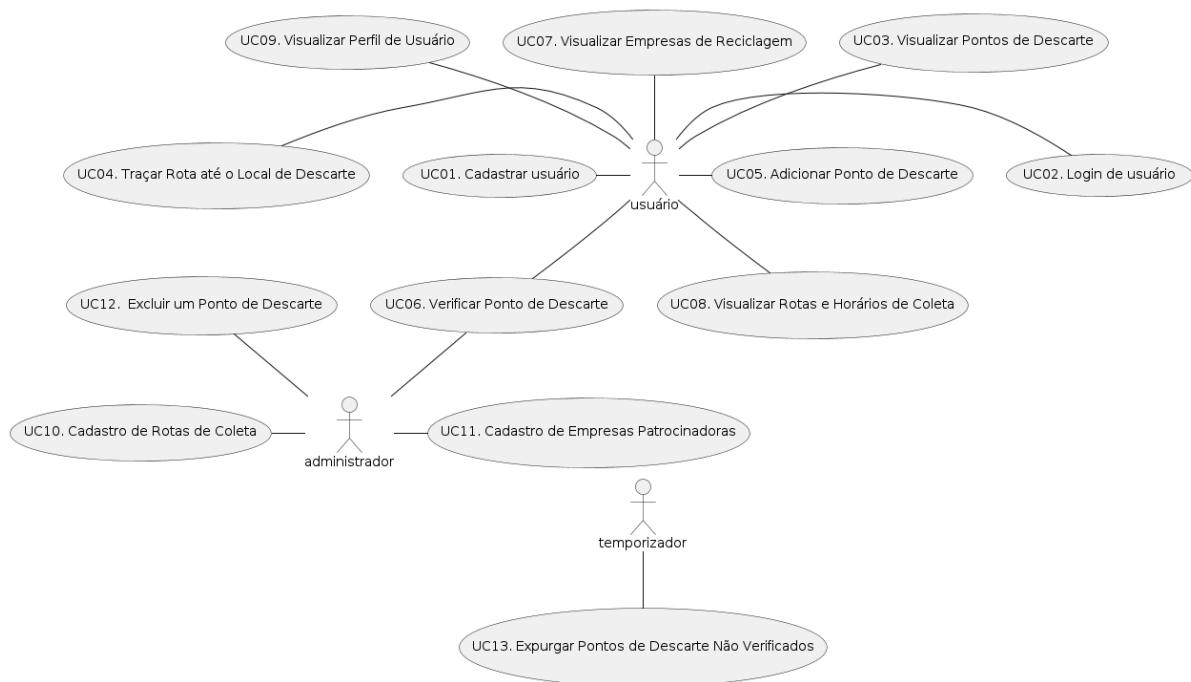
RN06: Os dados do usuário poderão ser visualizados apenas pelo próprio usuário, não sendo compartilhados em nenhum local público da aplicação.

RN07: As opções para o tipo de coleta são: Normal e Reciclável.

3.2 Casos de uso e Protótipos

Essa seção é dedicada à descrição dos casos de uso e protótipos do sistema. Casos de uso, conforme descrito por Alff (2023), são uma técnica de modelagem que descreve a interação entre um ator (geralmente um usuário) e um sistema para atingir um objetivo específico. A importância desses casos de uso reside em sua capacidade de fornecer uma visão clara e concisa do que o sistema fará e como será utilizado, garantindo que os requisitos do sistema estejam claramente definidos e que as expectativas do usuário sejam atendidas. Já os protótipos são versões iniciais do sistema que são criadas para testar e validar ideias de design antes da implementação final.

Figura 2 – Diagrama de caso de uso



Fonte: Autor (2024)

UC01: Cadastro de Usuário

- Ator: Usuário
- Pré-condição: O usuário deve estar na página de cadastro.
- Fluxo Principal: O usuário preenche os campos necessários e submete o formulário. O sistema valida as informações e, se estiverem corretas, cria uma nova conta de usuário.
- Rastreabilidade: RF01, RN01
- Protótipo: Apêndice A, Figura 21

UC02: Login de Usuário

- Ator: Usuário
- Pré-condição: O usuário deve estar na página de login.
- Fluxo Principal: O usuário insere suas credenciais e submete o formulário. O sistema valida as credenciais e, se estiverem corretas, autentica o usuário.
- Rastreabilidade: RF02
- Protótipo: Apêndice A, Figura 22

UC03: Visualizar Pontos de Descarte

- Ator: Usuário
- Pré-condição: O usuário deve estar na página inicial.
- Fluxo Principal: O usuário clica na opção para carregar os pontos de descarte e seleciona as categorias. O sistema exibe os pontos de descarte no mapa.
- Rastreabilidade: RF03, RN04, RN05
- Protótipo: Apêndice A, Figura 23

UC04: Traçar Rota até o Local de Descarte

- Ator: Usuário
- Pré-condição: O usuário deve ter selecionado um ponto de descarte.
- Fluxo Principal: O usuário seleciona um pin e em seguida a opção para traçar uma rota até o local de descarte. O sistema abre o Google Maps e traça a rota até o local de descarte.
- Rastreabilidade: RF04
- Protótipo: Apêndice A, Figura 24

UC05: Adicionar Ponto de Descarte

- Ator: Usuário
- Pré-condição: O usuário deve estar autenticado.
- Fluxo Principal: O usuário seleciona a opção para adicionar um novo ponto de descarte. O usuário preenche as informações necessárias e submete o formulário. O sistema valida as informações e, se estiverem corretas, adiciona o novo ponto de descarte no mapa como 'não verificado'.
- Rastreabilidade: RF06, RN02
- Protótipo: Apêndice A, Figura 25

UC06: Verificar Ponto de Descarte

- Ator: Usuário
- Pré-condição: O usuário deve estar autenticado e o ponto de descarte deve estar marcado como 'não verificado'.
- Fluxo Principal: O usuário seleciona um ponto de descarte e confirma a localização. O sistema verifica se este é o terceiro usuário a confirmar a localização. Se for, o sistema marca o ponto de descarte como 'verificado'.
- Rastreabilidade: RF07, RN05, RN02, RN04
- Protótipo: Apêndice A, Figura 26

UC07: Visualizar Empresas de Reciclagem

- Ator: Usuário
- Pré-condição: O usuário deve estar na página inicial.
- Fluxo Principal: O usuário acessa a seção "Empresas de Reciclagem". O sistema exibe as informações de contato público das empresas patrocinadoras e as categorias de resíduos que desejam comprar.
- Rastreabilidade: RF08
- Protótipo: Apêndice A, Figura 27

UC08: Visualizar Rotas e Horários de Coleta

- Ator: Usuário
- Pré-condição: O usuário deve estar na página inicial.

- Fluxo Principal: O usuário acessa a opção para visualizar as rotas e horários de coleta. Filtra por nome de rua e/ou tipo de coleta. O sistema exibe as rotas dos caminhões de coleta e os horários em que operam.
- Rastreabilidade: RF09, RN07
- Protótipo: Apêndice A, Figura 28

UC09: Visualizar Perfil de Usuário

- Ator: Usuário
- Pré-condição: O usuário deve estar autenticado.
- Fluxo Principal: O usuário acessa a opção para visualizar seu perfil. O sistema exibe as informações do perfil do usuário.
- Rastreabilidade: RF10, RN06
- Protótipo: Apêndice A, Figura 29

UC10: Cadastro de Rotas de Coleta

- Ator: Administrador
- Pré-condição: O administrador deve estar autenticado.
- Fluxo Principal: O administrador acessa a opção para cadastrar novas rotas de coleta. O administrador preenche as informações necessárias e submete o formulário. O sistema valida as informações e, se estiverem corretas, adiciona a nova rota de coleta.
- Rastreabilidade: RF11
- Protótipo: Apêndice A, Figura 30

UC11: Cadastro de Empresas Patrocinadoras

- Ator: Administrador
- Pré-condição: O administrador deve estar autenticado.
- Fluxo Principal: O administrador acessa a opção para cadastrar novas empresas patrocinadoras. O administrador preenche as informações necessárias e submete o formulário. O sistema valida as informações e, se estiverem corretas, adiciona a nova empresa patrocinadora.
- Rastreabilidade: RF12
- Protótipo: Apêndice A, Figura 31

UC12: Excluir um Ponto de Descarte

- Ator: Administrador
- Pré-condição: O administrador deve estar autenticado.
- Fluxo Principal: O administrador seleciona um ponto de descarte. O administrador clica no botão de excluir. O sistema pede uma confirmação e, se for confirmada, será excluído o ponto de descarte.
- Rastreabilidade: RF13
- Protótipo: Apêndice A, Figura 32

UC13: Expurgar Pontos de Descarte Não Verificados

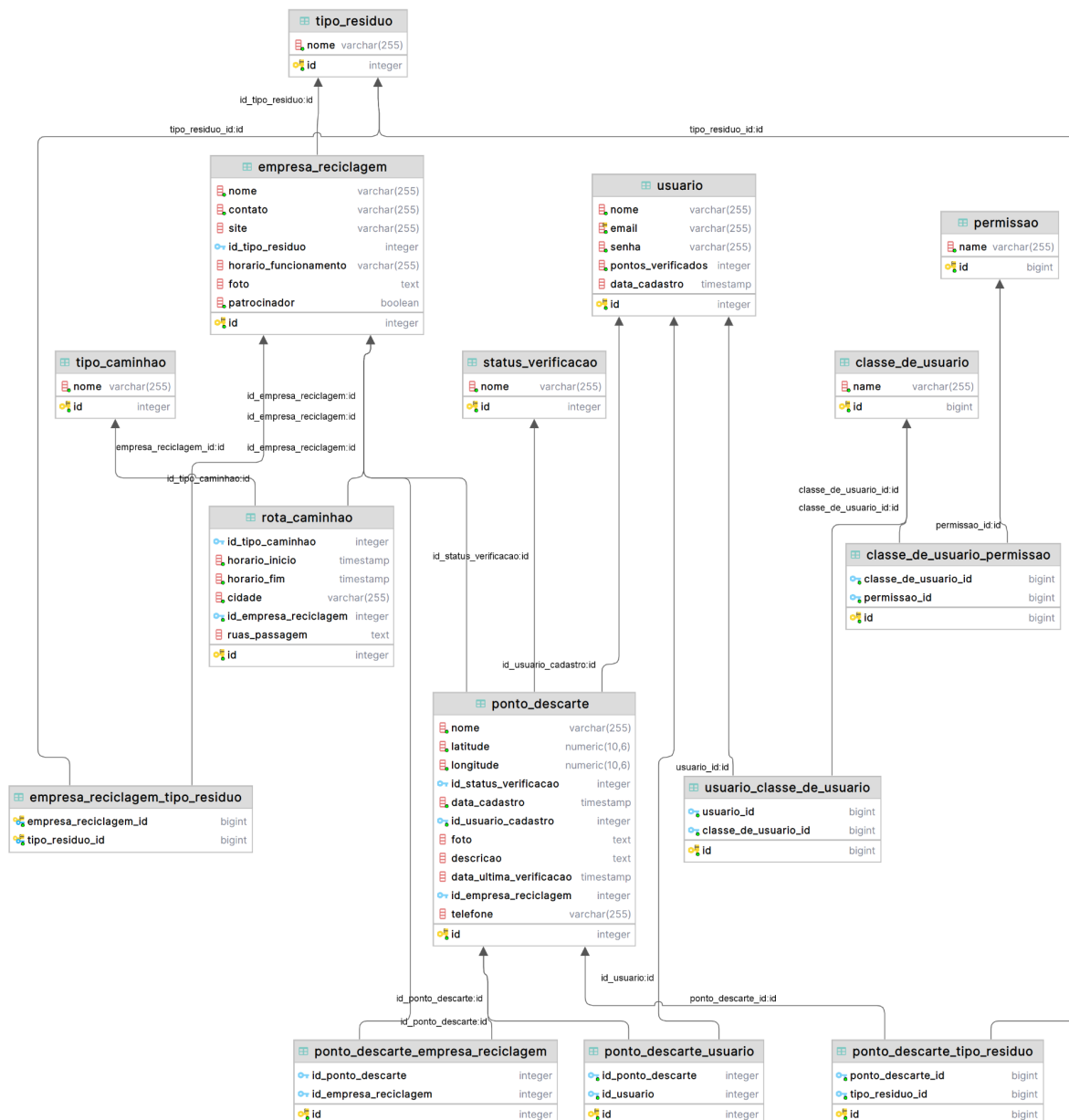
- Ator: Temporizador
- Pré-condição: O ponto de descarte deve ter sido criado há mais de 2 meses e ter menos de 3 verificações.

- Fluxo Principal: O sistema, através de um temporizador, irá verificar periodicamente os pontos de descarte. Se um ponto de descarte foi criado há mais de 2 meses e tem menos de 3 verificações, o sistema automaticamente irá expurgar (remover) o ponto de descarte.
- Rastreabilidade: RF14

3.3 Diagramas de Entidade e Relacionamento

A técnica de modelagem de dados é empregada para organizar e documentar a representação dos dados em um sistema. O Modelo Entidade-Relacionamento é um dos diagramas utilizados para essa finalidade, onde os dados são representados por entidades e suas relações (LI; CHEN, 2009). A Figura 15 mostra o diagrama escolhido para este estudo.

Figura 3 – Modelo Entidade-Relacionamento



--
Fonte: Autor (2024)

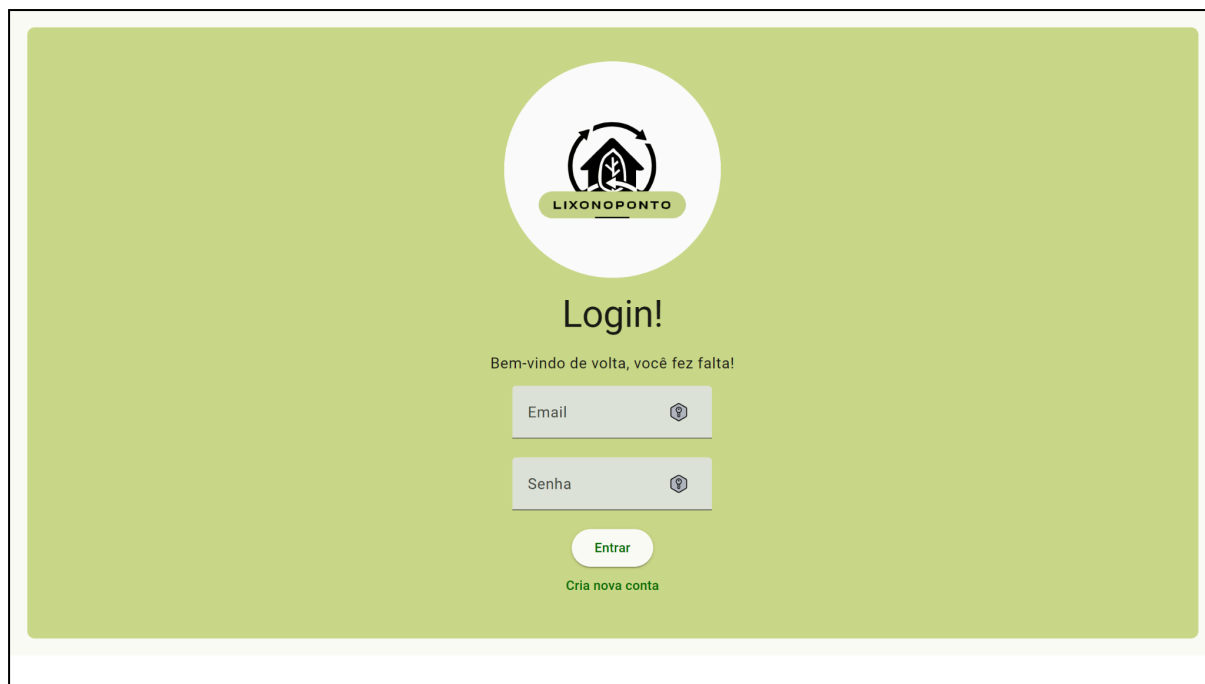
4. Operacionalidade

Conforme definido na seção 3.1.1 e com base nas tecnologias escolhidas na seção 2, o sistema, desenvolvido utilizando o *framework* Angular para o front-end e Java com Spring Boot para o back-end, possui funcionalidades específicas para dois perfis de usuários: comum e administrador. Ambos os perfis compartilham algumas funcionalidades, que são descritas na seção 4.1. As funcionalidades específicas para cada perfil são descritas nas seções 4.2 e 4.3.

4.1 Funcionalidades Comuns

Na Figura 4, é possível observar a tela de login onde qualquer usuário tem a possibilidade de inserir suas credenciais para acessar a aplicação (RF02, UC02). Caso o usuário ainda não possua uma conta, a opção "Criar nova conta" pode ser selecionada para direcioná-lo à página de registro.

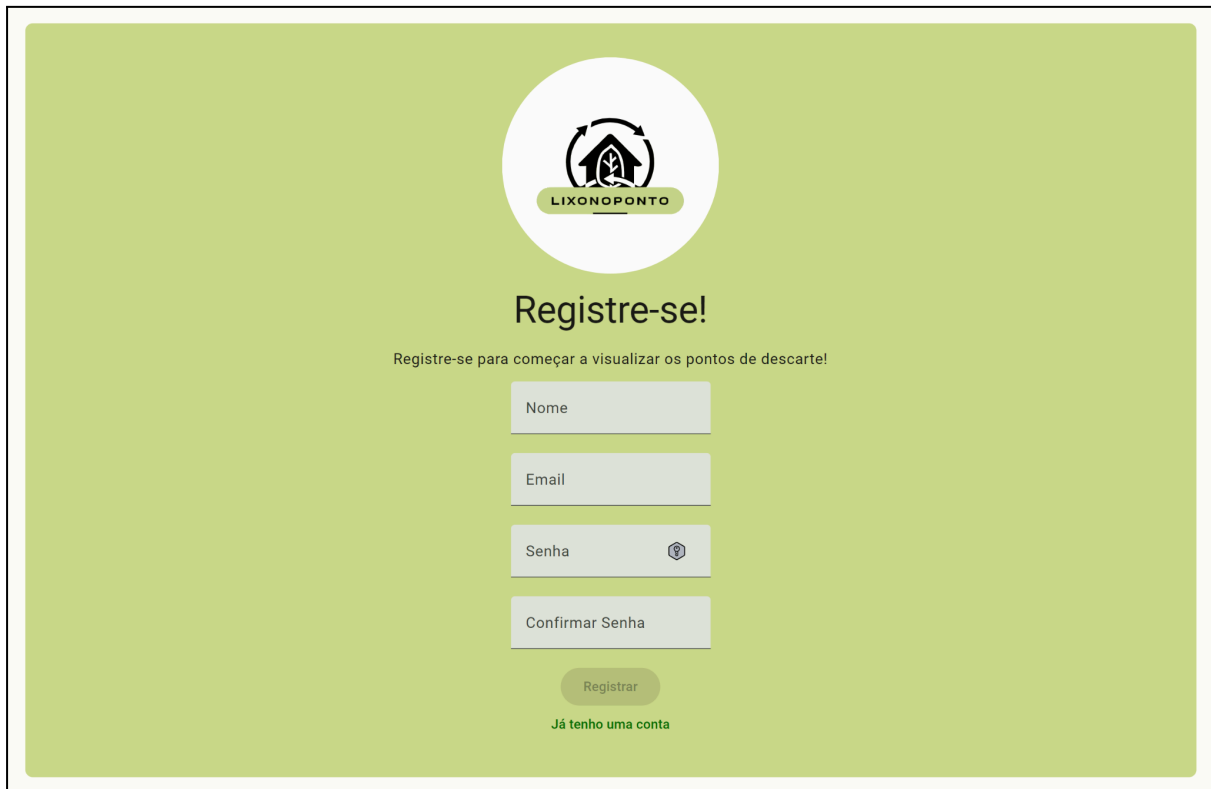
Figura 4 - Tela de Login



Fonte: Autor (2024)

A tela de registro é apresentada na Figura 5. Neste ambiente, qualquer usuário pode preencher os campos necessários para criar uma nova conta na aplicação (RF01, UC01).

Figura 5 - Tela de Registro



A tela de registro do aplicativo LIXONOPONTO apresenta um fundo verde claro. No topo central, há um ícone circular branco com um símbolo de casa e setas, e o nome 'LIXONOPONTO' em um retângulo verde abaixo dele. Abaixo do ícone, o texto 'Registre-se!' é exibido em uma fonte grande e preta. Segue-se a instrução 'Registre-se para começar a visualizar os pontos de descarte!'. O formulário de registro contém quatro campos de entrada: 'Nome', 'Email', 'Senha' (com um ícone de olho para alternar a visibilidade) e 'Confirmar Senha'. Abaixo dos campos, há um botão verde arredondado com o texto 'Registrar' e um link verde 'Já tenho uma conta'.

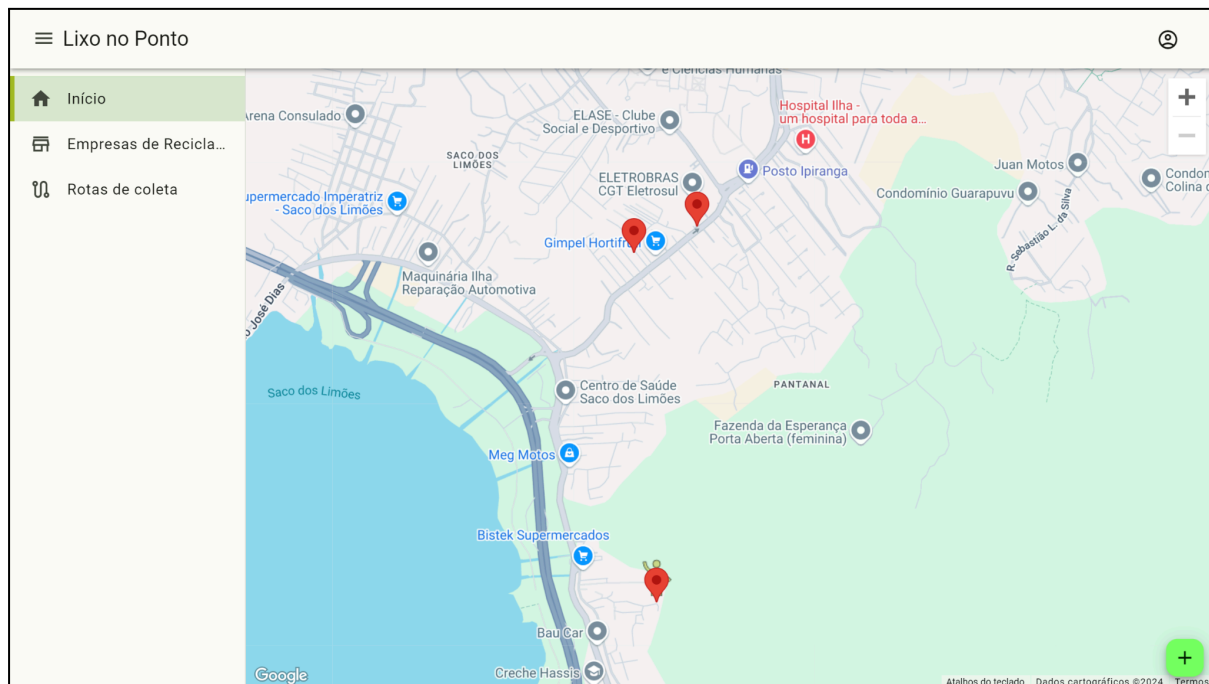
Fonte: Autor (2024)

Após o login, o usuário terá acesso a um menu lateral à esquerda e um menu horizontal no topo, independentemente do perfil. No topo, o usuário pode expandir o menu, acessar seu perfil ou realizar o *logout* da aplicação.

4.2 Funcionalidades do Perfil Usuário

A página inicial para o perfil de usuário é ilustrada na Figura 6. Nesta página, o usuário pode visualizar um mapa com pontos de descarte disponíveis (RF03, UC03). Adicionalmente, um novo ponto de descarte pode ser adicionado ao clicar no botão inferior direito (RF06, UC05).

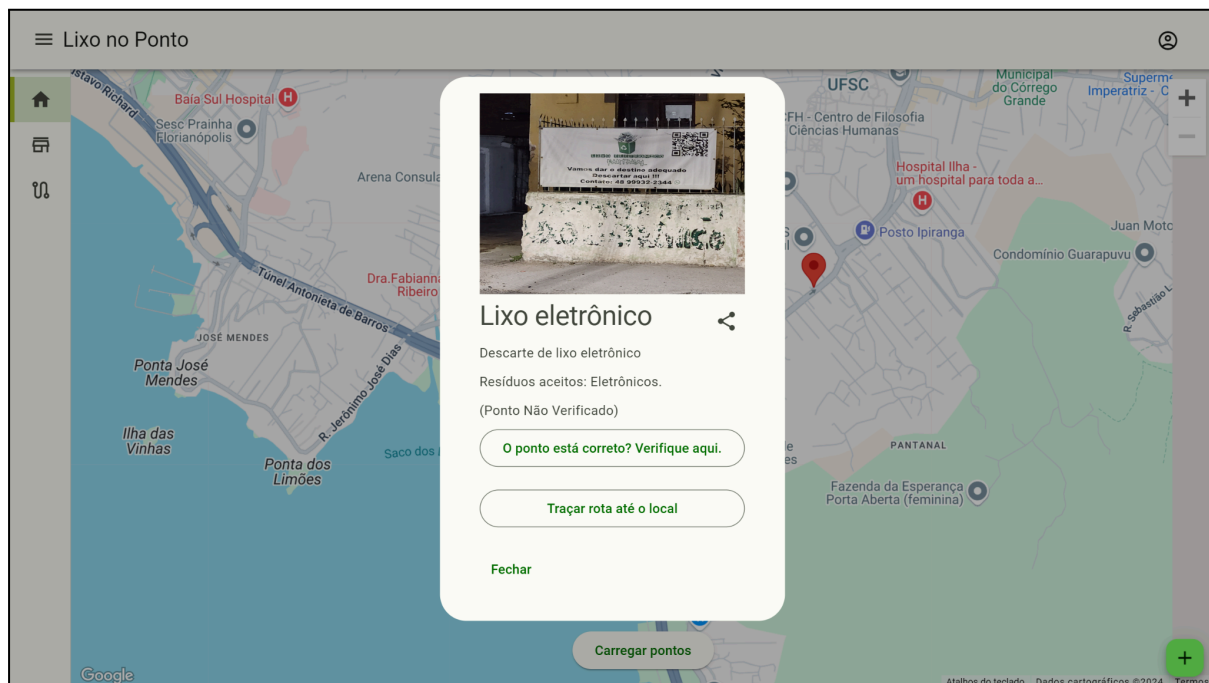
Figura 6 - Página Inicial do Usuário



Fonte: Autor (2024)

Ao selecionar um ponto de descarte, um modal com informações detalhadas sobre o ponto é apresentado, conforme ilustrado na Figura 7. Neste modal, o usuário tem a opção de copiar o link da localização do ponto, acessar informações do ponto, verificar um ponto (RF07, UC06) e traçar uma rota até o local do ponto (RF04, UC04).

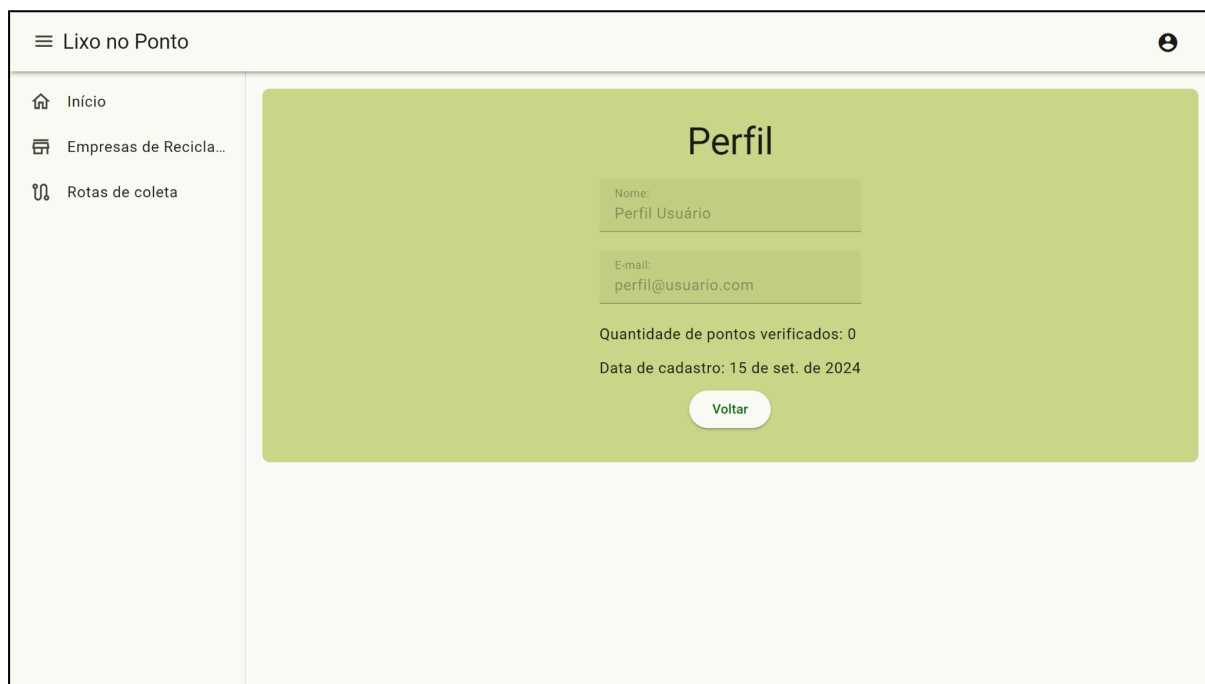
Figura 7 - Modal de Informações do Ponto de Descarte



Fonte: Autor (2024)

Na Figura 8, é possível visualizar a página de perfil do usuário. Aqui, o usuário pode visualizar as informações do seu perfil (RF10, UC09).

Figura 8 - Página de Perfil do Usuário



Fonte: Autor (2024)

A página de Empresas de Reciclagem é apresentada na Figura 9. Nesta página, o usuário pode ver uma lista de empresas de reciclagem da sua cidade cadastradas na aplicação, bem como cada resíduo que a empresa coleta e suas informações de contato (RF08, UC07).

Figura 9 - Página de Empresas de Reciclagem



Fonte: Autor (2024)

A Figura 10 mostra a página de Rotas e Horários de Coleta de Lixo. Nesta página, o usuário pode ver uma lista de ruas da sua cidade cadastradas, cada item é uma rota por onde uma coleta, seja seletiva ou comum, irá passar, assim como o horário de início do trajeto do caminhão (RF09, UC08).

Figura 10 - Página de Rotas e Horários de Coleta de Lixo

The screenshot displays the 'Rotas e Horários de Coleta de Lixo' page. At the top, there is a navigation bar with 'Lixo no Ponto' and a user profile icon. Below the navigation bar, the page title 'Rotas e Horários de Coleta de Lixo' is centered. There are two filter boxes: 'Filtrar por nome da rua' and 'Tipo de coleta'. The main content area lists eight collection routes, each with a circular icon indicating the type of collection (Seletiva or Comum), the route name, and the collection time.

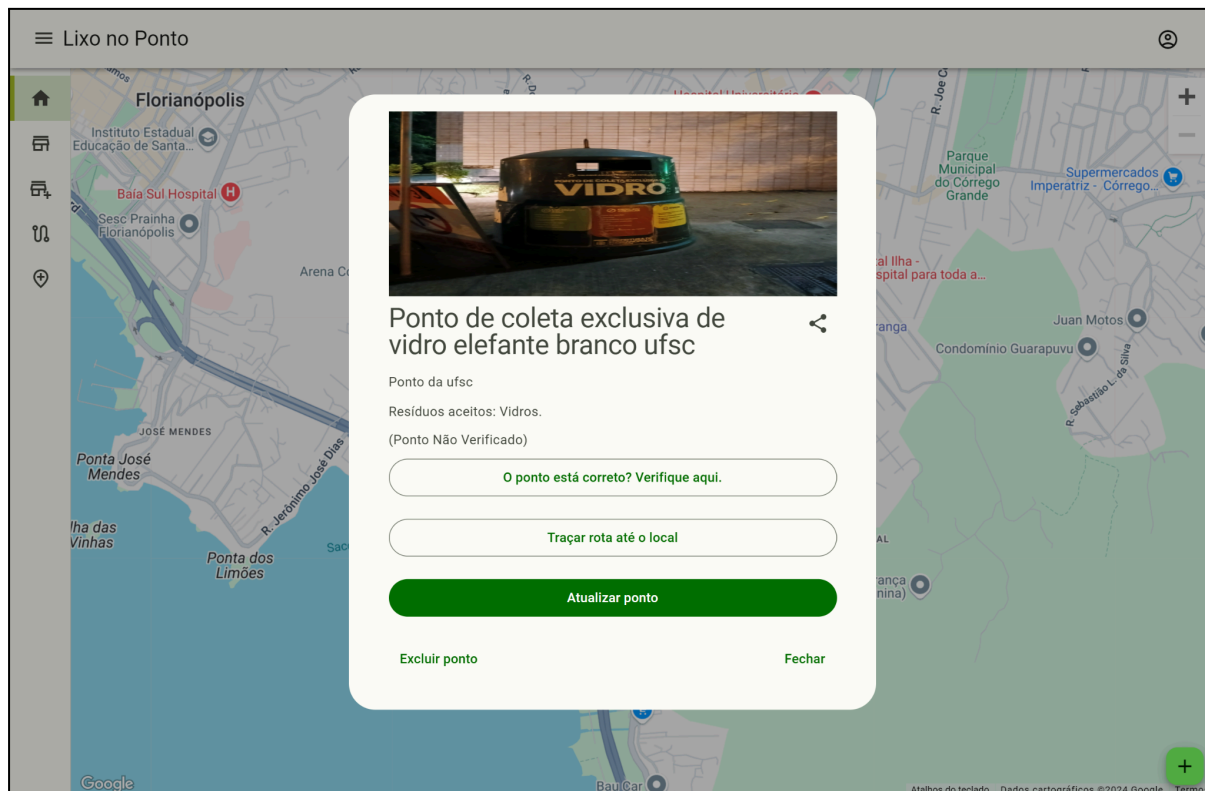
Tipo de Coleta	Nome da Rota	Horário da Rota
Coleta Seletiva	Delminda Silveira	20:00 - 22:00
Coleta Seletiva	Frei Caneca	20:00 - 22:00
Coleta Seletiva	transversais da Beira Mar	20:00 - 22:00
Coleta Seletiva	Geral	20:00 - 22:00
Coleta Seletiva	José Mendes	20:00 - 22:00
Coleta Seletiva	Servidões	20:00 - 22:00
Coleta Comum	Balneário do Estreito e Canto	20:00 - 22:00
Coleta Comum	Fátima	07:00 - 12:00

Fonte: Autor (2024)

4.3 Funcionalidades do Perfil Administrador

A Figura 11 mostra a página de informações de um ponto de descarte para o perfil de administrador. Além das funcionalidades disponíveis para o perfil de usuário, o administrador pode atualizar um ponto de descarte e excluir um ponto de descarte (RF13, UC12).

Figura 11 - Página Ponto de Descarte do Administrador



Fonte: Autor (2024)

A página de Cadastro de Empresas de Reciclagem é apresentada na Figura 12. Nesta página, o administrador pode inserir uma nova empresa de reciclagem (RF12, UC11).

Figura 12 - Página de Cadastro de Empresas de Reciclagem

Fonte: Autor (2024)

A Figura 13 mostra a página de Cadastro de Rotas e Horários de Coleta de Lixo. Nesta página, o administrador pode inserir uma nova rota de coleta de lixo (RF11, UC10).

Figura 13 - Página de Cadastro de Rotas e Horários de Coleta de Lixo

A interface de usuário para o cadastro de rotas de coleta de lixo apresenta o seguinte layout:

- Menu Lateral:**
 - Início
 - Empresas de Recicla...
 - Cadastro de Empres...
 - Rotas de coleta
 - Cadastro de Rotas d...** (destacado)
- Conteúdo Principal:**
 - Título:** Cadastrar rotas de coleta
 - Cidade*:** Campo de texto obrigatório.
 - Tipo do Caminhão*:** Menu suspenso obrigatório.
 - Ruas de Passagem*:** Campo de texto obrigatório.
 - Nota:** Separar as ruas utilizado o caracter ','
 - Horário Início*:** Campo de texto obrigatório com máscara de hora.
 - Formato:** 00:00
 - Horário Fim*:** Campo de texto obrigatório com máscara de hora.
 - Formato:** 00:00
 - Empresa Relacionada*:** Menu suspenso obrigatório.
 - Botão:** Cadastro

Fonte: Autor (2024)

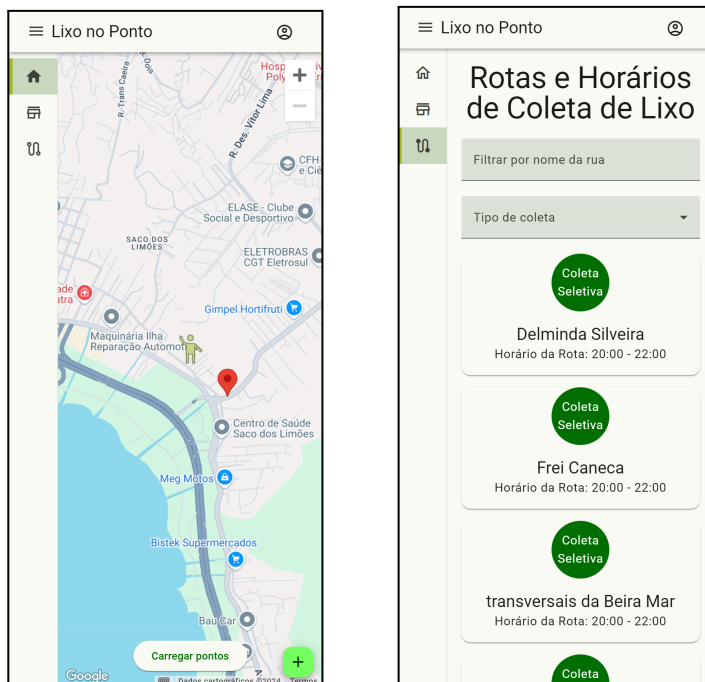
4.4 Funcionalidades Invisíveis

A UC13 é uma funcionalidade invisível que ocorre no back-end do sistema. Um agendador (*scheduler*) é configurado para executar uma tarefa de limpeza que remove pontos de descarte que não tiveram mais de 3 verificações após 2 meses de sua criação. Esta funcionalidade garante a atualização e a relevância dos pontos de descarte exibidos para os usuários.

4.5 Flexibilidade

A usabilidade do projeto é ampliada com a introdução de uma aplicação web progressiva (PWA) para dispositivos móveis. As PWAs combinam o melhor das aplicações web e nativas, proporcionando uma experiência de usuário eficiente. Através da PWA, os usuários podem acessar todo o projeto diretamente de seus dispositivos móveis, garantindo uma interface responsiva e funcional, independentemente da plataforma utilizada, conforme a Figura 14a, que mostra a página inicial de busca de pontos de descarte em um iPhone XR, e a Figura 14b, que exibe a página de busca de rotas e horários de coleta de lixo em um iPhone XR.

Figura 14 - (a) Página Inicial e (b) Rotas em dispositivos móveis



Fonte: Autor (2024)

Quando um usuário de dispositivo móvel Android acessa a aplicação, é exibido um *pop-up* que oferece a instalação da aplicação, conforme mostrado na Figura 15. Essa funcionalidade é possível devido à conexão HTTPS presente na aplicação.

Figura 15 - Pop-up de aviso para instalar PWA



Fonte: Autor (2024)

5. Avaliação

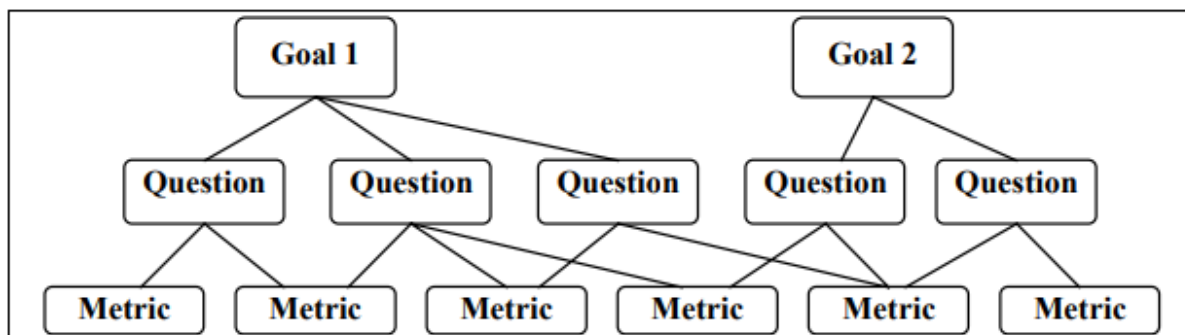
Para avaliar o sistema desenvolvido, foi utilizado a abordagem Goal-Question-Metric (GQM). GQM é uma estratégia de avaliação que permite a definição de metas específicas para o sistema, formula perguntas que interpretam essas metas em termos operacionais e especifica as métricas necessárias para responder a essas perguntas (Basili et al., 1994).

A sigla GQM significa:

- **Goal (Objetivo):** Define o que se espera alcançar com a avaliação. Pode ser relacionado à melhoria de algum aspecto do sistema, como eficiência, eficácia, satisfação do usuário, entre outros.
- **Question (Pergunta):** São questões que buscam interpretar os objetivos de forma que possam ser mensurados. As perguntas são formuladas para cobrir todos os aspectos do objetivo definido.
- **Metric (Métrica):** São as medidas que serão coletadas para responder às perguntas. As métricas devem ser escolhidas de forma a fornecer uma resposta quantitativa para as perguntas (Mandić et al., 2010).

A estrutura do GQM é organizada em três níveis, conforme demonstrado visualmente na Figura 16.

Figura 16 - Exemplo visual da estrutura do GQM.



Fonte: Basili et al. (1994)

A seção de planejamento envolve a definição dos objetivos, perguntas e métricas. Depois, vem a etapa de execução, com a determinação, na seção 5.1, de como, quando e por quem as avaliações serão coletadas, conforme descrito na seção 5.2. E, por fim, temos a etapa 5.3 de análise dos dados coletados. A aplicação da abordagem GQM garante que todos os aspectos relevantes do sistema sejam considerados e avaliados de forma sistemática e estruturada (Fontoura & Price, 2004).

5.1. Planejamento

Neste capítulo, são definidos os objetivos do projeto e, para cada objetivo, as questões que ajudam a avaliar seu cumprimento. Para cada questão, também são definidas

as métricas correspondentes que são utilizadas para medir os resultados. Utiliza-se a Escala Likert para as medições das questões.

A Escala Likert será empregada para quantificar as respostas dos usuários, permitindo uma avaliação precisa e padronizada das percepções e opiniões em relação aos objetivos do projeto. Essa escala é amplamente utilizada em pesquisas para medir atitudes e comportamentos, oferecendo uma gama de opções que variam de "discordo totalmente" a "concordo totalmente", facilitando a análise estatística dos dados coletados (BISCHOFF, 2022).

Tabela 1 - Questões e métricas do Planejamento

Objetivo (G1): Ampliar o conhecimento sobre a prática e políticas de gestão de resíduos urbanos da comunidade em um ambiente urbano.
Q1.1: Quão efetivo é o sistema em informar sobre os pontos de descarte e rotas de coleta de lixo?
M1.1.1: Quantos pontos de descarte você conhecia antes de usar o sistema? M1.1.2: Quantos pontos de descarte você conseguiu identificar após usar o sistema? M1.1.3: Quantas rotas de coleta de lixo você conhecia antes de usar o sistema? M1.1.4: Quantas rotas de coleta de lixo você conseguiu identificar após usar o sistema?
Indicadores: Pontos de descarte conhecidos previamente (M1.1.1) versus pontos de descarte identificados após o uso do sistema (M1.1.2). Rotas de coleta de lixo conhecidas previamente (M1.1.3) versus rotas de lixo identificadas após o uso do sistema (M1.1.4). Para todos os indicadores é satisfatório que o usuário tenha descoberto pelo menos uma entidade a mais do que já conhecia.
Q1.2: Qual o grau de compreensão das informações sobre os pontos de descarte e rotas de coleta de lixo?
M1.2.1: As informações sobre os pontos de descarte são de fácil compreensão. M1.2.2: As informações sobre os pontos de descarte são adequadas (suficientes, claras e corretas). Resposta na escala Likert: (1) Discordo totalmente; (2) Discordo parcialmente; (3) Indiferente; (4) Concordo parcialmente; (5) Concordo totalmente M1.2.3: As informações sobre as rotas de coleta de lixo são de fácil compreensão. M1.2.4: As informações sobre as rotas de coleta de lixo são adequadas (suficientes, claras e corretas). Resposta na escala Likert: (1) Discordo totalmente; (2) Discordo parcialmente; (3) Indiferente; (4) Concordo parcialmente; (5) Concordo totalmente
Indicadores: Quantidade de participantes que concordaram totalmente ou parcialmente com a M1.2.1, M1.2.2, M1.2.3 e M1.2.4.
Objetivo (G2): Avaliar a usabilidade do projeto Lixo no Ponto em relação à experiência do usuário e à eficiência do sistema, do ponto de vista dos usuários finais, no contexto de um ambiente urbano.
Q2.1: Qual a pontuação na escala SUS obtida pelo sistema?

M2.1.1: Eu acho que gostaria de usar esse sistema com frequência.
 M2.1.2: Eu acho o sistema desnecessariamente complexo.
 M2.1.3: Eu achei o sistema fácil de usar.
 M2.1.4: Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.
 M2.1.5: Eu acho que as várias funções do sistema estão muito bem integradas.
 M2.1.6: Eu acho que o sistema apresenta muita inconsistência.
 M2.1.7: Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.
 M2.1.8: Eu achei o sistema atrapalhado de usar.
 M2.1.9: Eu me senti confiante ao usar o sistema.
 M2.1.10: Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Resposta na escala Likert: (1) Discordo totalmente; (2) Discordo parcialmente; (3) Indiferente; (4) Concordo parcialmente; (5) Concordo totalmente

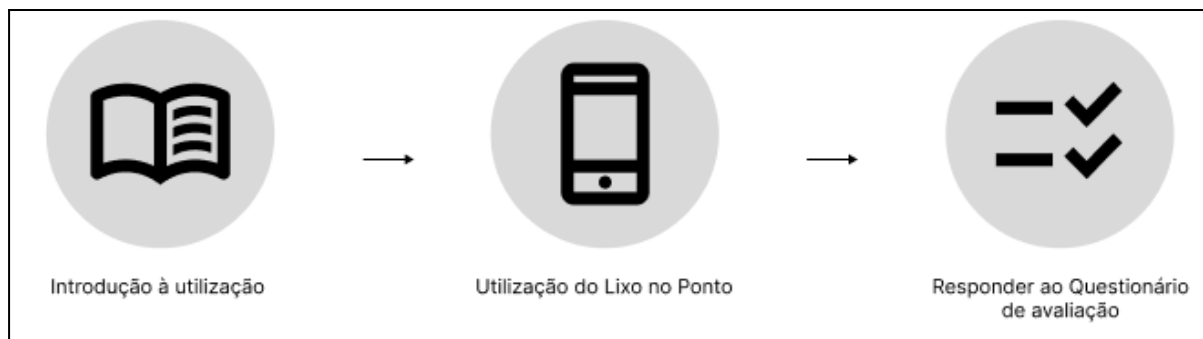
Indicadores:

Utilizado a interpretação de que a média SUS acima de 70 pontos é considerada usabilidade aceitável e abaixo de 50 pontos é considerada inaceitável. Entre 50 e 70 pontos é considerada marginalmente aceitável.

5.2. Execução

A execução da avaliação seguiu os passos descritos na Figura 17.

Figura 17 - Passo a passo da etapa de execução



Fonte: Autor (2024)

Na primeira etapa, a introdução à utilização, foi fornecido aos usuários um documento detalhado, disponível no Apêndice B. Este documento orienta os usuários a realizarem determinadas tarefas, que estão listadas na Tabela 2. Essas tarefas têm como objetivo familiarizar os usuários com as funcionalidades do aplicativo Lixo no Ponto.

Tabela 2 - Tarefas Pré Avaliação

Lista de Tarefas
Tarefa 1. Realizar o cadastro de sua conta
Tarefa 2. Realizar a entrada na sua conta
Tarefa 3. Buscar pontos de descarte no mapa

Tarefa 4. Cadastrar um ponto de descarte real/fictício com preferência por um real
Tarefa 5. Visualizar seu próprio perfil
Tarefa 6. Visualizar as empresas de reciclagem
Tarefa 7. Visualizar as rotas de coleta de lixo
Tarefa 8. Filtrar rotas de coleta de lixo seja por rua ou por tipo de coleta

Na segunda etapa, os usuários utilizam o aplicativo Lixo no Ponto para realizar as tarefas especificadas. Esta etapa é crucial para que os usuários possam experimentar as funcionalidades do aplicativo em um contexto real, permitindo uma avaliação mais precisa da satisfação em relação ao sistema.

Por fim, na terceira etapa, os usuários respondem ao questionário, cujas perguntas estão descritas na Seção 5.1 e o formulário completo está disponível no Apêndice C. A coleta de dados é realizada através de um formulário do Google Forms, permitindo a visualização de todos os resultados, sejam eles gerais ou individuais por usuário. Além da coleta dos dados das perguntas, são coletados dados relacionados ao perfil do usuário, incluindo nome, idade, sexo, plataforma utilizada (web, mobile) e experiência com aplicações (pouca, média, muita). Esta etapa permite a coleta de dados quantitativos e qualitativos sobre a experiência dos usuários, fornecendo insights para a avaliação do projeto.

O perfil dos participantes da avaliação, composta por 16 pessoas, revela que 37,5% são do sexo feminino e 62,5% do sexo masculino. A maioria dos usuários, 62,5%, utilizaram dispositivos móveis para realizar a avaliação, enquanto 37,5% utilizaram computador ou notebook. Em relação à experiência com aplicativos, 81,2% dos usuários relataram ter muita experiência, enquanto 18,8% têm experiência mediana. Esses dados são fundamentais para entender o perfil dos usuários.

5.3. Análise dos Resultados

Esta seção apresenta uma análise detalhada dos resultados obtidos a partir da avaliação do sistema. São discutidos a efetividade do sistema em informar sobre os pontos de descarte e rotas de coleta de lixo, o grau de compreensão das informações pelos usuários e a pontuação na escala SUS obtida pelo sistema.

5.3.1. Quão efetivo é o sistema em informar sobre os pontos de descarte e rotas de coleta de lixo? (Q1.1)

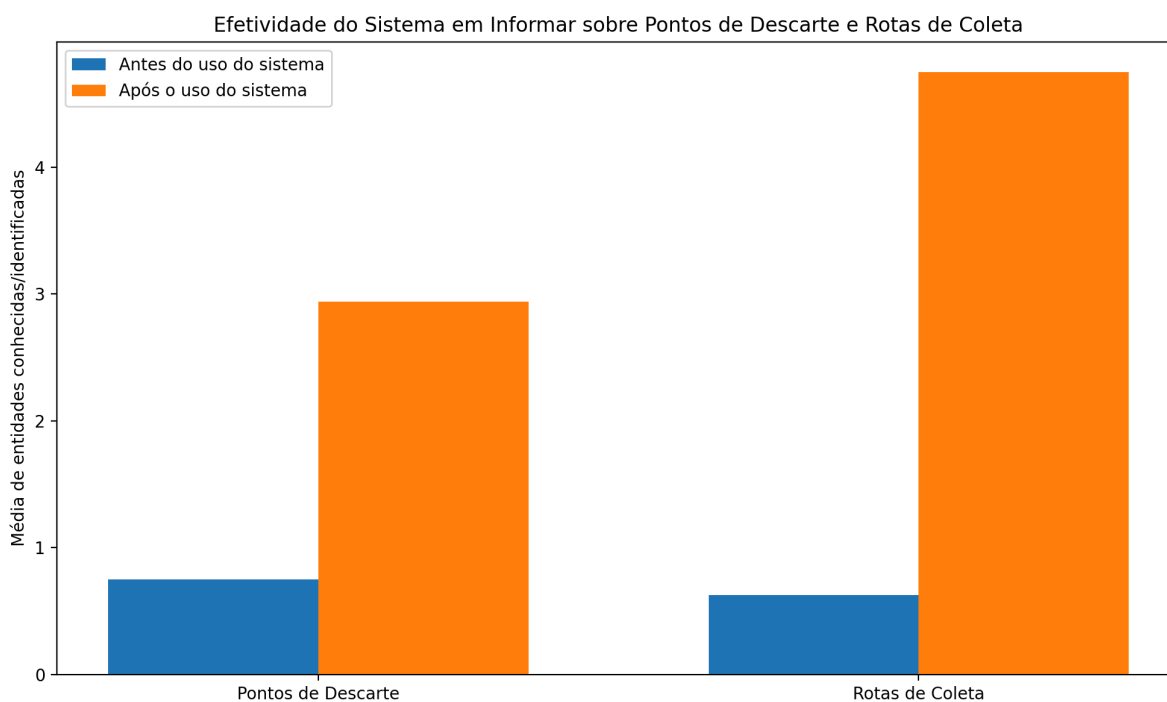
A análise dos dados coletados para a questão Q1.1, que investiga a efetividade do sistema em informar sobre os pontos de descarte e rotas de coleta de lixo, revela resultados significativos. Os indicadores utilizados para esta análise são: pontos de descarte conhecidos previamente (M1.1.1) versus pontos de descarte identificados após o uso do

sistema (M1.1.2) e rotas de coleta de lixo conhecidas previamente (M1.1.3) versus rotas de coleta de lixo identificadas após o uso do sistema (M1.1.4).

Os dados obtidos mostram que a média de pontos de descarte conhecidos antes do uso do sistema é de 0,75, enquanto a média de pontos de descarte identificados após o uso do sistema é de 2,94. Este aumento significativo indica que o sistema é altamente efetivo em informar os usuários sobre novos pontos de descarte. A porcentagem de usuários que descobriram pelo menos um novo ponto de descarte é de 93,75%, o que reforça a eficácia do sistema nesta dimensão.

Em relação às rotas de coleta de lixo, a média de rotas conhecidas antes do uso do sistema é de 0,62, enquanto a média de rotas identificadas após o uso do sistema é de 4,75. Este aumento substancial demonstra que o sistema também é muito eficaz em informar os usuários sobre novas rotas de coleta de lixo. A totalidade dos usuários (100%) descobriu pelo menos uma nova rota de coleta, o que evidencia a eficiência do sistema nesta área.

Figura 18 - Relação entre Pontos de Descartes e Rotas de Coletas Conhecidos Antes e Depois do Uso da Aplicação



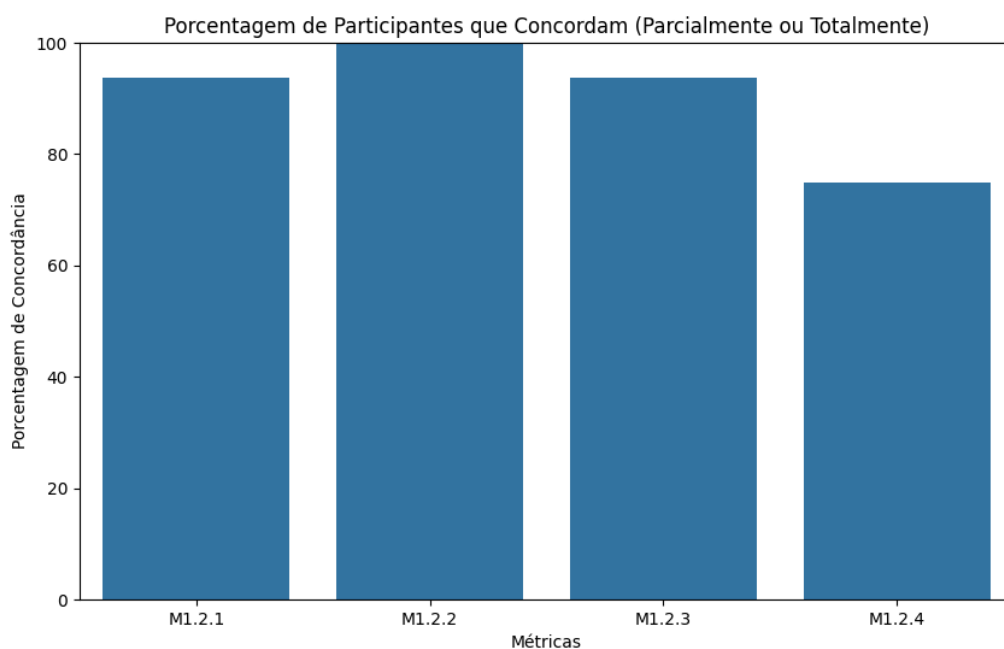
Fonte: Autor (2024)

Conforme a figura acima, os resultados indicam que o sistema cumpre satisfatoriamente seu objetivo de informar os usuários sobre pontos de descarte e rotas de coleta de lixo, com a maioria dos usuários descobrindo novas entidades após a utilização do sistema. A análise quantitativa dos dados, representada graficamente, corrobora a alta efetividade do sistema em ambas as dimensões avaliadas.

5.3.2. Qual o grau de compreensão das informações sobre os pontos de descarte e rotas de coleta de lixo? (Q1.2)

A análise dos dados da questão Q1.2, que avalia como as pessoas entendem as informações sobre os pontos de descarte e rotas de coleta de lixo, mostra resultados interessantes. A métrica M1.2.1, que verifica se as informações sobre os pontos de descarte são fáceis de entender, indica que 93,75% dos participantes concordam totalmente ou parcialmente com essa afirmação. Já a métrica M1.2.2, que avalia se as informações sobre os pontos de descarte são adequadas (suficientes, claras e corretas), mostra um índice de concordância de 100%. Isso sugere que a maioria dos usuários acha as informações apresentadas fáceis de entender e adequadas, reforçando que o sistema comunica essas informações de maneira adequada.

Figura 19 - Grau de Compreensão das Informações Sobre os Pontos de Descarte e Rotas de Coleta de Lixo



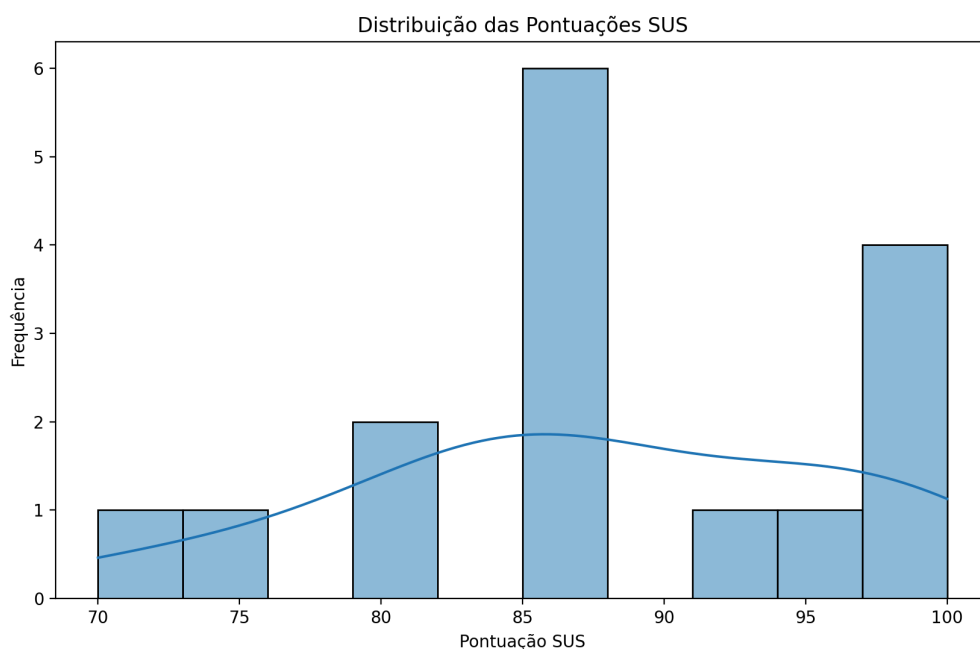
Fonte: Autor (2024)

Ao analisar as rotas de coleta de lixo, observa-se que a métrica M1.2.3, que verifica se as informações são fáceis de entender, indica que 93,75% dos participantes concordam totalmente ou parcialmente. A métrica M1.2.4, que avalia se as informações sobre as rotas de coleta de lixo são adequadas (suficientes, claras e corretas), apresenta um índice de concordância de 75%. Embora a maioria dos participantes considere as informações adequadas, esse resultado sugere que ainda há espaço para melhorar a clareza e suficiência das informações fornecidas sobre as rotas de coleta. A Figura 19, localizada no meio do texto, mostra a porcentagem de participantes que concordam (parcialmente ou totalmente) com as métricas passadas, destacando o impacto positivo do sistema na disseminação de informações sobre gestão de resíduos.

5.3.3. Qual a pontuação na escala SUS obtida pelo sistema? (Q2.1)

O sistema obteve uma pontuação média SUS de 87,50, com um desvio padrão de 8,71. Esta pontuação SUS é considerada excelente quando está acima de 80, com pontuações nessa faixa geralmente interpretadas como indicativas de um sistema com usabilidade muito boa. Segundo a pesquisa realizada por MeasuringU (2022), "um SUS score acima de 80.3 é classificado como 'A', indicando que os usuários estão propensos a recomendar o sistema". Além disso, pontuação acima de 68 são frequentemente vistos como acima da média, sugerindo uma usabilidade satisfatória (UIUX Trend, 2022).

Figura 20 - Pontuação SUS da Avaliação



Fonte: Autor (2024)

A distribuição das pontuações mostra que a pontuação mínima foi de 70,0 e a máxima de 100,0, com 50% das pontuações (mediana) acima de 86,25. Estes resultados sugerem que os usuários consideram o sistema altamente utilizável e satisfatório em termos de experiência do usuário.

5.4. Ameaças ao Estudo

A avaliação do sistema Lixo no Ponto é bem estruturada, contudo existem algumas ameaças que podem afetar a validade dos resultados. Aqui estão as principais ameaças identificadas:

- **Viés de Seleção:** A amostra de participantes pode não ser representativa da população geral de usuários do sistema. A maioria dos participantes relatou

ter muita experiência com aplicativos, o que pode não refletir a experiência do usuário médio.

- **Generalização dos Resultados:** Os resultados obtidos podem não ser generalizáveis para outros contextos ou populações, especialmente considerando que a maioria dos participantes utilizou dispositivos móveis.
- **Instrumentos de Medição:** A utilização de questionários e escalas de Likert pode introduzir vieses de resposta, como a tendência dos participantes a escolherem respostas intermediárias ou extremas.
- **Interpretação das Métricas:** A interpretação das métricas pode variar entre os participantes, especialmente em relação às questões subjetivas de usabilidade e satisfação.
- **Tamanho da Amostra:** A amostra de 16 participantes pode ser insuficiente para garantir a robustez estatística dos resultados, limitando a capacidade de generalização das conclusões.

Identificar e mitigar essas ameaças é crucial para garantir que os resultados da avaliação do sistema Lixo no Ponto sejam válidos e confiáveis.

6. Conclusões

Conclui-se que o desenvolvimento da aplicação PWA para a gestão de resíduos urbanos atingiu plenamente os objetivos propostos. A aplicação, acessível em dispositivos móveis e desktops, foi projetada utilizando Angular para o front-end e Java com Spring Boot para o back-end, tendo sua usabilidade considerada adequada pelos usuários que a avaliaram. A definição dos requisitos funcionais e não funcionais, seguida pela elaboração de casos de uso e protótipos, permitiu a implementação de um sistema Produto Mínimo Viável (MVP)², com funcionalidades mínimas para usuários comuns e administradores, incluindo cadastro e verificação de pontos de descarte, visualização de rotas de coleta e gerenciamento de empresas de reciclagem.

A avaliação do sistema foi realizada por meio da abordagem Goal-Question-Metric (GQM), que possibilitou a definição de objetivos específicos, formulação de perguntas e coleta de métricas. A análise dos resultados obtidos através de questionários aplicados aos usuários revelou avaliações positivas, indicando que a maioria dos participantes considerou o sistema fácil de usar, com informações claras e adequadas, além de funcionalidades bem integradas. Tais resultados confirmam que o sistema desenvolvido atende às necessidades dos usuários e contribui para a gestão eficiente de resíduos urbanos, demonstrando o sucesso do projeto.

6.1 Trabalhos Futuros

Para aprimorar a aplicação e expandir seu impacto, sugerem-se os seguintes trabalhos futuros: (i) estabelecer parcerias com cooperativas que realizam a separação dos resíduos e os encaminham para os locais corretos, o que pode melhorar a eficiência do

² Refere-se a uma versão simplificada de um produto, criada com o mínimo de esforço e recursos, com o objetivo de testar a ideia no mercado e obter feedback dos usuários.

sistema e garantir que os resíduos sejam tratados de maneira adequada; (ii) buscar o apoio da prefeitura para manter o projeto em funcionamento e para cadastrar novos pontos de descarte, incluindo a designação de um gerente interno responsável por coordenar a criação e garantir a atualização contínua dos pontos de coleta; (iii) adicionar novas funcionalidades ao sistema, como notificações em tempo real sobre a coleta de resíduos; (iv) continuar a realizar testes de usabilidade com os usuários para identificar áreas de melhoria e garantir que a aplicação permaneça intuitiva e fácil de usar; e (v) desenvolver campanhas de conscientização para incentivar a população a utilizar a aplicação e a participar ativamente da gestão de resíduos urbanos.

Referências bibliográficas

ABRELPE. Panorama dos Resíduos Sólidos no Brasil 2019. São Paulo: ABRELPE, 2020. Disponível em: <https://abrelpe.org.br/download-panorama-2018-2019/>. Acesso em: 25 ago. 2023.

ALURA. ANGULAR: O QUE É, PARA QUE SERVE E UM GUIA PARA INICIAR NO FRAMEWORK JAVASCRIPT. 2024. Disponível em: <https://www.alura.com.br/artigos/angular-js>. Acesso em: 08 abr. 2024.

AMOASEI, Juliana (org.). O que são regras de negócio? 2023. Disponível em: <https://www.alura.com.br/artigos/o-que-sao-regras-de-negocio>. Acesso em: 16 jun. 2024.

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. Goal Question Metric Paradigm. 2. ed. EUA: Wiley, 1994.

AOKI, Y. et al. Horizontal Relation Identification Method to Handle Misalignment of Goals and Strategies Across Organizational Units. IEEE Access, v. 7, p. 89766-89776, 2019. Disponível em: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8752360>. Acesso em: 18 de set. 2024.

BAUMINAS. Má gestão de efluentes: quais são as consequências para o ambiente? 2021. Disponível em: <https://bauminas.com.br/ma-gestao-de-efluentes-quais-sao-as-consequencias-para-o-ambiente/>. Acesso em: 03 set. 2023.

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. The goal question metric approach. Encyclopedia of software engineering, v. 2, p. 528-532, 1994. Disponível em: <https://www.semanticscholar.org/paper/The-Goal-Question-Metric-Approach-Basili-Caldiera/02e65151786574852007ecd007ee270c50470af0>. Acesso em: 18 de set. 2024.

BISCHOFF, Alex (org.). O que é uma escala Likert e como você pode usá-la para entender melhor seu público? 2022. Disponível em: <https://delighted.com/pt-br/blog/likert-scale>. Acesso em: 22 set. 2024.

BRASIL. Lei nº 12.305, de 2 de agosto de 2010. Institui a Política Nacional de Resíduos Sólidos; altera a Lei no 9.605, de 12 de fevereiro de 1998; e dá outras providências. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2007-2010/2010/lei/l12305.htm. Diário Oficial da República Federativa do Brasil, Brasília, DF, 03 ago. 2010. Acesso em: 25 ago. 2023.

BRASIL. Lei nº 6.938, de 31 de agosto de 1981. Dispõe sobre a Política Nacional do Meio Ambiente, seus fins e mecanismos de formulação e aplicação, e dá outras providências. Diário Oficial da República Federativa do Brasil, Brasília, DF, 2 set. 1981. Seção 1, p. 16509.

BROADCOM. Spring Framework Overview. Disponível em: <https://docs.spring.io/spring-framework/reference/overview.html>. Acesso em: 06 jun. 2024.

CASA DO DESENVOLVEDOR. PWA – Progressive Web Apps: o que é e como essa tecnologia pode ajudar sua empresa. 2023. Disponível em: <https://blog.casadodesenvolvedor.com.br/o-que-e-um-progressive-web-app-pwa/>. Acesso em: 08 abr. 2024.

CASA DO DESENVOLVEDOR. Requisitos funcionais e não funcionais: o que são e como identificar? 2023. Disponível em: <https://blog.casadodesenvolvedor.com.br/requisitos-funcionais-e-nao-funcionais/>. Acesso em: 16 jun. 2024.

CONVALE. Inovações Tecnológicas no Tratamento e Reciclagem de Resíduos Sólidos. 2023. Disponível em: <https://convale.ce.gov.br/informa/43/inovacoes-tecnologicas-no-tratamento-e-reciclagem->. Acesso em: 06 jun. 2024.

Coodesh. O que é PostgreSQL? Disponível em: <https://coodesh.com/blog/dicionario/o-que-e-postgresql/>. Acesso em: 08 abr. 2024.

ECYCLE. Resíduos Sólidos Urbanos: o que são e legislação. Disponível em: <https://www.ecycle.com.br/residuos-solidos/>. Acesso em: 03 set. 2023.

ELKINGTON, J. Towards the sustainable corporation: Win-win-win business strategies Fiveable (org.). HTTP (Hypertext Transfer Protocol). Disponível em: <https://library.fiveable.me/key-terms/ap-comp-sci-p/http-hypertext-transfer-protocol>. Acesso em: 08 abr. 2024.
for sustainable development. California Management Review, v.36, n.2, 1994.

FONTOURA, L. M. Usando GQM para gerenciar riscos em projetos de software. 2004. Tese (Doutorado em Ciência da Computação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004. Disponível em: https://www.inf.ufrgs.br/amadeus/download/manzoni_sb2004.pdf. Acesso em: 18 de set. 2024.

ALFF, F.R. Análise de Requisitos. Especificação de casos de uso: 5 passos. 2023. Disponível em: <https://analisederequisitos.com.br/especificacao-de-casos-de-uso-exemplos/>. Acesso em: 16 jun. 2024.

GOOGLE LLC. Angular Material. Disponível em: <https://material.angular.io>. Acesso em: 08 abr. 2024.

GRONER, Loiane. Progressive Web Apps (PWA) e a evolução da Web. 2018. Disponível em: <https://imasters.com.br/front-end/progressive-web-apps-pwa-e-evolucao-da-web>. Acesso em: 08 abr. 2024.

GRUPO INTERAÇÃO. Tecnologia funciona como aliada na reciclagem do alumínio, indica pesquisa. Disponível em: <https://grupo-interacao.com/tecnologia-reciclagem-aluminio/>. Acesso em: 18 mar. 2024.

HUANG, Hua et al. Novel Mobile Application System for Implementation of an Eco-Incentive Scheme. 2022. Disponível em: <https://www.mdpi.com/2071-1050/14/5/3055>. Acesso em: 16 jun. 2024.
ISBN 978-85-7936-108-1

ISOPOR, Mundo. CONSEQUÊNCIAS DO DESCARTE INCORRETO DE RESÍDUOS NO BRASIL. 2022. Disponível em: <https://www.mundoisopor.com.br/sustentabilidade/descarte-incorreto-de-residuos-no-brasil>. Acesso em: 03 set. 2023.

JULIA WILLICH. Produttivo (org.). Gestão de resíduos: o que é, qual o objetivo e quais as vantagens? 2022. Disponível em: <https://www.produttivo.com.br/blog/gestao-de-residuos/>. Acesso em: 09 set. 2023.

KALFELTZ, Jade; SANTOS, Kainã Pacheco. LIXO IMPOSSÍVEL: Gestão de Resíduos em Florianópolis. Orientadora: Tattiana Teixeira. 2021. 38. TCC (Graduação) – Jornalismo, Jornalismo do Centro de Comunicação e Expressão, Universidade Federal de Santa Catarina, Florianópolis. 2021. Disponível em: [https://repositorio.ufsc.br/bitstream/handle/123456789/230080/TCC%20\(2\).pdf?sequence=2](https://repositorio.ufsc.br/bitstream/handle/123456789/230080/TCC%20(2).pdf?sequence=2). Acesso em: 03/09/2023.

KINGHOST. PWA — O Que São E Como Revolucionam A Experiência Do Usuário. 2024. Disponível em: <https://king.host/blog/tecnologia/o-que-e-pwa/>. Acesso em: 06 jun. 2024.

KITCHENHAM, B.; CHARTERS, S. Guidelines for performing Systematic Literature Reviews in Software Engineering. Durham, UK, 2007.

LEONARDO SZIGETHY. Resíduos sólidos urbanos no Brasil: desafios tecnológicos, políticos e econômicos: se cumpridas as determinações previstas na política nacional de resíduos sólidos, o país poderá transformar o problema representado pelos materiais descartados em solução econômica e social, mas são necessários investimentos, em todos os níveis. Se cumpridas as determinações previstas na Política Nacional de Resíduos Sólidos, o país poderá transformar o problema representado pelos materiais descartados em solução econômica e social, mas são necessários investimentos, em todos os níveis. 2021. Disponível em: <https://www.ipea.gov.br/cts/pt/central-de-conteudo/artigos/artigos/217-residuos-solidos-urbanos-no-brasil-desafios-tecnologicos-politicos-e-economicos>. Acesso em: 03 set. 2023.

LI, Qing; CHEN, Yu-Liu. Modeling and Analysis of Enterprise and Information Systems: from requirements to realization. Beijing: Springer, 2009. Disponível em: <https://link.springer.com/book/10.1007/978-3-540-89556-5>. Acesso em: 17 jun. 2024.

MANDIĆ, V. et al. Utilizing GQM+ Strategies for business value analysis: an approach for evaluating business goals. In: ACM-IEEE INTERNATIONAL SYMPOSIUM ON EMPIRICAL

SOFTWARE ENGINEERING AND MEASUREMENT, 2010, Anais... p. 1-10. Disponível em: https://www.researchgate.net/publication/221494968_Utilizing_GQMstrategies_for_business_value_analysis_An_approach_for_evaluating_business_goals. Acesso em: 18 de set. 2024.

MEASURINGU. Measuring usability with the System Usability Scale (SUS). 2022. Disponível em: <https://measuringu.com/sus/>. Acesso em: 8 out. 2024.

MICROSOFT (org.). Overview of Progressive Web Apps (PWAs). 2023. Disponível em: <https://learn.microsoft.com/en-us/microsoft-edge/progressive-web-apps-chromium/>. Acesso em: 08 abr. 2024.

MILHOMEM, LUCIANO (Recife). Gestão dos resíduos sólidos é chave para desenvolvimento sustentável da América Latina. 2022. Disponível em: <https://brasil.un.org/pt-br/175171-gest%C3%A3o-dos-res%C3%ADduos-s%C3%B3lidos-%C3%A9-chave-para-desenvolvimento-sustent%C3%A1vel-da-am%C3%A9rica-latina>. Acesso em: 11 maio 2023.

NASCIMENTO, Victor Fernandez; SOBRAL, Anahi Chimini; ANDRADE, Pedro Ribeiro de; OMETTO, Jean Pierre Henry Balbaud. Evolução e desafios no gerenciamento dos resíduos sólidos urbanos no Brasil. Ambiente e Agua - An Interdisciplinary Journal Of Applied Science, [S.L.], v. 10, n. 4, p. 891-892, 28 out. 2015. Instituto de Pesquisas Ambientais em Bacias Hidrográficas (IPABHi). <http://dx.doi.org/10.4136/ambi-agua.1635>. Disponível em: <https://www.scielo.br/j/ambiagua/a/NrQL6pPNpMRShCvQbKPWDhg/?lang=pt>. Acesso em: 03 set. 2023.

PHOENIXNAP (org.). Definição da World Wide Web (WWW). Disponível em: <https://www.phoenixnap.pt/gloss%C3%A1rio/defini%C3%A7%C3%A3o-de-rede-mundial#:~:text=A%20World%20Wide%20Web%2C%20muitas,hipertexto%20e%20acess%C3%ADveis%20pela%20Internet..> Acesso em: 08 abr. 2024.

PREFEITURA DE FLORIANÓPOLIS. Plano Municipal de Gestão Integrada de Resíduos Sólidos. Florianópolis: Prefeitura Municipal, 2018. Disponível em: <https://www.pmf.sc.gov.br/sistemas/pmgirs/>.

ROCHA, C. M. C.; MOURA JUNIOR, A. M.; MAGALHÃES, K. M. GESTÃO DE RESÍDUOS SÓLIDOS: PERCEPÇÃO AMBIENTAL DE UNIVERSITÁRIOS EM UMA INSTITUIÇÃO DE ENSINO SUPERIOR BRASILEIRA. REMEA - Revista Eletrônica do Mestrado em Educação Ambiental, [S. l.], v. 29, 2012. Disponível em: <https://periodicos.furg.br/remea/article/view/2962>. Acesso em: 3 set. 2023.

ROCKCONTENT (org.). Banco de dados: saiba o que é, os tipos e a importância para o site da sua empresa. Disponível em: <https://rockcontent.com/br/blog/banco-de-dados/>. Acesso em: 08 abr. 2024.

ROCKCONTENT. Experiência do usuário: o guia completo para agradar seu público digital em 2024. 2023. Disponível em: <https://rockcontent.com/br/blog/experiencia-do-usuario/>. Acesso em: 08 abr. 2024.

Secretaria de Meio Ambiente, Infraestrutura e Logística (SEMIL). Resíduos Sólidos. Disponível em: <https://semil.sp.gov.br/residuos-solidos/>. Acesso em: 26 ago. 2023.

Secretaria Nacional de Saneamento (SNS). Diagnóstico do manejo de Resíduos Sólidos Urbanos. 2019. Disponível em: <http://antigo.snis.gov.br/diagnostico-anual-residuos-solidos/diagnostico-do-manejo-de-residuos-solidos-urbanos-2019>. Acesso em: 26 ago. 2023.

Senado Federal. Aumento da produção de lixo no Brasil requer ação coordenada entre governos e cooperativas de catadores. 2021. Disponível em: <https://www12.senado.leg.br/noticias/infomaterias/2021/06/aumento-da-producao-de-lixo-no-brasil-requer-acao-coordenada-entre-governos-e-cooperativas-de-catadores>. Acesso em: 29 ago. 2023.

SILVA, E. L. da; MENEZES, E. M. Metodologia da Pesquisa e Elaboração de Dissertação. 4. ed. Florianópolis: Laboratório de Ensino a Distância da UFSC, 2005.

SOMMERVILLE, I. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

SOUTO, M. O que é Front-end Back-end e Full Stack - aprenda as diferenças entre essas áreas. 2024. Disponível em: <https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end>. Acesso em: 08 abr. 2024.

TECNOBLOG. O QUE É PWA? [PROGRESSIVE WEB APP]. 2023. Disponível em: <https://tecnoblog.net/responde/o-que-e-pwa-progressive-web-app/>. Acesso em: 08 abr. 2024.

UIUX TREND. System Usability Scale: What it is, Calculation + Usage. 2022. Disponível em: <https://www.questionpro.com/blog/system-usability-scale/>. Acesso em: 8 out. 2024.

VERTOWN (org.). 5 Pilares da Gestão de Resíduos: entenda sua importância! 2023. Disponível em: <https://www.vertown.com/blog/gestao-de-residuos/>. Acesso em: 09 set. 2023.

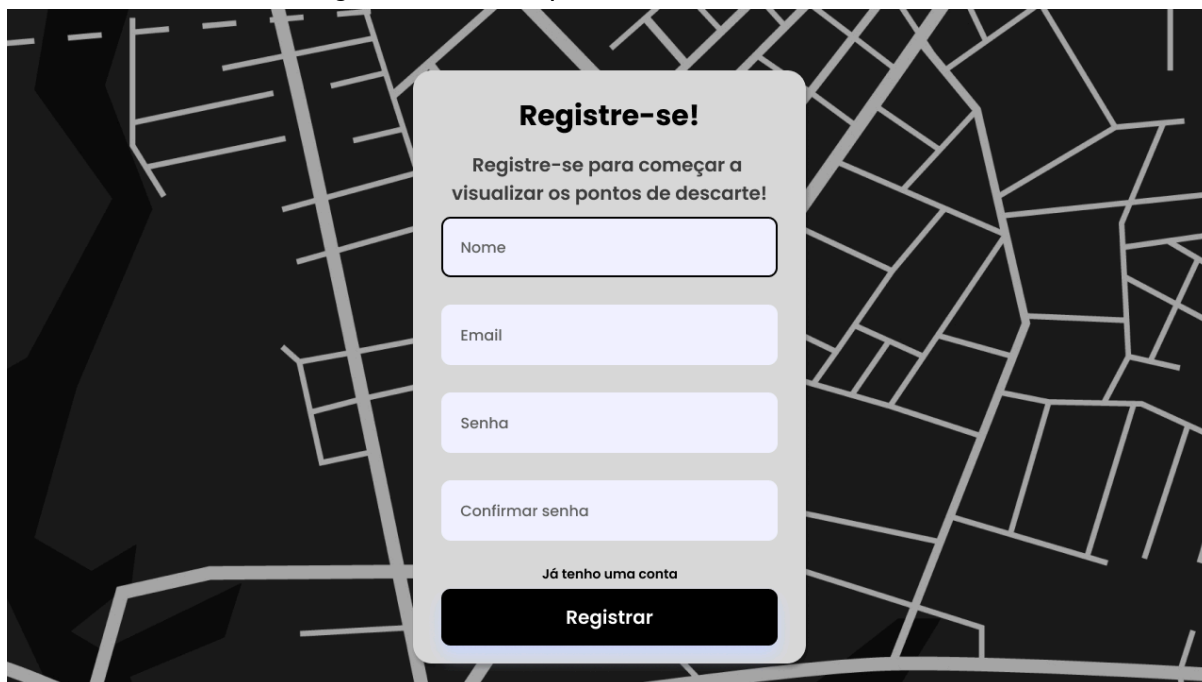
WOJNOWSKA-BARYŁA, I.; BERNAT, K.; ZABOROWSKA, M. Strategies of Recovery and Organic Recycling Used in Textile Waste Management. International Journal of Environmental Research and Public Health, v. 19, n. 10, p. 5859, 11 maio 2022. doi: 10.3390/ijerph19105859. PMID: 35627395; PMCID: PMC9140342.

Anexo(s) e Apêndice(s) (minha autoria):

Apêndice A - Protótipos da Aplicação:

Este apêndice apresenta os protótipos desenvolvidos para a aplicação. Os protótipos foram desenvolvidos considerando as funcionalidades descritas na Seção 4 do documento principal, divididas entre funcionalidades comuns, funcionalidades do perfil de usuário e funcionalidades do perfil de administrador.

Figura 21 – Protótipo de Cadastro de Usuário

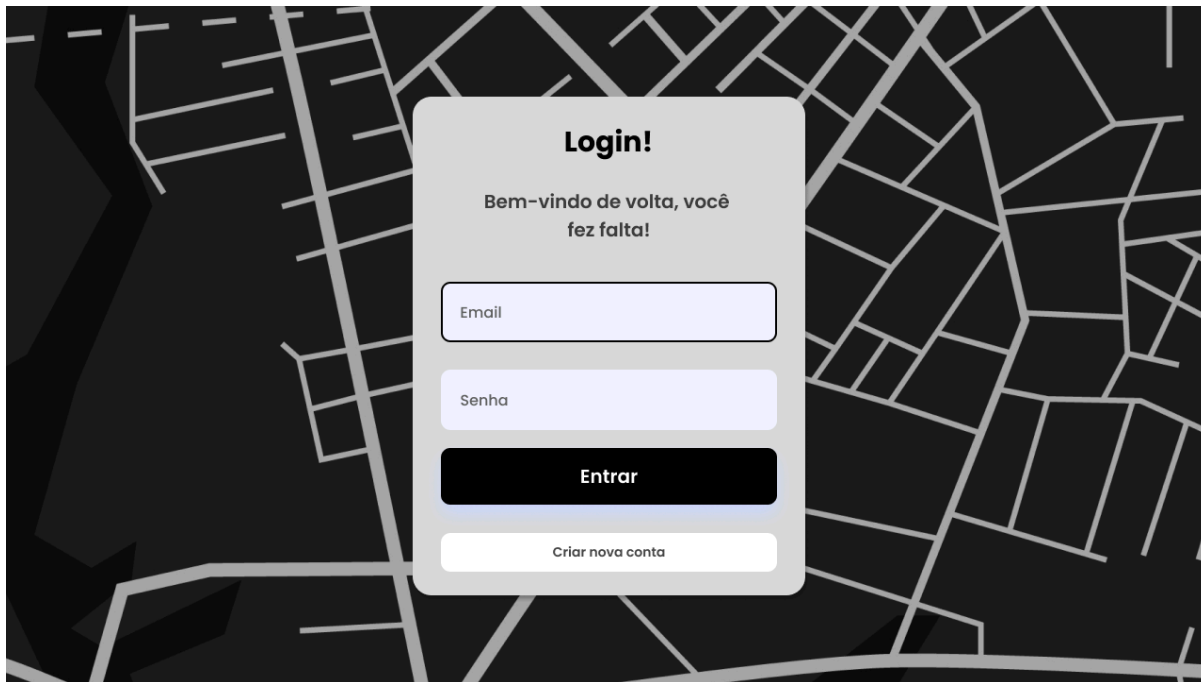


O protótipo de tela de registro de usuário apresenta um formulário centralizado sobre um fundo escuro com um padrão de linhas brancas que simulam um mapa urbano. O formulário é branco e contém o seguinte conteúdo:

- Registre-se!** (título)
- Registre-se para começar a visualizar os pontos de descarte! (mensagem orientadora)
- Nome (campo de texto)
- Email (campo de texto)
- Senha (campo de texto)
- Confirmar senha (campo de texto)
- Já tenho uma conta (link de navegação)
- Registrar (botão de ação)

Fonte: Autor (2024)

Figura 20 – Protótipo de Login de Usuário



Fonte: Autor (2024)

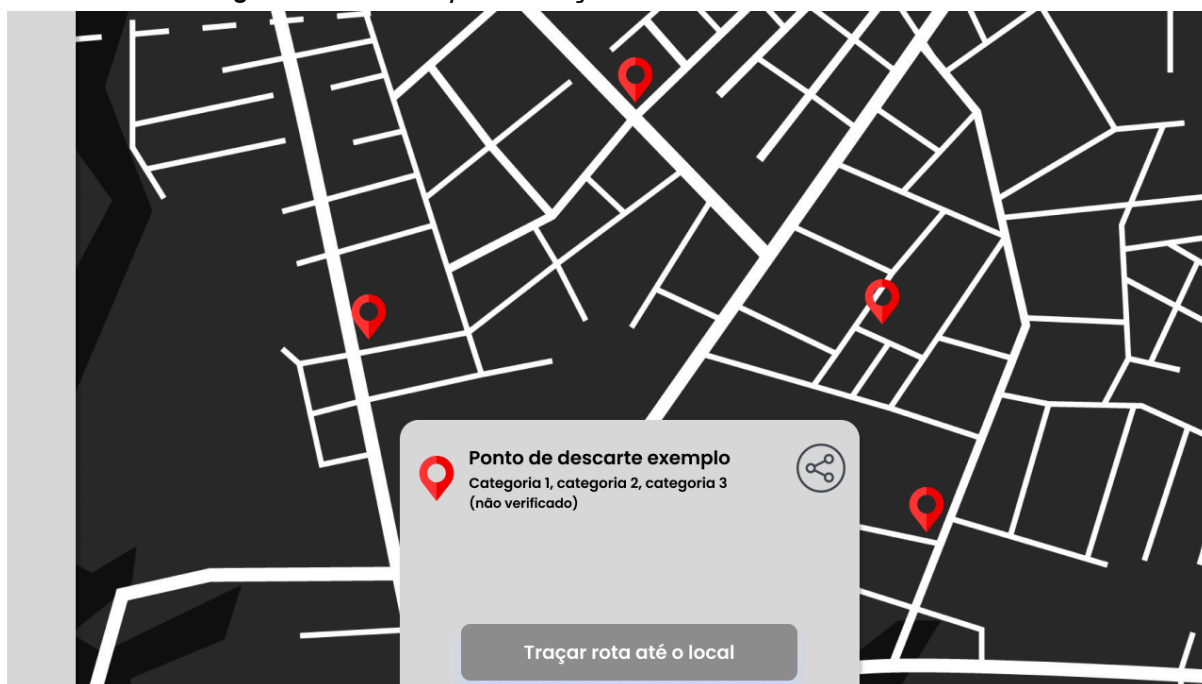
Figura 22 – Protótipo de Visualizar Pontos de Descarte





Fonte: Autor (2024)

Figura 23 – Protótipo de Traçar Rota até o Local de Descarte



Fonte: Autor (2024)

Figura 24 – Protótipo de Adicionar Ponto de Descarte

Cadastrar ponto de descarte

Nome

Descrição

Foto

Tipos de resíduos coletados

Cadastrar

Cadastrar ponto de descarte

Nome

Descrição

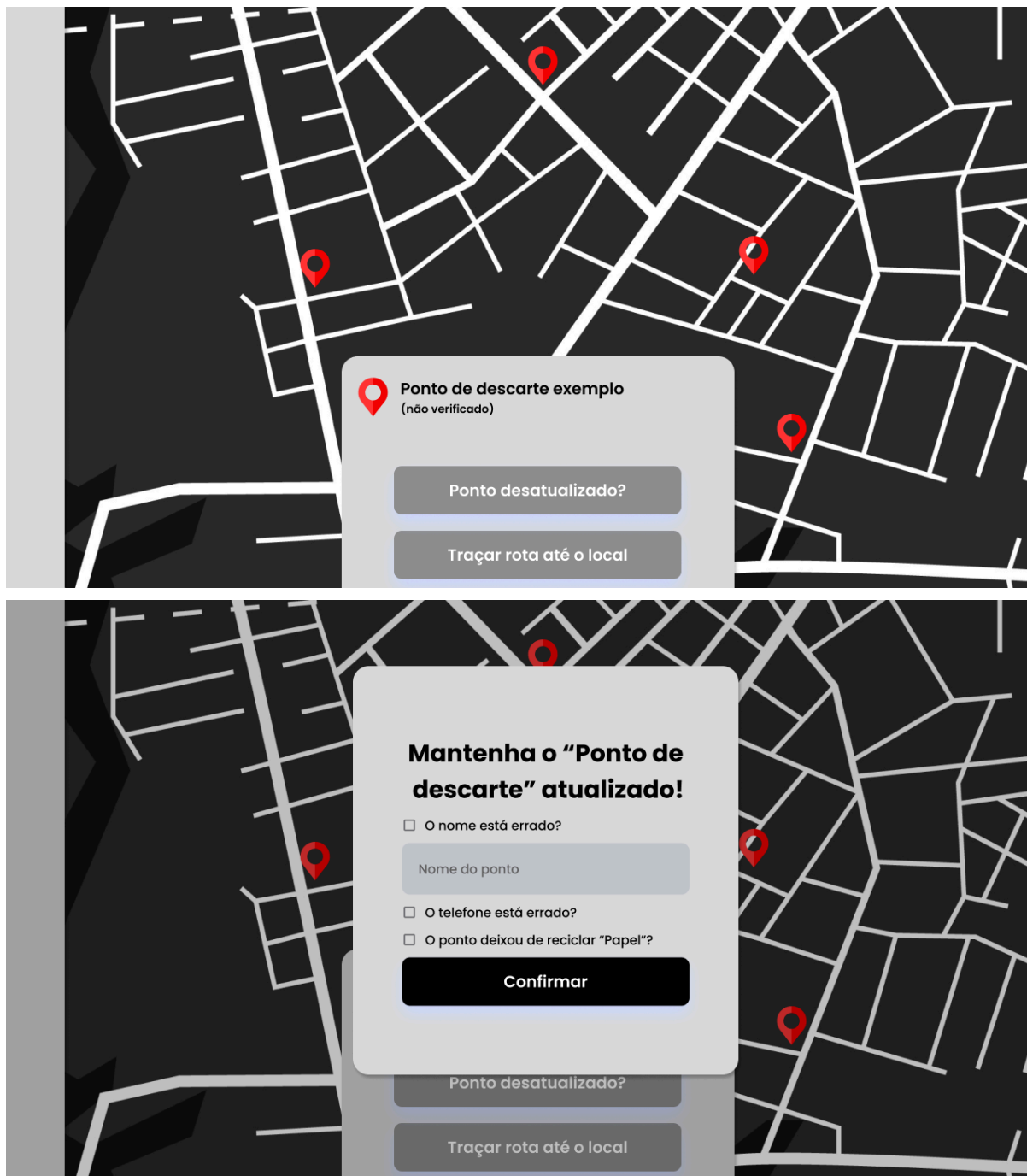
Foto

Tipos de resíduos coletados

- Material Orgânico
- Papel e Papelão
- Plásticos
- Vidros
- Metais

Fonte: Autor (2024)

Figura 25 – Protótipo de Verificar Ponto de Descarte



Fonte: Autor (2024)

Figura 26 – Protótipo de Visualizar Empresas de Reciclagem



Fonte: Autor (2024)

Figura 27 – Protótipo de Visualizar Rotas e Horários de Coleta



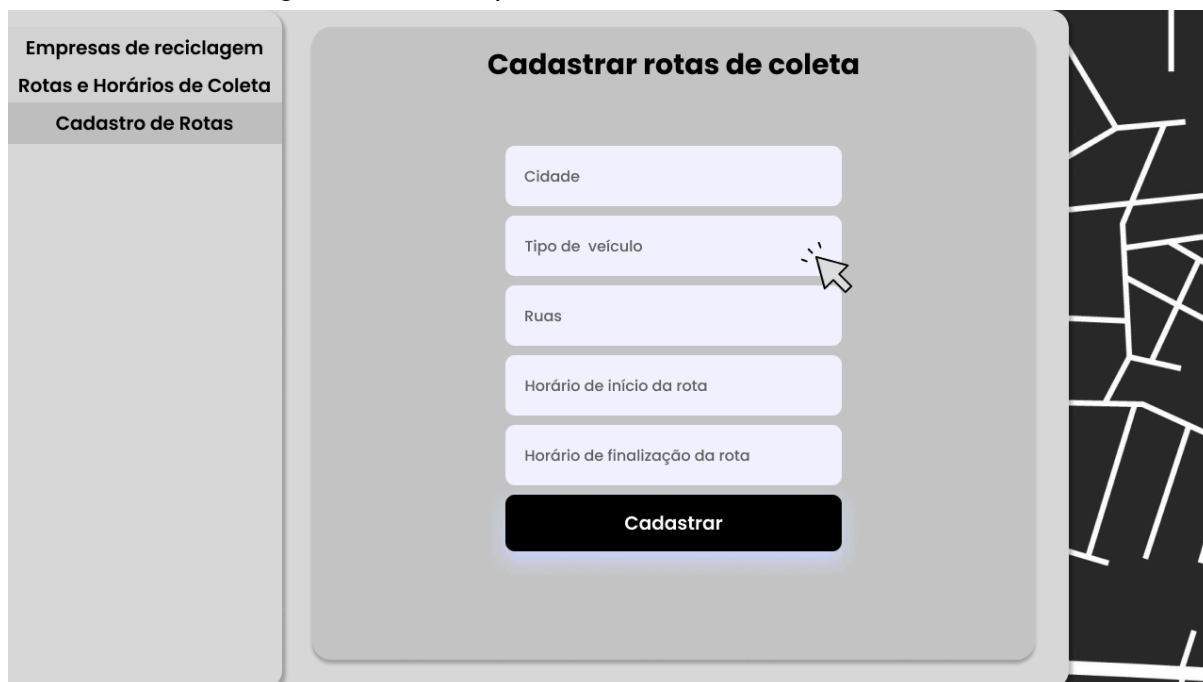
Fonte: Autor (2024)

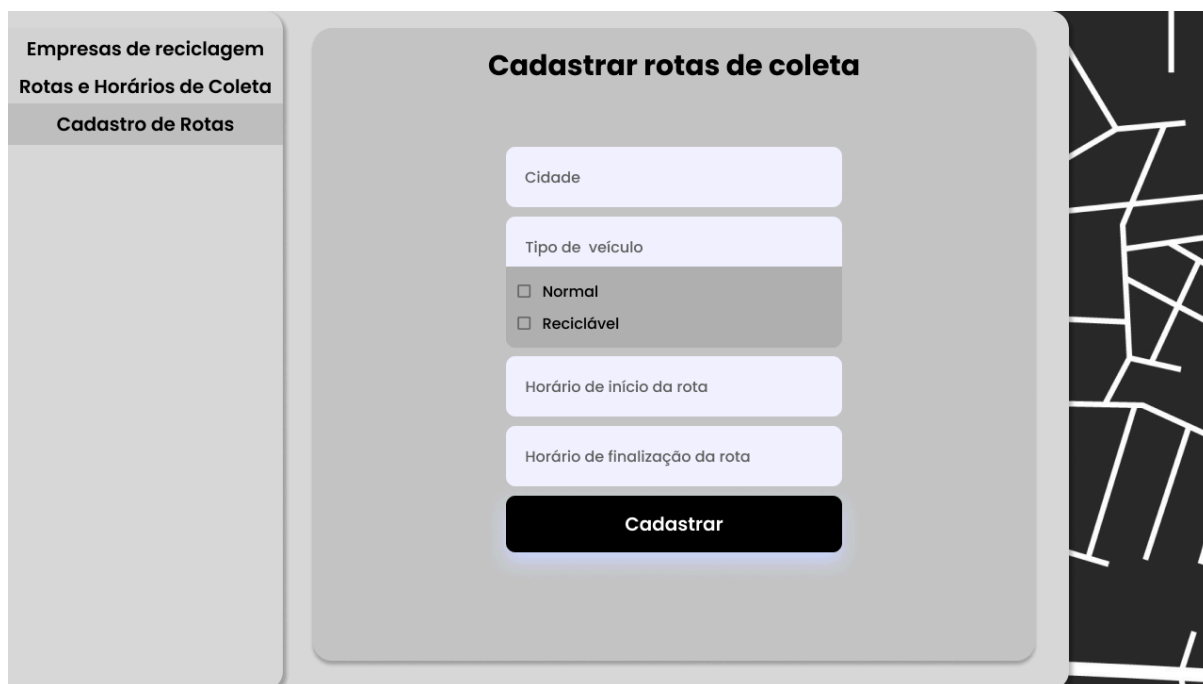
Figura 28 – Protótipo de Visualizar Perfil de Usuário



Fonte: Autor (2024)

Figura 29 – Protótipo de Cadastro de Rotas de Coleta





Empresas de reciclagem
Rotas e Horários de Coleta
Cadastro de Rotas

Cadastrar rotas de coleta

Cidade

Tipo de veículo

Normal
 Reciclável

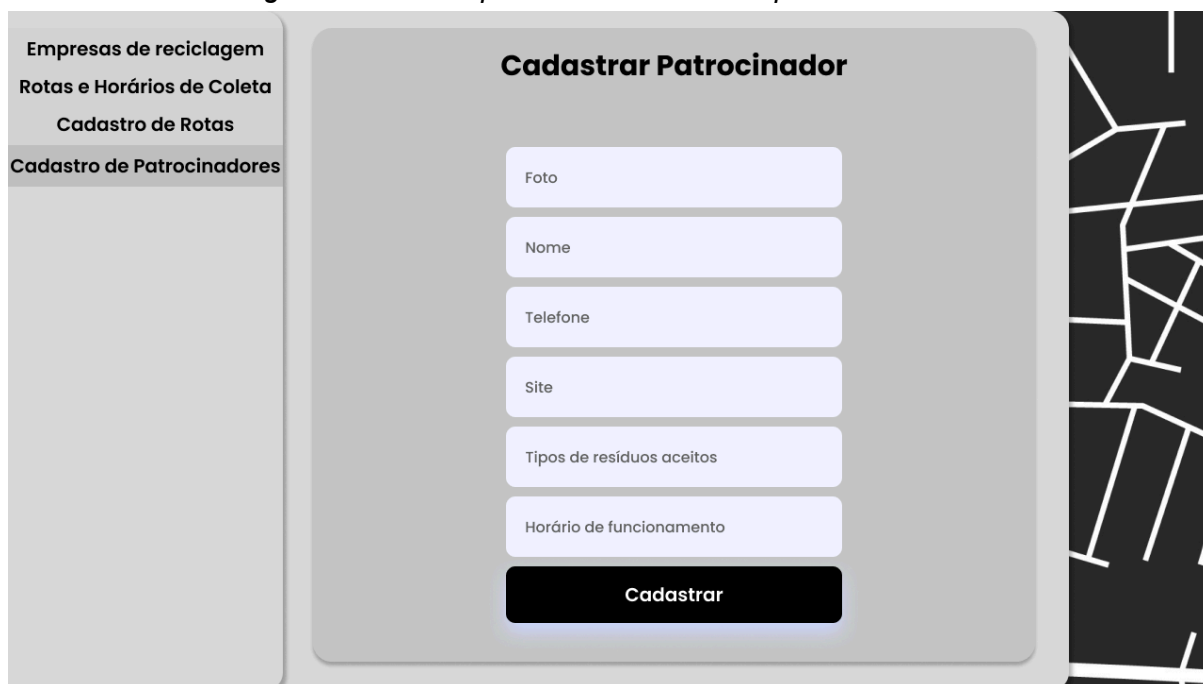
Horário de início da rota

Horário de finalização da rota

Cadastrar

Fonte: Autor (2024)

Figura 30 – Protótipo de Cadastro de Empresas Patrocinadoras



Empresas de reciclagem
Rotas e Horários de Coleta
Cadastro de Rotas
Cadastro de Patrocinadores

Cadastrar Patrocinador

Foto

Nome

Telefone

Site

Tipos de resíduos aceitos

Horário de funcionamento

Cadastrar

Fonte: Autor (2024)

Figura 31 – Protótipo de Excluir um Ponto de Descarte



Fonte: Autor (2024)

Apêndice B - Introdução à Avaliação:

Olá!

Gostaria de convidá-lo(a) a participar da avaliação do meu projeto de TCC, Lixo no Ponto (lixonoponto.live). Sua participação é muito importante para mim e ajudará a melhorar ainda mais meu projeto.

Para uma experiência completa, para quem não é de Florianópolis, peço que faça o teste posteriormente negando a localização e selecionando a cidade Florianópolis para ter uma experiência completa e ver os resultados que hoje são cadastrados apenas pelo equipe Lixo no Ponto.

Para participar, peço que você realize as seguintes tarefas na aplicação:

1. Realizar o cadastro de sua conta
2. Realizar a entrada na sua conta
3. Buscar pontos de descarte no mapa
4. Cadastrar um ponto de descarte real ou fictício (preferencialmente real)
5. Visualizar seu próprio perfil
6. Visualizar as empresas de reciclagem
7. Visualizar as rotas de coleta de lixo
8. Filtrar rotas de coleta de lixo por rua ou por tipo de coleta
9. Após completar essas tarefas, por favor, responda ao seguinte formulário:
<https://forms.gle/aaVL6nu9VVWBbPm36>

Agradecemos imensamente sua colaboração!

Atenciosamente,
Equipe Lixo no Ponto

Apêndice C - Questionário de Avaliação:

Formulário de Avaliação do Projeto Lixo no Ponto

Prezado(a),

Você está sendo convidado(a) a participar da avaliação do projeto "Lixo no Ponto", sob a responsabilidade do autor André Joaquim Bez Birolo, da Universidade Federal de Santa Catarina. Esta avaliação pretende avaliar o Trabalho de Conclusão de Curso (TCC). Para isso, você será convidado(a) a responder a uma série de perguntas sobre o projeto Lixo no Ponto.

A sua participação não trará nenhum benefício pessoal direto, mas os resultados serão importantes para ajudar a avaliar o projeto Lixo no Ponto.

Não esperamos que a sua participação nesta pesquisa traga qualquer desconforto ou incômodo, mas as perguntas podem tratar de assuntos que você ache delicados e as respostas podem exigir de você algum esforço.

Quando divulgarmos os resultados da pesquisa, não faremos qualquer menção a nomes de pessoas, locais ou episódios que possam servir para identificar você. Serão tomados todos os cuidados para que nenhum participante possa ser identificado.

Você não terá nenhuma despesa, e não será remunerado. Os resultados da pesquisa serão analisados e publicados, mas sua identidade será mantida em sigilo.

Antes de você responder às perguntas relacionadas à avaliação, apresentaremos o Termo de Consentimento Livre e Esclarecido (TCLE) para sua leitura e anuência.

Seja bem-vindo(a)!

Termo de Consentimento Livre e Esclarecido (TCLE)

Acredito ter sido suficientemente informado(a) a respeito das informações que li ou que foram lidas para mim sobre o projeto "Lixo no Ponto". Eu discuti com os pesquisadores responsáveis e está claro para mim quais são os propósitos do estudo, os procedimentos a serem realizados, quais são seus desconfortos, riscos e a garantia de confidencialidade dos meus dados. Entendo que sempre que eu tiver dúvidas elas serão esclarecidas e que minha participação é isenta de despesas. Concordo voluntariamente em fazer parte deste estudo e poderei retirar o meu consentimento a qualquer momento, antes ou durante o mesmo, sem penalidades ou prejuízo ou perda de qualquer benefício que eu possa ter adquirido.

* Indica uma pergunta obrigatória

01/10/2024, 14:54

Formulário de Avaliação do Projeto Lixo no Ponto

1. E-mail *

2. Agradecemos sua participação. Você concorda em participar dessa pesquisa? *

Marcar apenas uma oval. Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação. Eu não desejo participar do estudo.

Perfil do Usuário

Essa seção se destina à conhecer melhor nossos usuários, as informações aqui coletadas são essenciais para entender melhor o público-alvo e ajustar o projeto conforme necessário.

3. Qual seu nome? *

4. Qual sua idade? (Apenas números) *

5. Qual seu sexo? *

Marcar apenas uma oval. Masculino Feminino Prefiro não dizer Outro

01/10/2024, 14:54

Formulário de Avaliação do Projeto Lixo no Ponto

6. Qual plataforma você pretende utilizar para testar nosso projeto? *

Marcar apenas uma oval.

- Computador/Notebook
 Dispositivos Móveis

7. Você tem familiaridade com o uso de aplicativos em dispositivos móveis e/ou computador/notebook? *

Marcar apenas uma oval.

- Pouca
 Mediana
 Muita

Ampliar o conhecimento sobre a prática e políticas de gestão de resíduos urbanos da comunidade em um ambiente urbano.

Pedimos que utilize números apenas.

8. Quantos pontos de descarte você conhecia antes de usar o sistema?

9. Quantos pontos de descarte novos você conseguiu identificar após usar o sistema? *

10. Quantas rotas de coleta de lixo você conhecia antes de usar o sistema?

01/10/2024, 14:54

Formulário de Avaliação do Projeto Lixo no Ponto

11. Quantas rotas de coleta de lixo novas você conseguiu identificar após usar o sistema? *

12. As informações sobre os pontos de descarte são de fácil compreensão. *

Marcar apenas uma oval.

Sim

Não

13. As informações sobre as rotas de coleta de lixo são de fácil compreensão *

Marcar apenas uma oval.

Sim

Não

Escala Likert

Resposta na escala Likert: (1) Discordo totalmente; (2) Discordo parcialmente; (3) Indiferente; (4) Concordo parcialmente; (5) Concordo totalmente

14. As informações sobre os pontos de descarte são adequadas (suficientes, claras e corretas). *

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo totalmente

01/10/2024, 14:54

Formulário de Avaliação do Projeto Lixo no Ponto

15. As informações sobre as rotas de coleta de lixo são adequadas (suficientes, claras e corretas) *

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo totalmente

Avaliar a usabilidade do projeto Lixo no Ponto em relação à experiência do usuário e à eficiência do sistema, do ponto de vista dos usuários finais, no contexto de um ambiente urbano.

Resposta na escala Likert: (1) Discordo totalmente; (2) Discordo parcialmente; (3) Indiferente; (4) Concordo parcialmente; (5) Concordo totalmente

16. Eu acho que gostaria de usar esse sistema com frequência. *

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo totalmente

17. Eu acho o sistema desnecessariamente complexo. *

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo totalmente

18. Eu achei o sistema fácil de usar. *

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo totalmente

01/10/2024, 14:54

Formulário de Avaliação do Projeto Lixo no Ponto

19. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema. *

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo totalmente

20. Eu acho que as várias funções do sistema estão muito bem integradas. *

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo totalmente

21. Eu acho que o sistema apresenta muita inconsistência. *

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo totalmente

22. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente. *

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo totalmente

01/10/2024, 14:54

Formulário de Avaliação do Projeto Lixo no Ponto

23. Eu achei o sistema atrapalhado de usar. *

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo totalmente

24. Eu me senti confiante ao usar o sistema. *

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo totalmente

25. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema. *

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo totalmente

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

Apêndice D - Código Fonte Front End:

```

"node_modules/obuf": {
  "version": "1.1.2",
  "resolved": "https://registry.npmjs.org/obuf/-/obuf-1.1.2.tgz",
  "integrity":
"sha512-PX1wu0AmAdPqOL1mWhqmlOd8kOIZQwGZw6rh7uby9fTc5lhaOWFLX3l6R1hrF9
k3zUY40e6igsLGkDXK92LJNg==",
  "dev": true
},
"node_modules/on-finished": {
  "version": "2.4.1",
  "resolved": "https://registry.npmjs.org/on-finished/-/on-finished-2.4.1.tgz",
  "integrity":
"sha512-oVlzkG3ENAhCk2zdv7IJwd/QUD4z2RxRwpkcGY8psCVcCYZNq4wYnVWALHM+br
tuJjePWiYF/CImuDr8Ch5+kg==",
  "dependencies": {
    "ee-first": "1.1.1"
  },
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/on-headers": {
  "version": "1.0.2",
  "resolved": "https://registry.npmjs.org/on-headers/-/on-headers-1.0.2.tgz",
  "integrity":
"sha512-pZAE+FJLoylTytdqK0U5s+F1pjN0JP3OzFi/u8Rx+EV5/W+JTWGXG8xFzevE7AjBfD
qHv/8vL8qQslhHnqRkrA==",
  "dev": true,
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/once": {
  "version": "1.4.0",
  "resolved": "https://registry.npmjs.org/once/-/once-1.4.0.tgz",
  "integrity":
"sha512-lNaJgI+2Q5URQBkccEKHTQOPaXdUxnZZEIQTY0MFUAuaEqe1E+Nyvgdz/alyNi
6Z9MzO5dv1H8n58/GELp3+w==",
  "dev": true,
  "dependencies": {
    "wrappy": "1"
  }
},

```

```

"node_modules/onetime": {
  "version": "7.0.0",
  "resolved": "https://registry.npmjs.org/onetime/-/onetime-7.0.0.tgz",
  "integrity":
"sha512-VXJc87FScF88uafS3JIIDgvAm+c/Slfz06lorj2uAY34rIUu0Nt+v8wreilmcrgAjjlHp1rX
pTDILOGw29WwQ==",
  "dev": true,
  "dependencies": {
    "mimic-function": "^5.0.0"
  },
  "engines": {
    "node": ">=18"
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
},
"node_modules/open": {
  "version": "10.1.0",
  "resolved": "https://registry.npmjs.org/open/-/open-10.1.0.tgz",
  "integrity":
"sha512-mnkeQ1qP5Ue2wd+aivTD3NHd/IZ96Lu0jgf0pwktLPtx6cTZiH7tyeGRRHs0zX0bra
hXPnXIUnbeXyBBulaw==",
  "dev": true,
  "dependencies": {
    "default-browser": "^5.2.1",
    "define-lazy-prop": "^3.0.0",
    "is-inside-container": "^1.0.0",
    "is-wsl": "^3.1.0"
  },
  "engines": {
    "node": ">=18"
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
},
"node_modules/ora": {
  "version": "5.4.1",
  "resolved": "https://registry.npmjs.org/ora/-/ora-5.4.1.tgz",
  "integrity":
"sha512-5b6Y85tPxZZ7QytO+BQzysW31HJku27cRIIkbAXaNx+BdcVi+LIRFmVXzeF6a7JC
wJpyw5c4b+YSVImQIrBpuQ==",
  "dev": true,
  "dependencies": {
    "bl": "^4.1.0",
    "chalk": "^4.1.0",
    "cli-cursor": "^3.1.0",

```

```

    "cli-spinners": "^2.5.0",
    "is-interactive": "^1.0.0",
    "is-unicode-supported": "^0.1.0",
    "log-symbols": "^4.1.0",
    "strip-ansi": "^6.0.0",
    "wcwidth": "^1.0.1"
  },
  "engines": {
    "node": ">=10"
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
},
"node_modules/ora/node_modules/ansi-styles": {
  "version": "4.3.0",
  "resolved": "https://registry.npmjs.org/ansi-styles/-/ansi-styles-4.3.0.tgz",
  "integrity":
"sha512-zbB9rCJAT1rbjiVDb2hqKFHNYLxgtk8NURxZ3IZwD3F6NtzbXZQCnnSi1Lkx+IDohd
PIFp222wVALIheZJQSEg==",
  "dev": true,
  "dependencies": {
    "color-convert": "^2.0.1"
  },
  "engines": {
    "node": ">=8"
  },
  "funding": {
    "url": "https://github.com/chalk/ansi-styles?sponsor=1"
  }
},
"node_modules/ora/node_modules/chalk": {
  "version": "4.1.2",
  "resolved": "https://registry.npmjs.org/chalk/-/chalk-4.1.2.tgz",
  "integrity":
"sha512-oKnbfyRiXpUuez8iBMmyEa4nbj4lOQyuhc/wy9kY7/WVPcwIO9VA668Pu8RkO7+
0G76SLROeyw9CpQ061i4mA==",
  "dev": true,
  "dependencies": {
    "ansi-styles": "^4.1.0",
    "supports-color": "^7.1.0"
  },
  "engines": {
    "node": ">=10"
  },
  "funding": {
    "url": "https://github.com/chalk/chalk?sponsor=1"
  }
}

```

```

},
"node_modules/ora/node_modules/cli-cursor": {
  "version": "3.1.0",
  "resolved": "https://registry.npmjs.org/cli-cursor/-/cli-cursor-3.1.0.tgz",
  "integrity":
"sha512-l/zHAwsKf9FqGoXM4WWRACob9+SNUkZTd94DWF57E4toouRulbCxcUh6RKUEO
QIYTHJnzKPMYsvPNaaSLNfLZw==",
  "dev": true,
  "dependencies": {
    "restore-cursor": "^3.1.0"
  },
  "engines": {
    "node": ">=8"
  }
},
"node_modules/ora/node_modules/color-convert": {
  "version": "2.0.1",
  "resolved": "https://registry.npmjs.org/color-convert/-/color-convert-2.0.1.tgz",
  "integrity":
"sha512-RRECPsj7iu/xb5oKYcsFHSppFNnsj/52OVTRKb4zP5onXwVF3zVmmToNcOfGC+C
RDpfK/U584fMg38ZHCaElKQ==",
  "dev": true,
  "dependencies": {
    "color-name": "~1.1.4"
  },
  "engines": {
    "node": ">=7.0.0"
  }
},
"node_modules/ora/node_modules/color-name": {
  "version": "1.1.4",
  "resolved": "https://registry.npmjs.org/color-name/-/color-name-1.1.4.tgz",
  "integrity":
"sha512-dOy+3AuW3a2wNbZHIuMZpTcgjGuLU/uBL/ubcZF9OXbDo8ff4O8yVp5Bf0efS8uEo
Yo5q4Fx7dY9OgQGXgAsQA==",
  "dev": true
},
"node_modules/ora/node_modules/has-flag": {
  "version": "4.0.0",
  "resolved": "https://registry.npmjs.org/has-flag/-/has-flag-4.0.0.tgz",
  "integrity":
"sha512-EykJt/Q1KjTWctppgIAgfSO0tKVuZUjhgMr17kqTumMI6Afv3EISleU7qZUzoXDFTA
HTDC4NOoG/ZxU3EvIMPQ==",
  "dev": true,
  "engines": {
    "node": ">=8"
  }
},

```

```

"node_modules/ora/node_modules/onetime": {
  "version": "5.1.2",
  "resolved": "https://registry.npmjs.org/onetime/-/onetime-5.1.2.tgz",
  "integrity":
"sha512-kbpaSSGJTWdAY5KPVeMOKXSRPtr8C8C7wodJbcsd51jRnmD+GZu8Y0VoU6Dm5
Z4vWr0lg/1NKuWRKf7j5aaYSg==",
  "dev": true,
  "dependencies": {
    "mimic-fn": "^2.1.0"
  },
  "engines": {
    "node": ">=6"
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
},
"node_modules/ora/node_modules/restore-cursor": {
  "version": "3.1.0",
  "resolved": "https://registry.npmjs.org/restore-cursor/-/restore-cursor-3.1.0.tgz",
  "integrity":
"sha512-l+sSefzHpj5qimhFSE5a8nufZYAM3sBSVMAPtYkmC+4EH2anSGaEMXSD0izRQb
u9nfyQ9y5JrVmp7E8oZrUjvA==",
  "dev": true,
  "dependencies": {
    "onetime": "^5.1.0",
    "signal-exit": "^3.0.2"
  },
  "engines": {
    "node": ">=8"
  }
},
"node_modules/ora/node_modules/signal-exit": {
  "version": "3.0.7",
  "resolved": "https://registry.npmjs.org/signal-exit/-/signal-exit-3.0.7.tgz",
  "integrity":
"sha512-wnD2ZE+I+SPC/uoS0vXeE9L1+0wuaMqKlfz9AMUo38JsyLSBWSFcHR1Rri62LZc
12vLr1gb3jl7iwQhgwpAbGQ==",
  "dev": true
},
"node_modules/ora/node_modules/supports-color": {
  "version": "7.2.0",
  "resolved": "https://registry.npmjs.org/supports-color/-/supports-color-7.2.0.tgz",
  "integrity":
"sha512-qpCAvRI9stuOHveKsn7HncJRvv501qlackKzQIO/+Lwxc9+0q2wLyv4Dfvt80/DPn2pq
OBsJdDiogXGR9+OvwRw==",
  "dev": true,
  "dependencies": {

```

```

    "has-flag": "^4.0.0"
  },
  "engines": {
    "node": ">=8"
  }
},
"node_modules/ordered-binary": {
  "version": "1.5.1",
  "resolved": "https://registry.npmjs.org/ordered-binary/-/ordered-binary-1.5.1.tgz",
  "integrity":
"sha512-5VyHfHY3cd0iza71JepYG50My+YUbrFtGoUz2ooEydPyPM7Aai/JW098juLr+RG6+rDJuzNNTsEQU2DZa1A41A==",
  "dev": true
},
"node_modules/os-tmpdir": {
  "version": "1.0.2",
  "resolved": "https://registry.npmjs.org/os-tmpdir/-/os-tmpdir-1.0.2.tgz",
  "integrity":
"sha512-D2FR03Vir7Flu45XBY20mTb+/ZSWB00sjU9jdQXt83gDrI4Ztz5Fs7/yy74g2N5SVQY4xY1qDr4rNddwYRVX0g==",
  "dev": true,
  "engines": {
    "node": ">=0.10.0"
  }
},
"node_modules/p-limit": {
  "version": "4.0.0",
  "resolved": "https://registry.npmjs.org/p-limit/-/p-limit-4.0.0.tgz",
  "integrity":
"sha512-5b0R4txpzjPWVw/cXXUResoD4hb6U/x9BH08L7nw+GN1sezDzPdxeRvpc9c433fZhBan/wusjbCsqwqm4EIBIQ==",
  "dev": true,
  "dependencies": {
    "yocto-queue": "^1.0.0"
  },
  "engines": {
    "node": "^12.20.0 || ^14.13.1 || >=16.0.0"
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
},
"node_modules/p-locate": {
  "version": "6.0.0",
  "resolved": "https://registry.npmjs.org/p-locate/-/p-locate-6.0.0.tgz",
  "integrity":
"sha512-wPrq66LhI7/4AGC6I+cqxT07LhXvWL08LNxz1fENOW0Ap4sRZZ/gZpTTJ5jpurzzfS2W/Ge9BY3LgLjCShcw==",

```



```

    "dev": true,
    "dependencies": {
      "p-limit": "^4.0.0"
    },
    "engines": {
      "node": "^12.20.0 || ^14.13.1 || >=16.0.0"
    },
    "funding": {
      "url": "https://github.com/sponsors/sindresorhus"
    }
  },
  "node_modules/p-map": {
    "version": "4.0.0",
    "resolved": "https://registry.npmjs.org/p-map/-/p-map-4.0.0.tgz",
    "integrity":
"sha512-/bjOqmgETBYB5BoEeGVea8dmvHb2m9GLy1E9W43yeyfP6QQCZGFNa+XRceJE
uDB6zqr+gKpIAmILebMpykw/MQ==",
    "dev": true,
    "dependencies": {
      "aggregate-error": "^3.0.0"
    },
    "engines": {
      "node": ">=10"
    },
    "funding": {
      "url": "https://github.com/sponsors/sindresorhus"
    }
  },
  "node_modules/p-retry": {
    "version": "6.2.0",
    "resolved": "https://registry.npmjs.org/p-retry/-/p-retry-6.2.0.tgz",
    "integrity":
"sha512-JA6nkq6hKyWLLasXQXUrO4z8BUZGUt/LjJxx8Gb2+2ntodU/SS63YZ8b0LUTbQ8
ZB9iwOfhEPHg4ykKnn2KsA==",
    "dev": true,
    "dependencies": {
      "@types/retry": "0.12.2",
      "is-network-error": "^1.0.0",
      "retry": "^0.13.1"
    },
    "engines": {
      "node": ">=16.17"
    },
    "funding": {
      "url": "https://github.com/sponsors/sindresorhus"
    }
  },
  "node_modules/p-retry/node_modules/retry": {

```

```

"version": "0.13.1",
"resolved": "https://registry.npmjs.org/retry/-/retry-0.13.1.tgz",
"integrity":
"sha512-XQBQ3I8W1Cge0Seh+6gjj03LbmRFWuoszgK9ooCpwYIrrhoO80pfq4cUkU5Dkkn
wfOfFteRwIz56PYOGYyFWdg==",
  "dev": true,
  "engines": {
    "node": ">= 4"
  }
},
"node_modules/p-try": {
  "version": "2.2.0",
  "resolved": "https://registry.npmjs.org/p-try/-/p-try-2.2.0.tgz",
  "integrity":
"sha512-R4nPAVTAU0B9D35/Gk3uJf/7XYbQcyohSKdvAxIRSNghFI4e71hVoGnBNQz9cWa
XxO2110KTC+3jMdvvoKw6dQ==",
  "dev": true,
  "engines": {
    "node": ">=6"
  }
},
"node_modules/package-json-from-dist": {
  "version": "1.0.0",
  "resolved":
"https://registry.npmjs.org/package-json-from-dist/-/package-json-from-dist-1.0.0.tgz",
  "integrity":
"sha512-dATvCeZN/8wQsGywez1mzHtTIP22H8OEFPrVMLNr4/eGa+ijtLn/6M5f0dY8UKNrC2
O9UCU6SSoG3qRKnt7STw==",
  "dev": true
},
"node_modules/pacote": {
  "version": "18.0.6",
  "resolved": "https://registry.npmjs.org/pacote/-/pacote-18.0.6.tgz",
  "integrity":
"sha512-+eK3G27SMwsB8kLLuj4h1FUhHtwiEUo21Tw8wNjmvdlpOEr613edv+8FUsTj/4F/VN
5ywGE19X18N7CC2EJk6A==",
  "dev": true,
  "dependencies": {
    "@npmcli/git": "^5.0.0",
    "@npmcli/installed-package-contents": "^2.0.1",
    "@npmcli/package-json": "^5.1.0",
    "@npmcli/promise-spawn": "^7.0.0",
    "@npmcli/run-script": "^8.0.0",
    "cacache": "^18.0.0",
    "fs-minipass": "^3.0.0",
    "minipass": "^7.0.2",
    "npm-package-arg": "^11.0.0",
    "npm-packlist": "^8.0.0",

```

```

    "npm-pick-manifest": "^9.0.0",
    "npm-registry-fetch": "^17.0.0",
    "proc-log": "^4.0.0",
    "promise-retry": "^2.0.1",
    "sigstore": "^2.2.0",
    "ssri": "^10.0.0",
    "tar": "^6.1.11"
  },
  "bin": {
    "pacote": "bin/index.js"
  },
  "engines": {
    "node": "^16.14.0 || >=18.0.0"
  }
},
"node_modules/parent-module": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/parent-module/-/parent-module-1.0.1.tgz",
  "integrity":
"sha512-GQ2EWRpQV8/o+Aw8YqtfZZPfNRWZYkbidE9k5rpl/hC3vtHHBfGm2Ifi6qWV+coD
GkrUKZAxE3Lot5kcsRIh+g==",
  "dev": true,
  "dependencies": {
    "callsites": "^3.0.0"
  },
  "engines": {
    "node": ">=6"
  }
},
"node_modules/parse-json": {
  "version": "5.2.0",
  "resolved": "https://registry.npmjs.org/parse-json/-/parse-json-5.2.0.tgz",
  "integrity":
"sha512-ayCKvm/phCGxOkYRSCM82iDwct8/EonSEgCSxWxD7ve6jHggsFI4fZVQBPRNgQ
oKiuV/odhFrGzQXZwbifC8Rg==",
  "dev": true,
  "dependencies": {
    "@babel/code-frame": "^7.0.0",
    "error-ex": "^1.3.1",
    "json-parse-even-better-errors": "^2.3.0",
    "lines-and-columns": "^1.1.6"
  },
  "engines": {
    "node": ">=8"
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
}

```

```

},
"node_modules/parse-json/node_modules/json-parse-even-better-errors": {
  "version": "2.3.1",
  "resolved":
"https://registry.npmjs.org/json-parse-even-better-errors/-/json-parse-even-better-errors-2.3.1
.tgz",
  "integrity":
"sha512-xyFwyhro/JEof6Ghe2iz2NcXoj2sloNsWr/XsERDK/oiPCfaNhl5ONfp+jQdAZRQQ0IJ
WNzH9zIZF7li91kh2w==",
  "dev": true
},
"node_modules/parse-node-version": {
  "version": "1.0.1",
  "resolved":
"https://registry.npmjs.org/parse-node-version/-/parse-node-version-1.0.1.tgz",
  "integrity":
"sha512-3YHIOa/JgH6Mnpr05jP9eDG254US9ek25LyIxZIDltp2iJtwyaXQb57IBYldT3MowkU
FYEV2XXNAYIPIESvJIA==",
  "dev": true,
  "engines": {
    "node": ">= 0.10"
  }
},
"node_modules/parse5": {
  "version": "7.1.2",
  "resolved": "https://registry.npmjs.org/parse5/-/parse5-7.1.2.tgz",
  "integrity":
"sha512-Czj1WaSVpaoj0wbhMzLmWD69anp2WH7FXMB9n1Sy8/ZFF9jolsQVMu1lj5WlyG
mcBmhk7EOndpO4mlpihVqAXw==",
  "devOptional": true,
  "dependencies": {
    "entities": "^4.4.0"
  },
  "funding": {
    "url": "https://github.com/inikulin/parse5?sponsor=1"
  }
},
"node_modules/parse5-html-rewriting-stream": {
  "version": "7.0.0",
  "resolved":
"https://registry.npmjs.org/parse5-html-rewriting-stream/-/parse5-html-rewriting-stream-7.0.0.
tgz",
  "integrity":
"sha512-mazCyGWkmCRWDI15Zp+UiCqMp/0dgEmkZRVhlsqqKYr4SsVm/TvnSpD9fCvqCA
2zoWJcfRym846ejWBBHRiYEg==",
  "dev": true,
  "dependencies": {
    "entities": "^4.3.0",

```

```

    "parse5": "^7.0.0",
    "parse5-sax-parser": "^7.0.0"
  },
  "funding": {
    "url": "https://github.com/inikulin/parse5?sponsor=1"
  }
},
"node_modules/parse5-sax-parser": {
  "version": "7.0.0",
  "resolved": "https://registry.npmjs.org/parse5-sax-parser/-/parse5-sax-parser-7.0.0.tgz",
  "integrity":
"sha512-5A+v2SNsq8T6/mG3ahcz8ZtQ0OUFTatxPbeidoMB7tkJSGDY3tdfI4MHovtLQHkEn
5CGxijNWRQHhRQ6lRpXKg==",
  "dev": true,
  "dependencies": {
    "parse5": "^7.0.0"
  },
  "funding": {
    "url": "https://github.com/inikulin/parse5?sponsor=1"
  }
},
"node_modules/parseurl": {
  "version": "1.3.3",
  "resolved": "https://registry.npmjs.org/parseurl/-/parseurl-1.3.3.tgz",
  "integrity":
"sha512-CiyeOxFT/JZyN5m0z9Pfxw4SCBJ6Sygz1Dpl0wqjlhDEGGBP1GnsUVEL0p63hoG
1fcj3fHynXi9NYO4nWOL+qQ==",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/path": {
  "version": "0.12.7",
  "resolved": "https://registry.npmjs.org/path/-/path-0.12.7.tgz",
  "integrity":
"sha512-aXXC6s+1w7otVF9UletFkFcDsJeO7ISZBPUQhtb5O0xJe8LtYhj/GxldoL09bBj9+Zm
E2hNoHqQSFMN5fikh4Q==",
  "dependencies": {
    "process": "^0.11.1",
    "util": "^0.10.3"
  }
},
"node_modules/path-exists": {
  "version": "5.0.0",
  "resolved": "https://registry.npmjs.org/path-exists/-/path-exists-5.0.0.tgz",
  "integrity":
"sha512-RjhtfwJOxzcFmNOi6ltcbcu4lu+FL3zEj83dk4kAS+fVpTxXLO1b38RvJgT/0QwvV/L3
aY9TAnyv0EOqW4GoMQ==",

```

```

    "dev": true,
    "engines": {
      "node": "^12.20.0 || ^14.13.1 || >=16.0.0"
    }
  },
  "node_modules/path-is-absolute": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/path-is-absolute/-/path-is-absolute-1.0.1.tgz",
    "integrity":
"sha512-AVbw3UJ2e9bq64vSaS9Am0fje1Pa8pbGqTTsmXfaliMpnr5DIDhfJOuLj9Sf95ZPVD
AUerDfEk88MPmPe7UCQg==",
    "dev": true,
    "engines": {
      "node": ">=0.10.0"
    }
  },
  "node_modules/path-key": {
    "version": "3.1.1",
    "resolved": "https://registry.npmjs.org/path-key/-/path-key-3.1.1.tgz",
    "integrity":
"sha512-ojmeN0qd+y0jszEtoY48r0Peq5dwMEklICOu6Q5f41lfkswXuKtYrhgoTpLnlylcHm24U
hqx+5Tqm2InSwLhE6Q==",
    "dev": true,
    "engines": {
      "node": ">=8"
    }
  },
  "node_modules/path-parse": {
    "version": "1.0.7",
    "resolved": "https://registry.npmjs.org/path-parse/-/path-parse-1.0.7.tgz",
    "integrity":
"sha512-LDJzPVEEPR+y48z93A0Ed0yXb8pABYGWo/k5YYdYgpY2/2EsOsksJrq7IOHxryr
VOn1ejG6oAp8ahvOIQD8sw==",
    "dev": true
  },
  "node_modules/path-scurry": {
    "version": "1.11.1",
    "resolved": "https://registry.npmjs.org/path-scurry/-/path-scurry-1.11.1.tgz",
    "integrity":
"sha512-Xa4Nw17FS9ApQFJ9umLiJS4orGjm7ZzwUrwamcGQuHSzDyth9boKDaycYdDcZD
uqYATXw4HFXgaqWTctW/v1HA==",
    "dev": true,
    "dependencies": {
      "lru-cache": "^10.2.0",
      "minipass": "^5.0.0 || ^6.0.2 || ^7.0.0"
    },
    "engines": {
      "node": ">=16 || 14 >=14.18"
    }
  }

```

```

    },
    "funding": {
      "url": "https://github.com/sponsors/isaacs"
    }
  },
  "node_modules/path-scurry/node_modules/lru-cache": {
    "version": "10.4.3",
    "resolved": "https://registry.npmjs.org/lru-cache/-/lru-cache-10.4.3.tgz",
    "integrity":
"sha512-JNAzZcXrCt42VGLuYz0zfAzDfAvJWW6AfYIDBQyDV5DCII2m5sAmK+OIO7s59Xfs
RsWHp02jAJrRadPRGTt6SQ==",
    "dev": true
  },
  "node_modules/path-to-regexp": {
    "version": "0.1.10",
    "resolved": "https://registry.npmjs.org/path-to-regexp/-/path-to-regexp-0.1.10.tgz",
    "integrity":
"sha512-7lf7qcQidTku0Gu3YDPc8DJ1q7OOucfa/BSslwjuh56VU7katFvuM8hULfkWb3Fns/rs
VF7PwPKVw1sl5KQS9w=="
  },
  "node_modules/path-type": {
    "version": "5.0.0",
    "resolved": "https://registry.npmjs.org/path-type/-/path-type-5.0.0.tgz",
    "integrity":
"sha512-5HviZNaZcfqP95rwpv+1HDgUamezbqdSYTyztvwtJSnIH+3vnbmWslti8OFEndS9
84VT55M3jduxZbX351gg==",
    "dev": true,
    "engines": {
      "node": ">=12"
    }
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
},
"node_modules/picocolors": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/picocolors/-/picocolors-1.0.1.tgz",
  "integrity":
"sha512-anP1Z8qwhkbmu7MFP5iTt+wQKXgwzf7zTyGlcDzabySa9vd0Xt392U0rVzmz9poOa
Bj0uHJKyyo9/upk0HrEQew==",
  "dev": true
},
"node_modules/picomatch": {
  "version": "4.0.2",
  "resolved": "https://registry.npmjs.org/picomatch/-/picomatch-4.0.2.tgz",
  "integrity":
"sha512-M7BAV6Rlcy5u+m6oPhAPFgJTzAioX/6B0DxyvDlo9l8+T3nLKbrczg2WLUyzd45L8
RqfUMyGPzekaMvX2Ldkwg=="
}

```

```

    "dev": true,
    "engines": {
      "node": ">=12"
    },
    "funding": {
      "url": "https://github.com/sponsors/jonschlinkert"
    }
  },
  "node_modules/pify": {
    "version": "4.0.1",
    "resolved": "https://registry.npmjs.org/pify/-/pify-4.0.1.tgz",
    "integrity":
"sha512-uB80kBFb/tfd68bVleG9T5GGsGPjJrLAUpR5PZlRhBnlaRTQRjqdJSsIKkOP6OAlFbj
7GOrcudc5pNjZ+geV2g==",
    "dev": true,
    "optional": true,
    "engines": {
      "node": ">=6"
    }
  },
  "node_modules/pinkie": {
    "version": "2.0.4",
    "resolved": "https://registry.npmjs.org/pinkie/-/pinkie-2.0.4.tgz",
    "integrity":
"sha512-MnUuEycAemtSaeFSjXKW/aroV7akBbY+Sv+RkyqFjgAe73F+MR0TBWKBRDkmf
Wq/HiFmdavfZ1G7h4SPZXaCSg==",
    "dev": true,
    "engines": {
      "node": ">=0.10.0"
    }
  },
  "node_modules/pinkie-promise": {
    "version": "2.0.1",
    "resolved": "https://registry.npmjs.org/pinkie-promise/-/pinkie-promise-2.0.1.tgz",
    "integrity":
"sha512-0Gni6D4UcLTbv9c57DfxDGdr41XfgUjqWZu492f0clGr16zDU06BWP/RAEvOuo7C
Q0CNjHaLIM59YJJFm3NwIw==",
    "dev": true,
    "dependencies": {
      "pinkie": "^2.0.0"
    },
    "engines": {
      "node": ">=0.10.0"
    }
  },
  "node_modules/piscina": {
    "version": "4.6.1",
    "resolved": "https://registry.npmjs.org/piscina/-/piscina-4.6.1.tgz",

```



```

"integrity":
"sha512-z30AwWGtQE+Apr+2WBZensP2llvwoaMcOPkQIIEmSGMJNUvaYACyIPYrQM6wS
dUNJlnDVMSpLv7xTMJqIVshOA==",
  "dev": true,
  "optionalDependencies": {
    "nice-napi": "^1.0.2"
  }
},
"node_modules/pkg-dir": {
  "version": "7.0.0",
  "resolved": "https://registry.npmjs.org/pkg-dir/-/pkg-dir-7.0.0.tgz",

```

```

"integrity":
"sha512-le9z/WINcxxLp27BKOCHGde4ITq9UkiYKDzVo1nhk5sqGEXU3FpkwP5GM2voTGJ
kGd9B3Otl+Q4uwSOeSUtOBA==",
  "dev": true,
  "dependencies": {
    "find-up": "^6.3.0"
  },
  "engines": {
    "node": ">=14.16"
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
},
"node_modules/postcss": {
  "version": "8.4.38",
  "resolved": "https://registry.npmjs.org/postcss/-/postcss-8.4.38.tgz",

```

```

"integrity":
"sha512-WgIpdK03BSfXkHoQa3b/oulrotAkwrILDRSOB9D0bN86FdRyE9lppSp33aHNpGba0
JKCoB+drFLZkQoRRYae5A==",
  "dev": true,
  "funding": [
    {
      "type": "opencollective",
      "url": "https://opencollective.com/postcss/"
    },
    {
      "type": "tidelift",
      "url": "https://tidelift.com/funding/github/npm/postcss"
    },
    {
      "type": "github",
      "url": "https://github.com/sponsors/ai"
    }
  ],
  "dependencies": {
    "nanoid": "^3.3.7",

```

```

    "picocolors": "^1.0.0",
    "source-map-js": "^1.2.0"
  },
  "engines": {
    "node": "^10 || ^12 || >=14"
  }
},
"node_modules/postcss-loader": {
  "version": "8.1.1",
  "resolved": "https://registry.npmjs.org/postcss-loader/-/postcss-loader-8.1.1.tgz",
  "integrity":
"sha512-0leqyAsG6tYiDRCYKQJLAmgQr47DX6N7sFSWvQxt6AcupX8DIIdmykuk/o/tx0Lze3
ErGHJEp5OSRxlC6+NdQ==",
  "dev": true,
  "dependencies": {
    "cosmiconfig": "^9.0.0",
    "jiti": "^1.20.0",
    "semver": "^7.5.4"
  },
  "engines": {
    "node": ">= 18.12.0"
  },
  "funding": {
    "type": "opencollective",
    "url": "https://opencollective.com/webpack"
  },
  "peerDependencies": {
    "@rspack/core": "0.x || 1.x",
    "postcss": "^7.0.0 || ^8.0.1",
    "webpack": "^5.0.0"
  },
  "peerDependenciesMeta": {
    "@rspack/core": {
      "optional": true
    },
    "webpack": {
      "optional": true
    }
  }
},
"node_modules/postcss-media-query-parser": {
  "version": "0.2.3",
  "resolved":
"https://registry.npmjs.org/postcss-media-query-parser/-/postcss-media-query-parser-0.2.3.tgz",
  "integrity":
"sha512-3sOlxmbKcSHMjIUxQZKQ06jOswE7oVkJXpXmZdoB1r5l0q6gTFTQSHxNxOrCccEl
bW7dxNytifNEo8qidX2Vsig==",

```

```

    "dev": true
  },
  "node_modules/postcss-modules-extract-imports": {
    "version": "3.1.0",
    "resolved":
      "https://registry.npmjs.org/postcss-modules-extract-imports/-/postcss-modules-extract-imports-3.1.0.tgz",
    "integrity":
      "sha512-k3kNe0aNFQDAZGbin48pL2VNidTF0w4/eASDsxllyspobzU3wZQLOGj7L9gRe0Jo9/4uud09DsjFNH7winGv8Q==",
    "dev": true,
    "engines": {
      "node": "^10 || ^12 || >= 14"
    },
    "peerDependencies": {
      "postcss": "^8.1.0"
    }
  },
  "node_modules/postcss-modules-local-by-default": {
    "version": "4.0.5",
    "resolved":
      "https://registry.npmjs.org/postcss-modules-local-by-default/-/postcss-modules-local-by-default-4.0.5.tgz",
    "integrity":
      "sha512-6MieY7slfTK0hYfafw1OMEG+2bg8Q1ocHCpWLqOKj3JXIKu4G7btkmM/B7IFubYkYWmRSPLZi5chid63ZaZYw==",
    "dev": true,
    "dependencies": {
      "icss-utils": "^5.0.0",
      "postcss-selector-parser": "^6.0.2",
      "postcss-value-parser": "^4.1.0"
    },
    "engines": {
      "node": "^10 || ^12 || >= 14"
    },
    "peerDependencies": {
      "postcss": "^8.1.0"
    }
  },
  "node_modules/postcss-modules-scope": {
    "version": "3.2.0",
    "resolved":
      "https://registry.npmjs.org/postcss-modules-scope/-/postcss-modules-scope-3.2.0.tgz",
    "integrity":
      "sha512-oq+g1ssrsZOsx9M96c5w8laRmvEu9C3adDSjl8oTcbfkrTE8hx/zfyobUoWlxaKPO8bt6S62kxpw5GqypEw1QQ==",
    "dev": true,
    "dependencies": {

```

```

    "postcss-selector-parser": "^6.0.4"
  },
  "engines": {
    "node": "^10 || ^12 || >= 14"
  },
  "peerDependencies": {
    "postcss": "^8.1.0"
  }
},
"node_modules/postcss-modules-values": {
  "version": "4.0.0",
  "resolved":
    "https://registry.npmjs.org/postcss-modules-values/-/postcss-modules-values-4.0.0.tgz",
  "integrity":
    "sha512-RDxHkAiEG178gS2ofyvCsu7iycRv7oqw5xMWN9iMoR0N/7mf9D50ecQqUo5BZ9Zh
    2vH4bCUR/ktCqbB9m8vJjQ==",
  "dev": true,
  "dependencies": {
    "icss-utils": "^5.0.0"
  },
  "engines": {
    "node": "^10 || ^12 || >= 14"
  },
  "peerDependencies": {
    "postcss": "^8.1.0"
  }
},
"node_modules/postcss-selector-parser": {
  "version": "6.1.1",
  "resolved":
    "https://registry.npmjs.org/postcss-selector-parser/-/postcss-selector-parser-6.1.1.tgz",
  "integrity":
    "sha512-b4dlw/9V8A71rLIDsSwVmak9z2DuBUB7CA1/wSdeINEzqsjoSPeADTWNO09lpH49
    Diy3/JIZ2bSPB1dl3LJCHg==",
  "dev": true,
  "dependencies": {
    "cssesc": "^3.0.0",
    "util-deprecate": "^1.0.2"
  },
  "engines": {
    "node": ">=4"
  }
},
"node_modules/postcss-value-parser": {
  "version": "4.2.0",
  "resolved":
    "https://registry.npmjs.org/postcss-value-parser/-/postcss-value-parser-4.2.0.tgz",

```

```

"integrity":
"sha512-1NNCs6uurfkVbeXG4S8JFT9t19m45ICnif8zWLd5oPSZ50QnwMfK+H3jv408d4jw/7
Bttv5axS5IiHoLaVNHeQ==",
  "dev": true
},
"node_modules/proc-log": {
  "version": "4.2.0",
  "resolved": "https://registry.npmjs.org/proc-log/-/proc-log-4.2.0.tgz",
"integrity":
"sha512-g8+OnU/L2v+wyiVK+D5fA34J7EH8jZ8DDlwhRCMxmMj7UCBvxiO1mGeN+36JXI
KF4zevU4kRBd8lVgG9vLeIA==",
  "dev": true,
  "engines": {
    "node": "^14.17.0 || ^16.13.0 || >=18.0.0"
  }
},
"node_modules/process": {
  "version": "0.11.10",
  "resolved": "https://registry.npmjs.org/process/-/process-0.11.10.tgz",
"integrity":
"sha512-cdGef/drWFoydD1JsMzuFf8100nZI+GT+yacc2bEced5f9Rjk4z+WtFUTBu9PhOi9j/jf
mBPu0mMEY4wldAF8A==",
  "engines": {
    "node": ">= 0.6.0"
  }
},
"node_modules/process-nextick-args": {
  "version": "2.0.1",
"resolved":
"https://registry.npmjs.org/process-nextick-args/-/process-nextick-args-2.0.1.tgz",
"integrity":
"sha512-3ouUOpQhtgrbOa17J7+uxOTpITYWaGP7/AhoR3+A+/1e9skrzelGi/dXzEYyvbubE
F6Wn2ypscTKiKJFFn1ag==",
  "dev": true
},
"node_modules/promise-inflight": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/promise-inflight/-/promise-inflight-1.0.1.tgz",
"integrity":
"sha512-6zWPyEOFaQBJYcGMHBKTKJ3u6TBsnMFOIZSa6ce1e/ZrrsOlnHRHbabMjLiBYK
p+n44X9eUI6VUPaukCXHuG4g==",
  "dev": true
},
"node_modules/promise-retry": {
  "version": "2.0.1",
  "resolved": "https://registry.npmjs.org/promise-retry/-/promise-retry-2.0.1.tgz",

```

```

"integrity":
"sha512-y+WKFIBR8BGXnsNIIHFGPZmyDf3DFMoLhafIAnyZgV6rG6xu+JwesTo2Q9R6XwY
mtmwAFcAk3e35jEdoeh/3g==",
  "dev": true,
  "dependencies": {
    "err-code": "^2.0.2",
    "retry": "^0.12.0"
  },
  "engines": {
    "node": ">=10"
  }
},
"node_modules/proxy-addr": {
  "version": "2.0.7",
  "resolved": "https://registry.npmjs.org/proxy-addr/-/proxy-addr-2.0.7.tgz",
"integrity":
"sha512-IIQsMLSUDUPT44jdrU/O37qInifitDP+ZwrmmZcoSKyLKvtZxpyV0n2/bD/N4tBAAZ/g
JEdZU7KMraoK1+XYAg==",
  "dependencies": {
    "forwarded": "0.2.0",
    "ipaddr.js": "1.9.1"
  },
  "engines": {
    "node": ">= 0.10"
  }
},
"node_modules/proxy-addr/node_modules/ipaddr.js": {
  "version": "1.9.1",
  "resolved": "https://registry.npmjs.org/ipaddr.js/-/ipaddr.js-1.9.1.tgz",
"integrity":
"sha512-0KI/607xoxSToH7GjN1FfSbLoU0+btTicjsQSWQlh/hZykN8KpmMf7uYwPW3R+akZ
6R/w18ZIXSHBYXiYUPO3g==",
  "engines": {
    "node": ">= 0.10"
  }
},
"node_modules/prr": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/prr/-/prr-1.0.1.tgz",
"integrity":
"sha512-yPw4Sng1gWghHQWj0B3ZggWUm4qVbPwPFcRG8KyxiU7J2OHFSoEHKS+EZ3f
v5l1t9CyCiop6l/ZYeWbrgoQejw==",
  "dev": true,
  "optional": true
},
"node_modules/punycode": {
  "version": "1.4.1",
  "resolved": "https://registry.npmjs.org/punycode/-/punycode-1.4.1.tgz",

```

```

"integrity":
"sha512-jmYNEIW7yvO7TV33CjSmvSiE2yco3bV2czu/OzDKdMNVZQWfxCbiURLhf+47syQ
RBntjflLdd/H0egrzIG+oaFQ==",
  "dev": true
},
"node_modules/qjobs": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/qjobs/-/qjobs-1.2.0.tgz",
"integrity":
"sha512-8YOJEHtxpySA3fFDyCRxA+UUUV+fA+rTWnuWvyIOK/NCjhY+b4ocCtmu8TtsWb+m
YeU+GCHf/S66KZF/AsteKHg==",
  "dev": true,
  "engines": {
    "node": ">=0.9"
  }
},
"node_modules/qs": {
  "version": "6.13.0",
  "resolved": "https://registry.npmjs.org/qs/-/qs-6.13.0.tgz",
"integrity":
"sha512-+38qI9SO8tfZ4QmJNpIMUxqjbe7LKvvZgWdExBOmd+egZTtjLB67Gu0HRX3u/XO
q7UU2Nx6nsjvS16Z9uwfpg==",
  "dependencies": {
    "side-channel": "^1.0.6"
  },
  "engines": {
    "node": ">=0.6"
  },
  "funding": {
    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/queue-microtask": {
  "version": "1.2.3",
  "resolved": "https://registry.npmjs.org/queue-microtask/-/queue-microtask-1.2.3.tgz",
"integrity":
"sha512-NuaNSa6fIKT5JaSYQzJok04JzTL1CA6aGhv5rfLW3PggA+M2ChpZQnAC8h8i4ZFK
BS8X5RqkDBHA7r4hej3K9A==",
  "dev": true,
  "funding": [
    {
      "type": "github",
      "url": "https://github.com/sponsors/feross"
    },
    {
      "type": "patreon",
      "url": "https://www.patreon.com/feross"
    }
  ],

```

```

    {
      "type": "consulting",
      "url": "https://feross.org/support"
    }
  ]
},
"node_modules/randombytes": {
  "version": "2.1.0",
  "resolved": "https://registry.npmjs.org/randombytes/-/randombytes-2.1.0.tgz",
  "integrity":
"sha512-vYI3iOX+4CKUWuxGi9Uknie6fsqXqS9FE2Zaic4tNFD2N2QQaXOMFbuKK4QmDH
C0JO6B1Zp41J0LpT0oR68amQ==",
  "dev": true,
  "dependencies": {
    "safe-buffer": "^5.1.0"
  }
},
"node_modules/range-parser": {
  "version": "1.2.1",
  "resolved": "https://registry.npmjs.org/range-parser/-/range-parser-1.2.1.tgz",
  "integrity":
"sha512-Hrgsx+orqoygnmhFbKaHE6c296J+HTAQXoxEF6gNupROmmGJRozfG3ccAveqC
Brwr/2yxQ5BVd/GTI5agOwSg==",
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/raw-body": {
  "version": "2.5.2",
  "resolved": "https://registry.npmjs.org/raw-body/-/raw-body-2.5.2.tgz",
  "integrity":
"sha512-8zGqypfENjClqGhgXToC8aB2r7YrBX+AQAfIPs/Mlk+BtPTztOvTS01NRW/3Eh60J+
a48lt8qsCzirQ6loCVfA==",
  "dependencies": {
    "bytes": "3.1.2",
    "http-errors": "2.0.0",
    "iconv-lite": "0.4.24",
    "unpipe": "1.0.0"
  },
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/readable-stream": {
  "version": "3.6.2",
  "resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-3.6.2.tgz",

```



```

"integrity":
"sha512-9u/sniCrY3D5WdsERHzHE4G2YCXqoG5FTHUiCC4SIbr6XcLZBY05ya9EKjYek9O
5xOAwjGq+1JdGBAS7Q9ScoA==",
  "dev": true,
  "dependencies": {
    "inherits": "^2.0.3",
    "string_decoder": "^1.1.1",
    "util-deprecate": "^1.0.1"
  },
  "engines": {
    "node": ">= 6"
  }
},
"node_modules/readdirp": {
  "version": "3.6.0",
  "resolved": "https://registry.npmjs.org/readdirp/-/readdirp-3.6.0.tgz",
  "integrity":
"sha512-hOS089on8RduqdbhvQ5Z37A0ESjsqz6qnRcfsMU3495FuTdqSm+7bhJ29JvIOsB
DEEnan5DPu9t3To9VRIMzA==",
  "dev": true,
  "dependencies": {
    "picomatch": "^2.2.1"
  },
  "engines": {
    "node": ">=8.10.0"
  }
},
"node_modules/readdirp/node_modules/picomatch": {
  "version": "2.3.1",
  "resolved": "https://registry.npmjs.org/picomatch/-/picomatch-2.3.1.tgz",
  "integrity":
"sha512-JU3teHTNjmE2VCGFzuY8EXzCDVwEqB2a8fslvwaStHhAWJEEvd1o1QD80CU6+
ZdEXXSLbSsuLwJjkCBWqRQUVA==",
  "dev": true,
  "engines": {
    "node": ">=8.6"
  },
  "funding": {
    "url": "https://github.com/sponsors/jonschlinkert"
  }
},
"node_modules/reflect-metadata": {
  "version": "0.2.2",
  "resolved": "https://registry.npmjs.org/reflect-metadata/-/reflect-metadata-0.2.2.tgz",
  "integrity":
"sha512-urBwgfrvVP/eAyXx4hluJivBKzuEbSQs9rKWCrCkbSxNv8mxPcUZKeuoF3Uy4mJl3
Lwprp6yy5/39VWigZ4K6Q==",
  "dev": true

```

```

},
"node_modules/regenerate": {
  "version": "1.4.2",
  "resolved": "https://registry.npmjs.org/regenerate/-/regenerate-1.4.2.tgz",
  "integrity":
"sha512-zrceR/XhGYU/d/opr2EKO7aRHUeiBI8qjtfHqADTwZd6SzfY16la6kqD0MIUs5z5hx6A
aKa+PixpPrR289+I0A==",
  "dev": true
},
"node_modules/regenerate-unicode-properties": {
  "version": "10.1.1",
  "resolved":
"https://registry.npmjs.org/regenerate-unicode-properties/-/regenerate-unicode-properties-10.
1.1.tgz",
  "integrity":
"sha512-X007RyZLsCJVvrjgEFVpLUTZwyOZk3oiL75ZcuYjIIWd6rNJtOjkBwQc5AsRrpbKVk
xN6sklw/k/9m2jJYOf8Q==",
  "dev": true,
  "dependencies": {
    "regenerate": "^1.4.2"
  },
  "engines": {
    "node": ">=4"
  }
},
"node_modules/regenerator-runtime": {
  "version": "0.14.1",
  "resolved":
"https://registry.npmjs.org/regenerator-runtime/-/regenerator-runtime-0.14.1.tgz",
  "integrity":
"sha512-dYnhHh0nJoMfknZs6GmmhFknAGRrLznOu5nc9ML+EJxGvrx6H7teuevqVqCuPcP
K//3eDrrjQhehXVx9cnkGdw==",
  "dev": true
},
"node_modules/regenerator-transform": {
  "version": "0.15.2",
  "resolved":
"https://registry.npmjs.org/regenerator-transform/-/regenerator-transform-0.15.2.tgz",
  "integrity":
"sha512-hfMp2BoF0qOk3uc5V20ALGDS2ddjQaLrdl7xrGXvAlow7qeWRM2VA2HuCHkUKk9
slq3VwEwLNK3DFBqDfPGYtg==",
  "dev": true,
  "dependencies": {
    "@babel/runtime": "^7.8.4"
  }
},
"node_modules/regex-parser": {
  "version": "2.3.0",

```

```

    "resolved": "https://registry.npmjs.org/regex-parser/-/regex-parser-2.3.0.tgz",
    "integrity":
"sha512-TVILVSz2jY5D47F4mA4MppkBraFEaiUWJO/TcZHEIu13AqoZMkK1WMA4Om1YkY
bTx+9Ki1/tSUXbceyr9saRg==",
    "dev": true
  },
  "node_modules/regexpu-core": {
    "version": "5.3.2",
    "resolved": "https://registry.npmjs.org/regexpu-core/-/regexpu-core-5.3.2.tgz",
    "integrity":
"sha512-RAM5FIZz+Lhmo7db9L298p2vHP5ZywrVXmVXpmAD9GuL5MPH6t9ROw1iA/wfHk
Q76Qe7AaPF0nGuim96/lrQMQ==",
    "dev": true,
    "dependencies": {
      "@babel/regjsgen": "^0.8.0",
      "regenerate": "^1.4.2",
      "regenerate-unicode-properties": "^10.1.0",
      "regjsparser": "^0.9.1",
      "unicode-match-property-ecmascript": "^2.0.0",
      "unicode-match-property-value-ecmascript": "^2.1.0"
    },
    "engines": {
      "node": ">=4"
    }
  },
  "node_modules/regjsparser": {
    "version": "0.9.1",
    "resolved": "https://registry.npmjs.org/regjsparser/-/regjsparser-0.9.1.tgz",
    "integrity":
"sha512-dQUtn90WanSNI+7mQKcXAgZxvUe7Z0SqXlgzv0za4LwiUhyzBC58yQO3liFoUgu8
GiJVInAhJjkj1N0EtQ5nkQ==",
    "dev": true,
    "dependencies": {
      "jsesc": "~0.5.0"
    },
    "bin": {
      "regjsparser": "bin/parser"
    }
  },
  "node_modules/regjsparser/node_modules/jsesc": {
    "version": "0.5.0",
    "resolved": "https://registry.npmjs.org/jsesc/-/jsesc-0.5.0.tgz",
    "integrity":
"sha512-uZz5UnB7u4T9LvwfFqXii7pZSouaRPorGs5who1lp7VO0wxanFvBL7GkM6dTHlgX
+jhBApRetaWpnDabOeTcnA==",
    "dev": true,
    "bin": {
      "jsesc": "bin/jsesc"
    }
  }

```

```

    }
  },
  "node_modules/require-directory": {
    "version": "2.1.1",
    "resolved": "https://registry.npmjs.org/require-directory/-/require-directory-2.1.1.tgz",
    "integrity":
"sha512-fGxEI7+wsG9xrvdjsrlmL22OMTTiHRwAMroiEeMgq8gzoLC/PQr7RsRDSTLUg/bZA
ZtF+TVIkHc6/4RIKrui+Q==",
    "dev": true,
    "engines": {
      "node": ">=0.10.0"
    }
  },
  "node_modules/require-from-string": {
    "version": "2.0.2",
    "resolved": "https://registry.npmjs.org/require-from-string/-/require-from-string-2.0.2.tgz",
    "integrity":
"sha512-Xf0nWe6RseziFMu+Ap9biiUbmplq6S9/p+7w7YXP/JBHhrUDDUhwa+vANyubuqfZ
WTveU//DYVGsDG7RKL/vEw==",
    "dev": true,
    "engines": {
      "node": ">=0.10.0"
    }
  },
  "node_modules/requires-port": {
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/requires-port/-/requires-port-1.0.0.tgz",
    "integrity":
"sha512-KigOCHcocU3XODJxsu8i/j8T9tzT4adHiecwORRQ0ZZFcp7ahwXuRU1m+yuO90C
5ZUyGeGfocHDI14M3L3yDAQ==",
    "dev": true
  },
  "node_modules/resolve": {
    "version": "1.22.8",
    "resolved": "https://registry.npmjs.org/resolve/-/resolve-1.22.8.tgz",
    "integrity":
"sha512-oKWePCxqpd6FILvGV1VU0x7bkPmmCNolxzMf4NczoDnQciWRAF+cPtZn5i6n+Rf
D2d9i0tZpKnG6Yk168ylyw==",
    "dev": true,
    "dependencies": {
      "is-core-module": "^2.13.0",
      "path-parse": "^1.0.7",
      "supports-preserve-symlinks-flag": "^1.0.0"
    },
    "bin": {
      "resolve": "bin/resolve"
    }
  },
  "funding": {

```

```

    "url": "https://github.com/sponsors/ljharb"
  }
},
"node_modules/resolve-from": {
  "version": "4.0.0",
  "resolved": "https://registry.npmjs.org/resolve-from/-/resolve-from-4.0.0.tgz",
  "integrity":
"sha512-pb/MYmXstAkysRFx8piNI1tGFNQIFA3vkE3Gq4EuA1dF6gHp/+vgZqsCGJapvy8N3
Q+4o7FwvquPJcnZ7RYy4g==",
  "dev": true,
  "engines": {
    "node": ">=4"
  }
},
"node_modules/resolve-url-loader": {
  "version": "5.0.0",
  "resolved": "https://registry.npmjs.org/resolve-url-loader/-/resolve-url-loader-5.0.0.tgz",
  "integrity":
"sha512-uZtduh8/8srhBoMx//5bwqjQ+rfYOUq8zC9NrMUGtjBiGTtFJM42s58/36+hTqeqlNcn
Ye08Nj3LkK9IW4N8Xg==",
  "dev": true,
  "dependencies": {
    "adjust-sourcemap-loader": "^4.0.0",
    "convert-source-map": "^1.7.0",
    "loader-utils": "^2.0.0",
    "postcss": "^8.2.14",
    "source-map": "0.6.1"
  },
  "engines": {
    "node": ">=12"
  }
},
"node_modules/resolve-url-loader/node_modules/loader-utils": {
  "version": "2.0.4",
  "resolved": "https://registry.npmjs.org/loader-utils/-/loader-utils-2.0.4.tgz",
  "integrity":
"sha512-xXqpXoINfFhgua9xiqD8fPFHgkoq1mmmpE92WIDbm9rNRd/EbRb+Gqf908T2DMfu
HjjJlksiK2RbHVOD/MqSw==",
  "dev": true,
  "dependencies": {
    "big.js": "^5.2.2",
    "emojis-list": "^3.0.0",
    "json5": "^2.1.2"
  },
  "engines": {
    "node": ">=8.9.0"
  }
},

```

```

"node_modules/resolve-url-loader/node_modules/source-map": {
  "version": "0.6.1",
  "resolved": "https://registry.npmjs.org/source-map/-/source-map-0.6.1.tgz",
  "integrity":
"sha512-UjgapumWlbMhkBgzT7Ykc5YXUT46F0iKu8SGXq0bcwP5dz/h0Plj6enJqjz1Zbq2I5
WaqYnrVbwWOWMyF3F47g==",
  "dev": true,
  "engines": {
    "node": ">=0.10.0"
  }
},
"node_modules/restore-cursor": {
  "version": "5.1.0",
  "resolved": "https://registry.npmjs.org/restore-cursor/-/restore-cursor-5.1.0.tgz",
  "integrity":
"sha512-oMA2dcrw6u0YfxJQXm342bFKX/E4sG9rbTzO9ptUcR/e8A33cHuvStiYOWH7fszkZI
Z1z/ta9AAoPk2F4qIOHA==",
  "dev": true,
  "dependencies": {
    "onetime": "^7.0.0",
    "signal-exit": ^4.1.0
  },
  "engines": {
    "node": ">=18"
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
},
"node_modules/retry": {
  "version": "0.12.0",
  "resolved": "https://registry.npmjs.org/retry/-/retry-0.12.0.tgz",
  "integrity":
"sha512-9LkiTwyUh6rT555DtE9rTX+BKByPfrMzEAtnIEtdEwr3Nkffwiihqe2bWADg+OQRjt9gl
6ICdmB/ZFDcGAtSow==",
  "dev": true,
  "engines": {
    "node": ">= 4"
  }
},
"node_modules/reusify": {
  "version": "1.0.4",
  "resolved": "https://registry.npmjs.org/reusify/-/reusify-1.0.4.tgz",
  "integrity":
"sha512-U99h4uQmXJytT818vhV9G0f9vMwZf8Z2UJ1LxUu1dJXmU0iXGjWj2vUjgkz1R10eS0C
7/uwyaLVT6fbtCwTyzw==",
  "dev": true,
  "engines": {

```

```

    "iojs": ">=1.0.0",
    "node": ">=0.10.0"
  }
},
"node_modules/rfdc": {
  "version": "1.4.1",
  "resolved": "https://registry.npmjs.org/rfdc/-/rfdc-1.4.1.tgz",
  "integrity":
"sha512-q1b3N5QkRUWUI7iyyLaaj3kOpIT0N2i9MqIEQXP73GVsN9cw3fdx8X63cEmWhJGi
2PPCF23ljp7ktmd39rawIA==",
  "dev": true
},
"node_modules/rimraf": {
  "version": "3.0.2",
  "resolved": "https://registry.npmjs.org/rimraf/-/rimraf-3.0.2.tgz",
  "integrity":
"sha512-JZkJMZkAGFFPP2YqXZXPbMIMBgsxzE8ILs4IMIX/2o0L9UBw9O/Y3o6wFw/i9YLa
pcUJWwqbi3kdxlPdC62TIA==",
  "deprecated": "Rimraf versions prior to v4 are no longer supported",
  "dev": true,
  "dependencies": {
    "glob": "^7.1.3"
  },
  "bin": {
    "rimraf": "bin.js"
  },
  "funding": {
    "url": "https://github.com/sponsors/isaacs"
  }
},
"node_modules/rollup": {
  "version": "4.18.0",
  "resolved": "https://registry.npmjs.org/rollup/-/rollup-4.18.0.tgz",
  "integrity":
"sha512-QmJz14PX3rzbJCN1SG4Xe/bAAX2a6NpCP8ab2vfu2GiUr8AQcr2nCV/oEO3yneFa
rB67zk8ShllyWb2LGTb3Sg==",
  "dev": true,
  "dependencies": {
    "@types/estree": "1.0.5"
  },
  "bin": {
    "rollup": "dist/bin/rollup"
  },
  "engines": {
    "node": ">=18.0.0",
    "npm": ">=8.0.0"
  },
  "optionalDependencies": {

```

```

"@rollup/rollup-android-arm-eabi": "4.18.0",
"@rollup/rollup-android-arm64": "4.18.0",
"@rollup/rollup-darwin-arm64": "4.18.0",
"@rollup/rollup-darwin-x64": "4.18.0",
"@rollup/rollup-linux-arm-gnueabihf": "4.18.0",
"@rollup/rollup-linux-arm-musleabihf": "4.18.0",
"@rollup/rollup-linux-arm64-gnu": "4.18.0",
"@rollup/rollup-linux-arm64-musl": "4.18.0",
"@rollup/rollup-linux-powerpc64le-gnu": "4.18.0",
"@rollup/rollup-linux-riscv64-gnu": "4.18.0",
"@rollup/rollup-linux-s390x-gnu": "4.18.0",
"@rollup/rollup-linux-x64-gnu": "4.18.0",
"@rollup/rollup-linux-x64-musl": "4.18.0",
"@rollup/rollup-win32-arm64-msvc": "4.18.0",
"@rollup/rollup-win32-ia32-msvc": "4.18.0",
"@rollup/rollup-win32-x64-msvc": "4.18.0",
"fs-events": "~2.3.2"
}
},
"node_modules/run-applescript": {
  "version": "7.0.0",
  "resolved": "https://registry.npmjs.org/run-applescript/-/run-applescript-7.0.0.tgz",
  "integrity":
"sha512-9by4Ij99JUur/MCFBUkDKLWK3G9HVXmabKz9U5MIIAluvuzkiOicRYS8XJLxX+xahD
+mLiiCYDqF9dKAgtzKP1A==",
  "dev": true,
  "engines": {
    "node": ">=18"
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
},
"node_modules/run-parallel": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/run-parallel/-/run-parallel-1.2.0.tgz",
  "integrity":
"sha512-514uqkX8WxIw0C1uEP0BXq6AR3hBY9V+Q62fy0VbhEYjL83MwWwY8yv8rF7o2nz8gqi3ZtX16xkxsXoYw==",
  "dev": true,
  "funding": [
    {
      "type": "github",
      "url": "https://github.com/sponsors/feross"
    },
    {
      "type": "patreon",
      "url": "https://www.patreon.com/feross"
    }
  ]
}

```



```

    },
    {
      "type": "consulting",
      "url": "https://feross.org/support"
    }
  ],
  "dependencies": {
    "queue-microtask": "^1.2.2"
  }
},
"node_modules/rxjs": {
  "version": "7.8.1",
  "resolved": "https://registry.npmjs.org/rxjs/-/rxjs-7.8.1.tgz",
  "integrity":
"sha512-AA3TVj+0A2iuIoQkWEK/tqFjBq2j+6PO6Y0zJcvzLAFhEFIO3HL0vls9hWLnCZbAAb
K0mar7oZ4V079I/qPMxg==",
  "dependencies": {
    "tslib": "^2.1.0"
  }
},
"node_modules/safe-buffer": {
  "version": "5.2.1",
  "resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.2.1.tgz",
  "integrity":
"sha512-rp3So07KcdmmKbGvgaNxQsJr7bGVSVk5S9Eq1F+ppbRo70+YeaDxkw5Dd8NPN
+GD6bjnYm2VuPuCXmpuYvmCXQ==",
  "funding": [
    {
      "type": "github",
      "url": "https://github.com/sponsors/feross"
    },
    {
      "type": "patreon",
      "url": "https://www.patreon.com/feross"
    },
    {
      "type": "consulting",
      "url": "https://feross.org/support"
    }
  ]
},
"node_modules/safer-buffer": {
  "version": "2.1.2",
  "resolved": "https://registry.npmjs.org/safer-buffer/-/safer-buffer-2.1.2.tgz",
  "integrity":
"sha512-YZo3K82SD7Riyi0E1EQPojLz7kpepnSQI9IyPbHHg1XXXevb5dJI7tpyN2ADxGcQb
HG7vcyRHk0cbwqcQriUtg=="
},

```

```

"node_modules/sass": {
  "version": "1.77.6",
  "resolved": "https://registry.npmjs.org/sass/-/sass-1.77.6.tgz",
  "integrity":
"sha512-ByXE1oLD79GVq9Ht1PeHWCPMPB8XHpBuz1r85oByKHjZY6qV6rWnQovQzXJXu
Q/XyE1Oj3iPk3lo28uzaRA2/Q==",
  "dev": true,
  "dependencies": {
    "chokidar": ">=3.0.0 <4.0.0",
    "immutable": "^4.0.0",
    "source-map-js": ">=0.6.2 <2.0.0"
  },
  "bin": {
    "sass": "sass.js"
  },
  "engines": {
    "node": ">=14.0.0"
  }
},
"node_modules/sass-loader": {
  "version": "14.2.1",
  "resolved": "https://registry.npmjs.org/sass-loader/-/sass-loader-14.2.1.tgz",
  "integrity":
"sha512-G0VcnMYU18a4N7VoNDegg2OuMjYtxnqzQWARVWCIVSZwJeiL9kg8QMsulZOpl
JgTzZLF6jGxI3AClj8I9nRdQ==",
  "dev": true,
  "dependencies": {
    "neo-async": "^2.6.2"
  },
  "engines": {
    "node": ">= 18.12.0"
  },
  "funding": {
    "type": "opencollective",
    "url": "https://opencollective.com/webpack"
  },
  "peerDependencies": {
    "@rspack/core": "0.x || 1.x",
    "node-sass": "^4.0.0 || ^5.0.0 || ^6.0.0 || ^7.0.0 || ^8.0.0 || ^9.0.0",
    "sass": "^1.3.0",
    "sass-embedded": "*",
    "webpack": "^5.0.0"
  },
  "peerDependenciesMeta": {
    "@rspack/core": {
      "optional": true
    }
  },
  "node-sass": {

```

```

    "optional": true
  },
  "sass": {
    "optional": true
  },
  "sass-embedded": {
    "optional": true
  },
  "webpack": {
    "optional": true
  }
},
"node_modules/sax": {
  "version": "1.4.1",
  "resolved": "https://registry.npmjs.org/sax/-/sax-1.4.1.tgz",
  "integrity":
"sha512-aWOz7yVScEGoKNd4PA10LZ8sk0A/z5+nXQG5giUO5rprX9jgYsT dov9qCchZiPIZ
ezbZH+jRut8nPodFAX4Jg==",
  "dev": true,
  "optional": true
},
"node_modules/schema-utils": {
  "version": "4.2.0",
  "resolved": "https://registry.npmjs.org/schema-utils/-/schema-utils-4.2.0.tgz",
  "integrity":
"sha512-L0jRsrPpj dckP3oPug3/VxNKt2trR8TcabrM6FOAAIvC/9Phcmm+cuAgTlxBqdBR1W
Jx7Naj9WHw+aOmheSVbw==",
  "dev": true,
  "dependencies": {
    "@types/json-schema": "^7.0.9",
    "ajv": "^8.9.0",
    "ajv-formats": "^2.1.1",
    "ajv-keywords": "^5.1.0"
  },
  "engines": {
    "node": ">= 12.13.0"
  },
  "funding": {
    "type": "opencollective",
    "url": "https://opencollective.com/webpack"
  }
},
"node_modules/schema-utils/node_modules/ajv-formats": {
  "version": "2.1.1",
  "resolved": "https://registry.npmjs.org/ajv-formats/-/ajv-formats-2.1.1.tgz",

```

```

"integrity":
"sha512-Wx0Kx52hxE7C18hkMEggYIEifqWZtYaRgouJor+WmDpNqyEK13vgEWyVNup7So
eeoLMsr4kf5h6dOW11115MUA==",
  "dev": true,
  "dependencies": {
    "ajv": "^8.0.0"
  },
  "peerDependencies": {
    "ajv": "^8.0.0"
  },
  "peerDependenciesMeta": {
    "ajv": {
      "optional": true
    }
  }
},
"node_modules/select-hose": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/select-hose/-/select-hose-2.0.0.tgz",
"integrity":
"sha512-mEugaLK+YfkijB4fx0e6klmuJdClT2LxCRcbEYPqRGCs4F2ogyfZU5IAZRdjCP8JPq
2AtdNoC/Dux63d9Kiryg==",
  "dev": true
},
"node_modules/selfsigned": {
  "version": "2.4.1",
  "resolved": "https://registry.npmjs.org/selfsigned/-/selfsigned-2.4.1.tgz",
"integrity":
"sha512-th5B4L2U+eGLq1TVh7zNRGBapioSORUeymlydxgFpwww9d2qyKvtuPU2jJuHvYA
wwqi2Y596QBL3eEqcPEYL8Q==",
  "dev": true,
  "dependencies": {
    "@types/node-forge": "^1.3.0",
    "node-forge": "^1"
  },
  "engines": {
    "node": ">=10"
  }
},
"node_modules/semver": {
  "version": "7.6.2",
  "resolved": "https://registry.npmjs.org/semver/-/semver-7.6.2.tgz",
"integrity":
"sha512-FNAIBWCx9qcRhoHcgCj0gvU7SN1IYU2ZXuSfl04bSC5OpvDHFyJCjdNHomPXjQ
ICBU67YW64PzY7/VIEH7F2w==",
  "dev": true,
  "bin": {
    "semver": "bin/semver.js"
  }
}

```

```

    },
    "engines": {
      "node": ">=10"
    }
  },
  "node_modules/send": {
    "version": "0.19.0",
    "resolved": "https://registry.npmjs.org/send/-/send-0.19.0.tgz",
    "integrity":
"sha512-dW41u5VfLXu8SJh5bwRmyYUbaAoSB3c9uQh6L8h/KtsFREPWpbX1IrljJo186Jc4n
mci/sGUZ9a0a0J2zgfq2hw==",
    "dependencies": {
      "debug": "2.6.9",
      "depd": "2.0.0",
      "destroy": "1.2.0",
      "encodeurl": "~1.0.2",
      "escape-html": "~1.0.3",
      "etag": "~1.8.1",
      "fresh": "0.5.2",
      "http-errors": "2.0.0",
      "mime": "1.6.0",
      "ms": "2.1.3",
      "on-finished": "2.4.1",
      "range-parser": "~1.2.1",
      "statuses": "2.0.1"
    },
    "engines": {
      "node": ">= 0.8.0"
    }
  },
  "node_modules/send/node_modules/debug": {
    "version": "2.6.9",
    "resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",
    "integrity":
"sha512-bC7ElrdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUcNowp6cvspg0jXznRTKDJm/E7A
dgFBVeAPVMNcKGSHEMA==",
    "dependencies": {
      "ms": "2.0.0"
    }
  },
  "node_modules/send/node_modules/debug/node_modules/ms": {
    "version": "2.0.0",
    "resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",
    "integrity":
"sha512-Tpp60P6IUJDTuOq/5Z8cdskzJujfwqfOTkrwIwj7IRISpnkJnT6SyJ4PCPnGMoFjC9d
dhal5KVIYtAt97ix05A=="
  },
  "node_modules/send/node_modules/mime": {

```

```

"version": "1.6.0",
"resolved": "https://registry.npmjs.org/mime/-/mime-1.6.0.tgz",
"integrity":
"sha512-x0Vn8spl+wuJ1O6S7gnbaQg8Pxb4NNHb7KSINmEWKiPE4RKOpIvijn+NkmYmmR
gP68mc70j2EbeTFRsrswaQeg==",
  "bin": {
    "mime": "cli.js"
  },
  "engines": {
    "node": ">=4"
  }
},
"node_modules/send/node_modules/ms": {
  "version": "2.1.3",
  "resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",
  "integrity":
"sha512-6FizubTLZG3J2a/NVCAleEhjq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9
xAOtHnSO/tXtF3WRTIA==",
  "node_modules/send/node_modules/statuses": {
    "version": "2.0.1",
    "resolved": "https://registry.npmjs.org/statuses/-/statuses-2.0.1.tgz",
    "integrity":
"sha512-RwNA9Z/7PrK06rYLlZFMlaF+I73iwpzsqRIFgbMLbTcLD6cOao82TaWefPXQvB2fO
C4AjuYSEndS7N/mTCbkdQ==",
    "engines": {
      "node": ">= 0.8"
    }
  },
  "node_modules/serialize-javascript": {
    "version": "6.0.2",
    "resolved": "https://registry.npmjs.org/serialize-javascript/-/serialize-javascript-6.0.2.tgz",
    "integrity":
"sha512-Saa1xPByTTq2gdeFZYLLo+RFE35NHZkAbqZeWNd3BpzppeVisAqpDjcp8dyf6ulv
EqJRd46jemmyA4iFleV/k8g==",
    "dev": true,
    "dependencies": {
      "randombytes": "^2.1.0"
    }
  },
  "node_modules/serve-index": {
    "version": "1.9.1",
    "resolved": "https://registry.npmjs.org/serve-index/-/serve-index-1.9.1.tgz",
    "integrity":
"sha512-pXHfKNP4qujrteMrSBb0rc8HJ9Ms/GrXwcUtUtD5s4ewDJI8bT3Cz2zTVRMKtri49p
Lx2e0Ya8ziP5Ya2pZZw==",
    "dev": true,
    "dependencies": {

```

```

    "accepts": "~1.3.4",
    "batch": "0.6.1",
    "debug": "2.6.9",
    "escape-html": "~1.0.3",
    "http-errors": "~1.6.2",
    "mime-types": "~2.1.17",
    "parseurl": "~1.3.2"
  },
  "engines": {
    "node": ">= 0.8.0"
  }
},
"node_modules/serve-index/node_modules/debug": {
  "version": "2.6.9",
  "resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",
  "integrity":
"sha512-bC7E1rdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUcNowp6cvspg0jXznRTKDJm/E7A
dgFBVeAPVMNcKGsHMA==",
  "dev": true,
  "dependencies": {
    "ms": "2.0.0"
  }
},
"node_modules/serve-index/node_modules/depd": {
  "version": "1.1.2",
  "resolved": "https://registry.npmjs.org/depd/-/depd-1.1.2.tgz",
  "integrity":
"sha512-7emPTI6Dpo6JRXOXjLRxck+FILRX5847cLKEn00PLAgc3g2hTZZgr+e4c2v6QpSm
LeFP3n5yUo7ft6avBK/5jQ==",
  "dev": true,
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/serve-index/node_modules/http-errors": {
  "version": "1.6.3",
  "resolved": "https://registry.npmjs.org/http-errors/-/http-errors-1.6.3.tgz",
  "integrity":
"sha512-lks+IVC8dgGyh97jvxveYTWQFvh4uw4yC12gVI63Cg30sjPX4wuGcdkICVXDAESr6
OJGjqGA8Iz5mkeN6zID7A==",
  "dev": true,
  "dependencies": {
    "depd": "~1.1.2",
    "inherits": "2.0.3",
    "setprototypeof": "1.1.0",
    "statuses": ">= 1.4.0 < 2"
  }
},
"engines": {

```

```

    "node": ">= 0.6"
  }
},
"node_modules/serve-index/node_modules/inherits": {
  "version": "2.0.3",
  "resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.3.tgz",
  "integrity":
"sha512-x00IRNXNy63jwGkJmzPigoySHbaqpNuzKbBOMzK+g2OdZpQ9w+sxCN+VSB3ja7I
Age2OP2qpfXTjeNcyjmW1uw==",
  "dev": true
},
"node_modules/serve-index/node_modules/ms": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",
  "integrity":
"sha512-Tpp60P6IUJDTuOq/5Z8cdskzJujfwqfOTkrwIwj7IRISpnkJnT6SyJ4PCPnGMoFjC9d
dhal5KVIYtAt97ix05A==",
  "dev": true
},
"node_modules/serve-index/node_modules/setprototypeof": {
  "version": "1.1.0",
  "resolved": "https://registry.npmjs.org/setprototypeof/-/setprototypeof-1.1.0.tgz",
  "integrity":
"sha512-BvE/TwpZX4FXExxOxZyRGQQv651MSwmWKZGqvmPcRijDqWub67kTKuIMx43c
ZZrS/cBBzwBcNDWoFxt2XEFIpQ==",
  "dev": true
},
"node_modules/serve-static": {
  "version": "1.16.2",
  "resolved": "https://registry.npmjs.org/serve-static/-/serve-static-1.16.2.tgz",
  "integrity":
"sha512-VqpjJZKadQB/PEbEwwFdO43Ax5dFBZ2UECszz8bQ7pi7wt//PWe1P6MN7eCnjsat
YtBT6EuiClbjSWP2WrloTw==",
  "dependencies": {
    "encodeurl": "~2.0.0",
    "escape-html": "~1.0.3",
    "parseurl": "~1.3.3",
    "send": "0.19.0"
  },
  "engines": {
    "node": ">= 0.8.0"
  }
},
"node_modules/serve-static/node_modules/encodeurl": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-2.0.0.tgz",

```



```

"integrity":
"sha512-Q0n9HRi4m6JuGIV1eFlmvJB7ZEVxu93lrMyiMsGC0lrMJMWzRgx6WGquyfQgZVb
31vhGgXnfmPNNXmxnOkRBrg==",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/set-function-length": {
  "version": "1.2.2",
  "resolved": "https://registry.npmjs.org/set-function-length/-/set-function-length-1.2.2.tgz",
  "integrity":
"sha512-pgRc4hJ4/sNjWCSS9AmnS40x3bNMDTknHgL5UaMBTMyJnU90EgWh1Rz+MC9e
Fu4BuN/UwZjKQuY/1v3rM7HMfg==",
  "dependencies": {
    "define-data-property": "^1.1.4",
    "es-errors": "^1.3.0",
    "function-bind": "^1.1.2",
    "get-intrinsic": "^1.2.4",
    "gopd": "^1.0.1",
    "has-property-descriptors": "^1.0.2"
  },
  "engines": {
    "node": ">= 0.4"
  }
},
"node_modules/setprototypeof": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/setprototypeof/-/setprototypeof-1.2.0.tgz",
  "integrity":
"sha512-E5LDX7Wrp85Kil5bhZv46j8jOeboKq5JMmYM3gVGdGH8xFpPWXUMsNriODCrko
xMEeNi/XZlWuRvY4XNwYmJpw=="
},
"node_modules/shallow-clone": {
  "version": "3.0.1",
  "resolved": "https://registry.npmjs.org/shallow-clone/-/shallow-clone-3.0.1.tgz",
  "integrity":
"sha512-/6KqX+GVUdqPuPPd2LxDDxzX6CAbjJehAAOKINpqqUpAqPM6HeL8f+o3a+JsyGj
n2lv0WY8UsTgUjU9Ok55NA=="
},
"dev": true,
"dependencies": {
  "kind-of": "^6.0.2"
},
"engines": {
  "node": ">=8"
}
},
"node_modules/shebang-command": {
  "version": "2.0.0",

```

```

    "resolved": "https://registry.npmjs.org/shebang-command/-/shebang-command-2.0.0.tgz",
    "integrity":
"sha512-kHxr2zZpYtdmrN1qDjrrX/Z1rR1kG8Dx+gkpK1G4eXmvXswmcE1hTWBWWYUzIraYw
1/yZp6YuDY77YtvbN0dmDA==",
    "dev": true,
    "dependencies": {
      "shebang-regex": "^3.0.0"
    },
    "engines": {
      "node": ">=8"
    }
  },
  "node_modules/shebang-regex": {
    "version": "3.0.0",
    "resolved": "https://registry.npmjs.org/shebang-regex/-/shebang-regex-3.0.0.tgz",
    "integrity":
"sha512-7++dFhtcx3353uBaq8DDR4NuxBetBzC7ZQOhmTQInHEd6bSrXdiEyzCvG07Z44U
YdLShWUyXt5M/yz8ekcb1A==",
    "dev": true,
    "engines": {
      "node": ">=8"
    }
  },
  "node_modules/shell-quote": {
    "version": "1.8.1",
    "resolved": "https://registry.npmjs.org/shell-quote/-/shell-quote-1.8.1.tgz",
    "integrity":
"sha512-6j1W9l1iAs/4xYBI1SYOVZyFcCis9b4KCLQ8fgAGG07QvzaRLVVRQvAy85yNmmZ
SjYjg4MWh4gNvlPujU/5LpA==",
    "dev": true,
    "funding": {
      "url": "https://github.com/sponsors/ljharb"
    }
  },
  "node_modules/side-channel": {
    "version": "1.0.6",
    "resolved": "https://registry.npmjs.org/side-channel/-/side-channel-1.0.6.tgz",
    "integrity":
"sha512-fDW/EZ6Q9RiO8eFG8Hj+7u/oW+XrPTIChwCOM2+th2A6ObIdtYYIpve9m+Kvl9Z4
C9qSEXlaGR6bTEYHReuglA==",
    "dependencies": {
      "call-bind": "^1.0.7",
      "es-errors": "^1.3.0",
      "get-intrinsic": "^1.2.4",
      "object-inspect": "^1.13.1"
    },
    "engines": {
      "node": ">= 0.4"
    }
  }

```

```

    },
    "funding": {
      "url": "https://github.com/sponsors/ljharb"
    }
  },
  "node_modules/signal-exit": {
    "version": "4.1.0",
    "resolved": "https://registry.npmjs.org/signal-exit/-/signal-exit-4.1.0.tgz",
    "integrity":
"sha512-bzyZ1e88w9O1iNjBKnOlvyTrWPDI46O1bG0D3XInv+9tkPrxrN8jUUTiFIDkkmKWgn
1M6CflA13SuGqOa9Korw==",
    "dev": true,
    "engines": {
      "node": ">=14"
    },
    "funding": {
      "url": "https://github.com/sponsors/isaacs"
    }
  },
  "node_modules/sigstore": {
    "version": "2.3.1",
    "resolved": "https://registry.npmjs.org/sigstore/-/sigstore-2.3.1.tgz",
    "integrity":
"sha512-8G+/XDU8wNsJOQS5ysDVO0Etg9/2uA5gR9I4ZwijlwxBcrU6RPfwi2+jJmbP+Ap1H
lp/nVAaEO4Fj22/SL2gQ==",
    "dev": true,
    "dependencies": {
      "@sigstore/bundle": "^2.3.2",
      "@sigstore/core": "^1.0.0",
      "@sigstore/protobuf-specs": "^0.3.2",
      "@sigstore/sign": "^2.3.2",
      "@sigstore/tuf": "^2.3.4",
      "@sigstore/verify": "^1.2.1"
    },
    "engines": {
      "node": "^16.14.0 || >=18.0.0"
    }
  },
  "node_modules/slash": {
    "version": "5.1.0",
    "resolved": "https://registry.npmjs.org/slash/-/slash-5.1.0.tgz",
    "integrity":
"sha512-ZA6oR3T/pEyuqwMgAKT0/hAv8oAXckzbkmR0UkUosQ+Mc4RxGoJkRmwHgHufa
enlyAgE1Mxgpdcf75y6XcnDg==",
    "dev": true,
    "engines": {
      "node": ">=14.16"
    }
  },

```

```

    "funding": {
      "url": "https://github.com/sponsors/sindresorhus"
    }
  },
  "node_modules/slice-ansi": {
    "version": "5.0.0",
    "resolved": "https://registry.npmjs.org/slice-ansi/-/slice-ansi-5.0.0.tgz",
    "integrity":
"sha512-FC+lgizVPfie0kkhqUScwRu1O/IF6NOgJmICgK+/LYxDCTk8sGelYaHDhFcDN+Sn3
Cv+3VSa4Byeo+IMCzpMgQ==",
    "dev": true,
    "dependencies": {
      "ansi-styles": "^6.0.0",
      "is-fullwidth-code-point": "^4.0.0"
    },
    "engines": {
      "node": ">=12"
    },
    "funding": {
      "url": "https://github.com/chalk/slice-ansi?sponsor=1"
    }
  },
  "node_modules/slice-ansi/node_modules/ansi-styles": {
    "version": "6.2.1",
    "resolved": "https://registry.npmjs.org/ansi-styles/-/ansi-styles-6.2.1.tgz",
    "integrity":
"sha512-bN798gFfQX+viw3R7yrGWRqnrN2oRkEkUjJl4JNn4E8GxxbjtG3FbrEIIY3I8/hrwUw1
eCZvi4QuOTP4MErVug==",
    "dev": true,
    "engines": {
      "node": ">=12"
    },
    "funding": {
      "url": "https://github.com/chalk/ansi-styles?sponsor=1"
    }
  },
  "node_modules/smart-buffer": {
    "version": "4.2.0",
    "resolved": "https://registry.npmjs.org/smart-buffer/-/smart-buffer-4.2.0.tgz",
    "integrity":
"sha512-94hK0Hh8rPqQl2xXc3HsaBoOXKV20MToPkcXvwblSWLEs+64sBq5kFgn2kJDHb1
Pry9yrP0dxrCI9RRci7RXKg==",
    "dev": true,
    "engines": {
      "node": ">= 6.0.0",
      "npm": ">= 3.0.0"
    }
  },

```

```

"node_modules/socket.io": {
  "version": "4.7.5",
  "resolved": "https://registry.npmjs.org/socket.io/-/socket.io-4.7.5.tgz",
  "integrity":
"sha512-DmeAkF6cwM9jSfmp6Dr/5/mfMwb5Z5qRrSXLpo3Fq5SqyU8CMF15jIN4ZhfSwu35
ksM1qmHZDQ/DK5XTccSTvA==",
  "dev": true,
  "dependencies": {
    "accepts": "~1.3.4",
    "base64id": "~2.0.0",
    "cors": "~2.8.5",
    "debug": "~4.3.2",
    "engine.io": "~6.5.2",
    "socket.io-adapter": "~2.5.2",
    "socket.io-parser": "~4.2.4"
  },
  "engines": {
    "node": ">=10.2.0"
  }
},
"node_modules/socket.io-adapter": {
  "version": "2.5.5",
  "resolved": "https://registry.npmjs.org/socket.io-adapter/-/socket.io-adapter-2.5.5.tgz",
  "integrity":
"sha512-eLDQas5dzPgOWCk9GuuJC2IBqItuhKI4uxGgo9aIV7MYbk2h9Q6uULEh8WBzTho
I71+qU9Ast9fVUmkqPP9wYg==",
  "dev": true,
  "dependencies": {
    "debug": "~4.3.4",
    "ws": "~8.17.1"
  }
},
"node_modules/socket.io-parser": {
  "version": "4.2.4",
  "resolved": "https://registry.npmjs.org/socket.io-parser/-/socket.io-parser-4.2.4.tgz",
  "integrity":
"sha512-/GbIkmo8ioc+NIWIhwdecY0ge+qVBSMdgxGygevmdHj24bsfgtCmcUUcQ5ZzcyIGF
HsN3k4HB4CgkI96KVnuew==",
  "dev": true,
  "dependencies": {
    "@socket.io/component-emitter": "~3.1.0",
    "debug": "~4.3.1"
  },
  "engines": {
    "node": ">=10.0.0"
  }
},
"node_modules/sockjs": {

```

```

"version": "0.3.24",
"resolved": "https://registry.npmjs.org/sockjs/-/sockjs-0.3.24.tgz",
"integrity":
"sha512-GJgLTZ7vYb/JtPSSZ10hsOYIvEYsJbNU+zPdIHcUaWVNUEPivzxku31865sSSud0
Da0W4IEeOPImw93zLQchuQ==",
  "dev": true,
  "dependencies": {
    "faye-websocket": "^0.11.3",
    "uuid": "^8.3.2",
    "websocket-driver": "^0.7.4"
  }
},
"node_modules/socks": {
  "version": "2.8.3",
  "resolved": "https://registry.npmjs.org/socks/-/socks-2.8.3.tgz",
  "integrity":
"sha512-l5x7VUWbjVFbafGLxPWkYsHlhEvmF85tbleFZWc8ZPtoMyybuEhL7Jye/ooC4/d4
8FgOjSJXgsF/AJPYCW8Zw==",
  "dev": true,
  "dependencies": {
    "ip-address": "^9.0.5",
    "smart-buffer": "^4.2.0"
  },
  "engines": {
    "node": ">= 10.0.0",
    "npm": ">= 3.0.0"
  }
},
"node_modules/socks-proxy-agent": {
  "version": "8.0.4",
  "resolved": "https://registry.npmjs.org/socks-proxy-agent/-/socks-proxy-agent-8.0.4.tgz",
  "integrity":
"sha512-GNAq/eg8Udq2x0eNiFkr9gRg5bA7PXEWagQdeRX4cPSG+X/8V38v637gim9bjFpt
Mk1QWsCTr0ttrJEiXbNnRw==",
  "dev": true,
  "dependencies": {
    "agent-base": "^7.1.1",
    "debug": "^4.3.4",
    "socks": "^2.8.3"
  },
  "engines": {
    "node": ">= 14"
  }
},
"node_modules/source-map": {
  "version": "0.7.4",
  "resolved": "https://registry.npmjs.org/source-map/-/source-map-0.7.4.tgz",

```

```

"integrity":
"sha512-l3BikUxvPOcn5E74dZiq5BGsTb5yEwhaTSzccU6t4sDOH8NWJCstKO5QT2CvtFoK
6F0saL7p9xHAqHOICPJygA==",
  "dev": true,
  "engines": {
    "node": ">= 8"
  }
},
"node_modules/source-map-js": {
  "version": "1.2.0",
  "resolved": "https://registry.npmjs.org/source-map-js/-/source-map-js-1.2.0.tgz",
  "integrity":
"sha512-itJW8lvSA0TXEphiRoawsCksnlf8SyvmFzlhltqAHluXd88pkCd+cXJVHTDwdCr0Izwp
tSm035IHQktUu1QUMg==",
  "dev": true,
  "engines": {
    "node": ">=0.10.0"
  }
},
"node_modules/source-map-loader": {
  "version": "5.0.0",
  "resolved": "https://registry.npmjs.org/source-map-loader/-/source-map-loader-5.0.0.tgz",
  "integrity":
"sha512-k2Dur7CbSLcAH73sBclkv5xjPV4SzqO1NJ7+XaQl8if3VODDUj3FNchNGpqqJSKb
vUfJuhVdv8K2Eu8/TNI2eA==",
  "dev": true,
  "dependencies": {
    "iconv-lite": "^0.6.3",
    "source-map-js": "^1.0.2"
  },
  "engines": {
    "node": ">= 18.12.0"
  },
  "funding": {
    "type": "opencollective",
    "url": "https://opencollective.com/webpack"
  },
  "peerDependencies": {
    "webpack": "^5.72.1"
  }
},
"node_modules/source-map-loader/node_modules/iconv-lite": {
  "version": "0.6.3",
  "resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.6.3.tgz",
  "integrity":
"sha512-4fck79wshMdzMp2rH06qWrJE4iolqLhCUH+OiuIlgU++RB0+94NIDL81atO7GX55u
UKueo0txHNtvEyl6D7WdMw==",
  "dev": true,

```

```

"dependencies": {
  "safer-buffer": ">= 2.1.2 < 3.0.0"
},
"engines": {
  "node": ">=0.10.0"
}
},
"node_modules/source-map-support": {
  "version": "0.5.21",
  "resolved":
"https://registry.npmjs.org/source-map-support/-/source-map-support-0.5.21.tgz",
  "integrity":
"sha512-uBHU3L3czslyYXKX88fdrGovxdSCoTGDRZ6SYXtSRxLZUzHg5P/66Ht6uoUIHu9E
Zod+inXhKo3qQgwXUT/y1w==",
  "dev": true,
  "dependencies": {
    "buffer-from": "^1.0.0",
    "source-map": "^0.6.0"
  }
},
"node_modules/source-map-support/node_modules/source-map": {
  "version": "0.6.1",
  "resolved": "https://registry.npmjs.org/source-map/-/source-map-0.6.1.tgz",
  "integrity":
"sha512-UjgapumWlbMhkBgzT7Ykc5YXUT46F0iKu8SGXq0bcwP5dz/h0Plj6enJqjz1Zbq2l5
WaqYnrVbwWOWMyF3F47g==",
  "dev": true,
  "engines": {
    "node": ">=0.10.0"
  }
},
"node_modules/spdx-correct": {
  "version": "3.2.0",
  "resolved": "https://registry.npmjs.org/spdx-correct/-/spdx-correct-3.2.0.tgz",
  "integrity":
"sha512-kN9dJbvnySHULluDH32WHRUu3Og7B9sbY7tsFLctQklqnMh3hErYgdMjTYuqmc
XX+IK5T1lnUt3G7zNswmZA==",
  "dev": true,
  "dependencies": {
    "spdx-expression-parse": "^3.0.0",
    "spdx-license-ids": "^3.0.0"
  }
},
"node_modules/spdx-exceptions": {
  "version": "2.5.0",
  "resolved": "https://registry.npmjs.org/spdx-exceptions/-/spdx-exceptions-2.5.0.tgz",

```



```

"integrity":
"sha512-PiU42r+xO4UbUS1buo3LPJkJO7430Xn5SVAhdpzssPHsjbYVflnnFdATgabnLude+
Cqu25p6N+g2lw/PFsa4w==",
  "dev": true
},
"node_modules/spdx-expression-parse": {
  "version": "3.0.1",
  "resolved":
"https://registry.npmjs.org/spdx-expression-parse/-/spdx-expression-parse-3.0.1.tgz",
  "integrity":
"sha512-cbqHunsQWnJNE6KhVSMsMeH5H/L9EpymbzqTQ3uLwNCLZ1Q481oWaofqH7nO
6V07xIXwY6PhQdQ2ledWx/ZK4Q==",
  "dev": true,
  "dependencies": {
    "spdx-exceptions": "^2.1.0",
    "spdx-license-ids": "^3.0.0"
  }
},
"node_modules/spdx-license-ids": {
  "version": "3.0.18",
  "resolved": "https://registry.npmjs.org/spdx-license-ids/-/spdx-license-ids-3.0.18.tgz",
  "integrity":
"sha512-xxRs31BqRYHwiMzudOrpSiHtZ8i/GeionCBDSilhYRj+9glcl8wCZTIXZKu9vZIVqViP
3dcp9qE5G6AllaD+TQ==",
  "dev": true
},
"node_modules/spdy": {
  "version": "4.0.2",
  "resolved": "https://registry.npmjs.org/spdy/-/spdy-4.0.2.tgz",
  "integrity":
"sha512-r46gZQZQV+KI9oltvl1JZZqJKGr+oEkB08A6BzkiR7593/7lbtuncXHd2YoYeTsG4157
ZssMu9KYvUHLcjcDoA==",
  "dev": true,
  "dependencies": {
    "debug": "^4.1.0",
    "handle-thing": "^2.0.0",
    "http-deceiver": "^1.2.7",
    "select-hose": "^2.0.0",
    "spdy-transport": "^3.0.0"
  },
  "engines": {
    "node": ">=6.0.0"
  }
},
"node_modules/spdy-transport": {
  "version": "3.0.0",
  "resolved": "https://registry.npmjs.org/spdy-transport/-/spdy-transport-3.0.0.tgz",

```

```

"integrity":
"sha512-hsLVFE5SjA6TCisWeJXFKniGGOpBgMLmerfO2aCyCU5s7nJ/rpAepqmFifv/GCbS
bueEeAJJnmSQ2rKC/g8Fcw==",
  "dev": true,
  "dependencies": {
    "debug": "^4.1.0",
    "detect-node": "^2.0.4",
    "hpack.js": "^2.1.6",
    "obuf": "^1.1.2",
    "readable-stream": "^3.0.6",
    "wbuf": "^1.7.3"
  }
},
"node_modules/sprintf-js": {
  "version": "1.1.3",
  "resolved": "https://registry.npmjs.org/sprintf-js/-/sprintf-js-1.1.3.tgz",
"integrity":
"sha512-Oo+0REFV59/rz3gfJNKQiBlwfHaSESI1pcGyABQsnnIfWOFt6JNj5gCog2U6MLZ//I
GYD+nA8nl+mTShREReaA==",
  "dev": true
},
"node_modules/ssri": {
  "version": "10.0.6",
  "resolved": "https://registry.npmjs.org/ssri/-/ssri-10.0.6.tgz",
"integrity":
"sha512-MGrFH9Z4NP9lyhqN16sDtBpRRNJ0Y2hNa6D65h736fVSaPCHr4DM4sWUNvVaS
uC+0OBGhwsrydQwmgfg5LncqQ==",
  "dev": true,
  "dependencies": {
    "minipass": "^7.0.3"
  },
  "engines": {
    "node": "^14.17.0 || ^16.13.0 || >=18.0.0"
  }
},
"node_modules/statuses": {
  "version": "1.5.0",
  "resolved": "https://registry.npmjs.org/statuses/-/statuses-1.5.0.tgz",
"integrity":
"sha512-OpZ3zP+jT1PI7I8nemJX4AKmAX070ZkYPVWV/AaKTJI+tXCTGyVdC1a4SL8RUQ
YEwk/f34ZX8UTyKN68FwrqAA==",
  "dev": true,
  "engines": {
    "node": ">= 0.6"
  }
},
"node_modules/streamroller": {
  "version": "3.1.5",

```

```

"resolved": "https://registry.npmjs.org/streamroller/-/streamroller-3.1.5.tgz",
"integrity":
"sha512-KFxaM7XT+irxvdqSP1LGLgNWbYN7ay5owZ3r/8t77p+EtSUAfUgtl7be3xtqtOmGUI
9K9YPO2ca8133RITjvKw==",
  "dev": true,
  "dependencies": {
    "date-format": "^4.0.14",
    "debug": "^4.3.4",
    "fs-extra": "^8.1.0"
  },
  "engines": {
    "node": ">=8.0"
  }
},
"node_modules/string_decoder": {
  "version": "1.3.0",
  "resolved": "https://registry.npmjs.org/string_decoder/-/string_decoder-1.3.0.tgz",
"integrity":
"sha512-hkRX8U1WjJFd8LsDJ2yQ/wWWxaopEsABU1XfkM8A+j0+85JAGppt16cr1Whg6KI
bb4okU6Mql6BOj+uup/wKeA==",
  "dev": true,
  "dependencies": {
    "safe-buffer": "~5.2.0"
  }
},
"node_modules/string-width": {
  "version": "7.2.0",
  "resolved": "https://registry.npmjs.org/string-width/-/string-width-7.2.0.tgz",
"integrity":
"sha512-tsaTIkKW9b4N+AEj+SVA+WhJzV7/zMhcSu78mLKWSk7cXMOSHsBKFWUs0fWw
q8QyK3MgJBQRX6Gbi4kYbdvGkQ==",
  "dev": true,
  "dependencies": {
    "emoji-regex": "^10.3.0",
    "get-east-asian-width": "^1.0.0",
    "strip-ansi": "^7.1.0"
  },
  "engines": {
    "node": ">=18"
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
},
"node_modules/string-width-cjs": {
  "name": "string-width",
  "version": "4.2.3",
  "resolved": "https://registry.npmjs.org/string-width/-/string-width-4.2.3.tgz",

```

```

"integrity":
"sha512-wKyQRQpjJ0sIp62ErSZdGsjMJWsap5oRNihHhu6G7JVO/9jIB6UyevL+tXuOqrng8j/
cxKTWwYUwvSTriiZz/g==",
  "dev": true,
  "dependencies": {
    "emoji-regex": "^8.0.0",
    "is-fullwidth-code-point": "^3.0.0",
    "strip-ansi": "^6.0.1"
  },
  "engines": {
    "node": ">=8"
  }
},
"node_modules/string-width-cjs/node_modules/emoji-regex": {
  "version": "8.0.0",
  "resolved": "https://registry.npmjs.org/emoji-regex/-/emoji-regex-8.0.0.tgz",
  "integrity":
"sha512-MSjYzcvWNOA0ewAHpz0MxpYFvwg6yjj1NG3xteoqz644VCo/RPgnr1/GGt+ic3iJTz
Q8Eu3TdM14SawnVUmGE6A==",
  "dev": true
},
"node_modules/string-width-cjs/node_modules/is-fullwidth-code-point": {
  "version": "3.0.0",
  "resolved":
"https://registry.npmjs.org/is-fullwidth-code-point/-/is-fullwidth-code-point-3.0.0.tgz",
  "integrity":
"sha512-zymm5+u+sCsSWyD9qNaejV3DFvhCKcKdizYaJUuHA83RLjb7nSuGnddCHGv0hk
+KY7BMAlsWeK4Ueg6EV6XQg==",
  "dev": true,
  "engines": {
    "node": ">=8"
  }
},
"node_modules/string-width/node_modules/ansi-regex": {
  "version": "6.0.1",
  "resolved": "https://registry.npmjs.org/ansi-regex/-/ansi-regex-6.0.1.tgz",
  "integrity":
"sha512-n5M855fKb2SsfMliFFoVrABHJC8QtHwVx+mHWP3QcEqBHYienj5dHSgjbxtC0WE
ZXYt4wcD6zrQEIDPhFuZgfA==",
  "dev": true,
  "engines": {
    "node": ">=12"
  }
},
"node_modules/string-width/node_modules/ansi-regex": {
  "url": "https://github.com/chalk/ansi-regex?sponsor=1"
}
},
"node_modules/string-width/node_modules/strip-ansi": {

```

```

"version": "7.1.0",
"resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-7.1.0.tgz",
"integrity":
"sha512-iq6eVVI64nQQTRYq2KtEg2d2uU7LEIhTJwsH4YzIHZshxlgZms/wlc4VoDQTIG/lvVlr
BKG06CrZnp0qv7hkcQ==",
  "dev": true,
  "dependencies": {
    "ansi-regex": "^6.0.1"
  },
  "engines": {
    "node": ">=12"
  },
  "funding": {
    "url": "https://github.com/chalk/strip-ansi?sponsor=1"
  }
},
"node_modules/strip-ansi": {
  "version": "6.0.1",
  "resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-6.0.1.tgz",
"integrity":
"sha512-Y38VPSHcckFrCpFnQ9vuSXMquuv5oXOKpGeT6aGrr3o3Gc9AIVa6JBfUSOCnbx
GGZF+/0ool7KrPuUSztUdU5A==",
  "dev": true,
  "dependencies": {
    "ansi-regex": "^5.0.1"
  },
  "engines": {
    "node": ">=8"
  }
},
"node_modules/strip-ansi-cjs": {
  "name": "strip-ansi",
  "version": "6.0.1",
  "resolved": "https://registry.npmjs.org/strip-ansi/-/strip-ansi-6.0.1.tgz",
"integrity":
"sha512-Y38VPSHcckFrCpFnQ9vuSXMquuv5oXOKpGeT6aGrr3o3Gc9AIVa6JBfUSOCnbx
GGZF+/0ool7KrPuUSztUdU5A==",
  "dev": true,
  "dependencies": {
    "ansi-regex": "^5.0.1"
  },
  "engines": {
    "node": ">=8"
  }
},
"node_modules/strip-final-newline": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/strip-final-newline/-/strip-final-newline-2.0.0.tgz",

```

```

"integrity":
"sha512-BrpvfNAE3dcvq7lI3xVumzjKjZQ5tI1sEUIKr3Uoks0XUI45St3FlatVqef9prk4jRDzhW6
WZg+3bk93y6pLjA==",
  "dev": true,
  "engines": {
    "node": ">=6"
  }
},
"node_modules/strip-outer": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/strip-outer/-/strip-outer-1.0.1.tgz",

```

```

"integrity":
"sha512-k55yxKHwaXnpYGsOzg4VI8+tDrWylxDEpknGjhTiZB8dFRU5rTo9CAzeycivxV3s+zI
TKwrs6WxMxR95n26kwg==",
  "dev": true,
  "dependencies": {
    "escape-string-regexp": "^1.0.2"
  },
  "engines": {
    "node": ">=0.10.0"
  }
},
"node_modules/supports-color": {
  "version": "5.5.0",
  "resolved": "https://registry.npmjs.org/supports-color/-/supports-color-5.5.0.tgz",

```

```

"integrity":
"sha512-QjVjwdXIt408MliAqCX4oUKsgU2EqAGzs2Ppkm4aQYbjm+ZEWECw4SfFNT4uM
NZma0ey4f5lgLrkB0aX0QMow==",
  "dev": true,
  "dependencies": {
    "has-flag": "^3.0.0"
  },
  "engines": {
    "node": ">=4"
  }
},
"node_modules/supports-preserve-symlinks-flag": {
  "version": "1.0.0",

```

```

"resolved":
"https://registry.npmjs.org/supports-preserve-symlinks-flag/-/supports-preserve-symlinks-flag
-1.0.0.tgz",

```

```

"integrity":
"sha512-ot0WnXS9fgdkglcePe6RHNk1WA8+muPa6cSjeR3V8K27q9BB1rTE3R1p7Hv0z1Z
yAc8s6Vvv8DlyWf681MA0w==",
  "dev": true,
  "engines": {
    "node": ">= 0.4"
  },

```

```

    "funding": {
      "url": "https://github.com/sponsors/ljharb"
    }
  },
  "node_modules/symbol-observable": {
    "version": "4.0.0",
    "resolved": "https://registry.npmjs.org/symbol-observable/-/symbol-observable-4.0.0.tgz",
    "integrity":
"sha512-b19dMThMV4HVFynSAM1++gBHAbk2Tc/osgLIBZMKsyqh34jb2e8Os7T6ZW/Bt3pJ
FdBTd2JwAnAAEQV7rSNvcQ==",
    "dev": true,
    "engines": {
      "node": ">=0.10"
    }
  },
  "node_modules/tapable": {
    "version": "2.2.1",
    "resolved": "https://registry.npmjs.org/tapable/-/tapable-2.2.1.tgz",
    "integrity":
"sha512-GNzQvQTOIP6RyTfE2Qxb8ZVINmw0n88vp1szwWRimP02mnTsx3Wtn5qRdqY9w
2XduFNUgvOwhNnQsjwCp+kqaQ==",
    "dev": true,
    "engines": {
      "node": ">=6"
    }
  },
  "node_modules/tar": {
    "version": "6.2.1",
    "resolved": "https://registry.npmjs.org/tar/-/tar-6.2.1.tgz",
    "integrity":
"sha512-DZ4yORTwrbTj/7MZYq2w+/ZFdl6OZ/f9SFHR+71gIVUZhOQPHzVCLpvRnPgYaMpf
Wxxk/4ONva3GQSyNIKRv6A==",
    "dev": true,
    "dependencies": {
      "chownr": "^2.0.0",
      "fs-minipass": "^2.0.0",
      "minipass": "^5.0.0",
      "minizlib": "^2.1.1",
      "mkdirp": "^1.0.3",
      "yallist": "^4.0.0"
    },
    "engines": {
      "node": ">=10"
    }
  },
  "node_modules/tar/node_modules/fs-minipass": {
    "version": "2.1.0",
    "resolved": "https://registry.npmjs.org/fs-minipass/-/fs-minipass-2.1.0.tgz",

```

```

"integrity":
"sha512-V/JgOLFCS+R6Vcq0slCuaeWEdNC3ouDIJMNIscH2VtALiu9mV4LPPrHc5cDI8k5a
w6J8jwgWWpiTo5RYhmlzvg==",
  "dev": true,
  "dependencies": {
    "minipass": "^3.0.0"
  },
  "engines": {
    "node": ">= 8"
  }
},
"node_modules/tar/node_modules/fs-minipass/node_modules/minipass": {
  "version": "3.3.6",
  "resolved": "https://registry.npmjs.org/minipass/-/minipass-3.3.6.tgz",

```

```

"integrity":
"sha512-DxiNidxSEK+tHG6zOlklvNOwm3hvCrbUrdtzY74U6HKTJxvIDfOUL5W5P2Ghd3DT
khhKPYGqeNUih5qcM4YBfw==",
  "dev": true,
  "dependencies": {
    "yallist": "^4.0.0"
  },
  "engines": {
    "node": ">=8"
  }
},
"node_modules/tar/node_modules/minipass": {
  "version": "5.0.0",
  "resolved": "https://registry.npmjs.org/minipass/-/minipass-5.0.0.tgz",

```

```

"integrity":
"sha512-3FnjYuehv9k6ovOEbyOswadCDPX1piCfhV8ncmYtHOjuPwylVWsghtLo7rabjC3Rx
5xD4HDx8Wm1xnMF7S5qFQ==",
  "dev": true,
  "engines": {
    "node": ">=8"
  }
},
"node_modules/tar/node_modules/mkdirp": {
  "version": "1.0.4",
  "resolved": "https://registry.npmjs.org/mkdirp/-/mkdirp-1.0.4.tgz",

```

```

"integrity":
"sha512-vVqVZQyf3WLx2Shd0qJ9xuvqgAyKPLAiqITEtqW0oiUjzo3PePDd6fW9iFz30ef7Ys
p/oiWqbhszeGWW2T6Gzw==",
  "dev": true,
  "bin": {
    "mkdirp": "bin/cmd.js"
  },
  "engines": {
    "node": ">=10"
  }

```



```

    }
  },
  "node_modules/tar/node_modules/yallist": {
    "version": "4.0.0",
    "resolved": "https://registry.npmjs.org/yallist/-/yallist-4.0.0.tgz",
    "integrity":
"sha512-3wdGidZyq5PB084XLES5TpOSRA3wjXAlIWMhum2kRcv/41Sn2emQ0dycQW4uZ
XLejwKvg6EsvbdIVL+FYEct7A==",
    "dev": true
  },
  "node_modules/terser": {
    "version": "5.29.2",
    "resolved": "https://registry.npmjs.org/terser/-/terser-5.29.2.tgz",
    "integrity":
"sha512-ZiGkhUBIM+7LwkNjXYJq8svgkd+QK3UUr0wJqY4MieaezBSAIPgbSPZylx0idM6X
WK5CMzSWa8MJlzmRcB8Caw==",
    "dev": true,
    "dependencies": {
      "@jridgewell/source-map": "^0.3.3",
      "acorn": "^8.8.2",
      "commander": "^2.20.0",
      "source-map-support": "~0.5.20"
    },
    "bin": {
      "terser": "bin/terser"
    },
    "engines": {
      "node": ">=10"
    }
  },
  "node_modules/terser-webpack-plugin": {
    "version": "5.3.10",
    "resolved":
"https://registry.npmjs.org/terser-webpack-plugin/-/terser-webpack-plugin-5.3.10.tgz",
    "integrity":
"sha512-BKFPWIPDndPs+NGGCr1U59t0XScL5317Y0UReNrHaw9/FwhPENlq6bfgs+4yPfy
P51vqC1bQ4rp1EfXW5ZSH9w==",
    "dev": true,
    "dependencies": {
      "@jridgewell/trace-mapping": "^0.3.20",
      "jest-worker": "^27.4.5",
      "schema-utils": "^3.1.1",
      "serialize-javascript": "^6.0.1",
      "terser": "^5.26.0"
    },
    "engines": {
      "node": ">= 10.13.0"
    }
  },

```

```

"funding": {
  "type": "opencollective",
  "url": "https://opencollective.com/webpack"
},
"peerDependencies": {
  "webpack": "^5.1.0"
},
"peerDependenciesMeta": {
  "@swc/core": {
    "optional": true
  },
  "esbuild": {
    "optional": true
  },
  "uglify-js": {
    "optional": true
  }
}
},
"node_modules/terser-webpack-plugin/node_modules/ajv": {
  "version": "6.12.6",
  "resolved": "https://registry.npmjs.org/ajv/-/ajv-6.12.6.tgz",
  "integrity":
"sha512-j3fVLgvTo527anyYyJOGTYJbG+vnnQYvE0m5mmkc1TK+nxAppkCLMIL0aZ4dbIVC
NoGShhm+kzE4ZUykBoMg4g==",
  "dev": true,
  "dependencies": {
    "fast-deep-equal": "^3.1.1",
    "fast-json-stable-stringify": "^2.0.0",
    "json-schema-traverse": "^0.4.1",
    "uri-js": "^4.2.2"
  },
  "funding": {
    "type": "github",
    "url": "https://github.com/sponsors/epoberezkin"
  }
},
"node_modules/terser-webpack-plugin/node_modules/ajv-keywords": {
  "version": "3.5.2",
  "resolved": "https://registry.npmjs.org/ajv-keywords/-/ajv-keywords-3.5.2.tgz",
  "integrity":
"sha512-5p6WTN0DdTGvQk6VjcEju19lgaHudalcfabD7yhDGeA6bcQnmL+CpveLJq/3hvfwd
1aof6L386Ougkx6RfyMIQ==",
  "dev": true,
  "peerDependencies": {
    "ajv": "^6.9.1"
  }
},

```

```

"node_modules/terser-webpack-plugin/node_modules/json-schema-traverse": {
  "version": "0.4.1",
  "resolved":
"sha512-xbbCH5dCYU5T8LcEhhuh7HJ88HXuW3qsI3Y0zOZFKfZEHcpWiHU/Jxzk629Brsab
/mMiHQti9wMP+845RPe3Vg==",
  "integrity":
"sha512-pN/yOAvCC+5rQ5nERGuwrjLIYvLTbCibnZ1I7B1LaiAz9BRBIE9GMgE/eqV30P7aJQ
Uf7Ddimy/RsbYO/GrVGg==",
  "dev": true
},
"node_modules/terser-webpack-plugin/node_modules/schema-utils": {
  "version": "3.3.0",
  "resolved": "https://registry.npmjs.org/schema-utils/-/schema-utils-3.3.0.tgz",
  "integrity":
"sha512-pN/yOAvCC+5rQ5nERGuwrjLIYvLTbCibnZ1I7B1LaiAz9BRBIE9GMgE/eqV30P7aJQ
Uf7Ddimy/RsbYO/GrVGg==",
  "dev": true,
  "dependencies": {
    "@types/json-schema": "^7.0.8",
    "ajv": "^6.12.5",
    "ajv-keywords": "^3.5.2"
  },
  "engines": {
    "node": ">= 10.13.0"
  },
  "funding": {
    "type": "opencollective",
    "url": "https://opencollective.com/webpack"
  }
},
"node_modules/thingies": {
  "version": "1.21.0",
  "resolved": "https://registry.npmjs.org/thingies/-/thingies-1.21.0.tgz",
  "integrity":
"sha512-hsqJsFMsv+aD4s3CWKk85ep/3I9XzYV/IXaSouJMYloDIgyi11cBhsqYe9/geRfB0Y
likBQg6raRaM+nIMP9g==",
  "dev": true,
  "engines": {
    "node": ">=10.18"
  },
  "peerDependencies": {
    "tslib": "^2"
  }
},
"node_modules/thunky": {
  "version": "1.1.0",
  "resolved": "https://registry.npmjs.org/thunky/-/thunky-1.1.0.tgz",

```

```

"integrity":
"sha512-eHY7nBftgThBqOyHGVN+l8gF0BucP09fMo0oO/Lb0w1OF80dJv+IDVpXG60WMQ
vkcxAkNybKsrEIE3ZtKGmPrA==",
  "dev": true
},
"node_modules/tmp": {
  "version": "0.0.33",
  "resolved": "https://registry.npmjs.org/tmp/-/tmp-0.0.33.tgz",
"integrity":
"sha512-jRCJlojKnZ3addtTOjdlqoRuPEKBvNXcGYqzO6zWZX8KfKEpnGY5jfggJQ3EjKuu8
D4bJRr0y+cYJFmYblmXGw==",
  "dev": true,
  "dependencies": {
    "os-tmpdir": "~1.0.2"
  },
  "engines": {
    "node": ">=0.6.0"
  }
},
"node_modules/to-fast-properties": {
  "version": "2.0.0",
  "resolved": "https://registry.npmjs.org/to-fast-properties/-/to-fast-properties-2.0.0.tgz",
"integrity":
"sha512-/OaKK0xYrs3DmxRYqL/yDc+FxFUVYhDIXMhRmv3z915w2HF1tnN1omB354j8VU
GO/hbRzYD6Y3sA7v7GS/ceog==",
  "dev": true,
  "engines": {
    "node": ">=4"
  }
},
"node_modules/to-regex-range": {
  "version": "5.0.1",
  "resolved": "https://registry.npmjs.org/to-regex-range/-/to-regex-range-5.0.1.tgz",
"integrity":
"sha512-65Z7Nyp7xYfHhENsHBM3rHypBN8yQOnUwbeH4l/vG1l/K0M0g/ZwKJ8KfZCPQ61m/Yb4Bdz6jgf0MDtaoG8YQ==",
  "dev": true,
  "dependencies": {
    "is-number": "^7.0.0"
  },
  "engines": {
    "node": ">=8.0"
  }
},
"node_modules/toidentifier": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/toidentifier/-/toidentifier-1.0.1.tgz",

```

```

"integrity":
"sha512-o5sSPKEkg/DIQNmH43V0/uerLrpzVedkUh8tGNvaeXpfpuwjKenlSox/2O/BTIZUtEe
+JG7s5YhEz608PIAHRA==",
  "engines": {
    "node": ">=0.6"
  }
},
"node_modules/tree-dump": {
  "version": "1.0.2",
  "resolved": "https://registry.npmjs.org/tree-dump/-/tree-dump-1.0.2.tgz",

```

```

"integrity":
"sha512-dpev9ABuLWdEubk+clal9cHwRNNDjkBBLXTwI4UCUFdQ5xXKqNXoK4FEciw/vxf+
NQ7Cb7sGUyeUtORvHIdRXQ==",
  "dev": true,
  "engines": {
    "node": ">=10.0"
  },
  "funding": {
    "type": "github",
    "url": "https://github.com/sponsors/streamich"
  },
  "peerDependencies": {
    "tslib": "2"
  }
},
"node_modules/tree-kill": {
  "version": "1.2.2",
  "resolved": "https://registry.npmjs.org/tree-kill/-/tree-kill-1.2.2.tgz",

```

```

"integrity":
"sha512-L0Orpi8qGpRG//Nd+H90vFB+3iHnue1zSSGmNOOCh1GLJ7rUKVwV2HvijphGQS
2UmhUZewS9VgvxYldgr+fG1A==",
  "dev": true,
  "bin": {
    "tree-kill": "cli.js"
  }
},
"node_modules/trim-repeated": {
  "version": "1.0.0",
  "resolved": "https://registry.npmjs.org/trim-repeated/-/trim-repeated-1.0.0.tgz",

```

```

"integrity":
"sha512-pkonvIKk8/ZuR0D5tLW8ljt5l8kmp2XKymhепUeOdCEfKpZaktSArkLHZt76OB1ZvO
9bssUsDty4SWhLvZpLg==",
  "dev": true,
  "dependencies": {
    "escape-string-regexp": "^1.0.2"
  },
  "engines": {
    "node": ">=0.10.0"

```

```

    }
  },
  "node_modules/tslib": {
    "version": "2.6.3",
    "resolved": "https://registry.npmjs.org/tslib/-/tslib-2.6.3.tgz",
    "integrity":
"sha512-xNvxJEOUiWPGhUuUdQgAJPkOOJfGnlyKySOc09XkkKsgdUV/3E2zvwZYdejimRg
PCGcym1juLH3226yA7sEFJKQ=="
  },
  "node_modules/tuf-js": {
    "version": "2.2.1",
    "resolved": "https://registry.npmjs.org/tuf-js/-/tuf-js-2.2.1.tgz",
    "integrity":
"sha512-GwlJau9XaA8nLVbUXsN3lIFi7WmQ48gBUrI3FTkkL/XLu/POhBzfmX9hd33FNMx1
qAsfl6ozO1iMmW9NC8YniA==",
    "dev": true,
    "dependencies": {
      "@tufjs/models": "2.0.1",
      "debug": "^4.3.4",
      "make-fetch-happen": "^13.0.1"
    },
    "engines": {
      "node": "^16.14.0 || >=18.0.0"
    }
  },
  "node_modules/type-fest": {
    "version": "0.21.3",
    "resolved": "https://registry.npmjs.org/type-fest/-/type-fest-0.21.3.tgz",
    "integrity":
"sha512-t0rzBq87m3fVcduHDUFhKmyyX+9eo6WQjZvf51Ea/M0Q7+T374Jp1aUiyUI0GKxp8
M/OETVHSDvmkyPgvX+X2w==",
    "dev": true,
    "engines": {
      "node": ">=10"
    },
    "funding": {
      "url": "https://github.com/sponsors/sindresorhus"
    }
  },
  "node_modules/type-is": {
    "version": "1.6.18",
    "resolved": "https://registry.npmjs.org/type-is/-/type-is-1.6.18.tgz",
    "integrity":
"sha512-TkRKR9sUTxeh8MdfuCSP7VizJyzRNMjj2J2do2Jr3Kym598JVdEksuzPQCnIFPW4
ky9Q+iA+ma9BGm06XQBy8g==",
    "dependencies": {
      "media-typer": "0.3.0",
      "mime-types": "~2.1.24"
    }
  }

```

```

    },
    "engines": {
      "node": ">= 0.6"
    }
  },
  "node_modules/typed-assert": {
    "version": "1.0.9",
    "resolved": "https://registry.npmjs.org/typed-assert/-/typed-assert-1.0.9.tgz",
    "integrity":
"sha512-KNNZtayBCtmnNmbo5mG47p1XsCyrx6iVqomjcZnec/1Y5GGARaxPs6r49RnSPeU
P3YjNYiU9sQHAtY4BBvnZwg==",
    "dev": true
  },
  "node_modules/typescript": {
    "version": "5.5.4",
    "resolved": "https://registry.npmjs.org/typescript/-/typescript-5.5.4.tgz",
    "integrity":
"sha512-Mtq29sKDAEYP7aljRgtPOpTvOfbwRWIS6dPRzwdE+C0R4brX/GUyhHSecbHMFL
NBLcJlPt9nl9yG5TZ1weH+Q==",
    "dev": true,
    "bin": {
      "tsc": "bin/tsc",
      "tsserver": "bin/tsserver"
    },
    "engines": {
      "node": ">=14.17"
    }
  },
  "node_modules/ua-parser-js": {
    "version": "0.7.38",
    "resolved": "https://registry.npmjs.org/ua-parser-js/-/ua-parser-js-0.7.38.tgz",
    "integrity":
"sha512-fYmly7fKTSFAhG3fuPlubeGaMoAd6r0rSnfEsO5nEY55i26KSLt9EH7PLQiiqPUhNq
YIJvSkTy1oArlcXAbPbA==",
    "dev": true,
    "funding": [
      {
        "type": "opencollective",
        "url": "https://opencollective.com/ua-parser-js"
      },
      {
        "type": "paypal",
        "url": "https://paypal.me/faisalman"
      },
      {
        "type": "github",
        "url": "https://github.com/sponsors/faisalman"
      }
    ]
  }

```

```

    ],
    "engines": {
      "node": "*"
    }
  },
  "node_modules/undici": {
    "version": "6.19.2",
    "resolved": "https://registry.npmjs.org/undici/-/undici-6.19.2.tgz",
    "integrity":
"sha512-JfjKqlaur3Q6biAtHJ564e3bWa8VvT+7cSiOJHFbX4Erv6CLGDpg8z+Fmg/1OI/47R
A+GI2QZaF48SSaLvyBA==",
    "dev": true,
    "engines": {
      "node": ">=18.17"
    }
  },
  "node_modules/undici-types": {
    "version": "6.13.0",
    "resolved": "https://registry.npmjs.org/undici-types/-/undici-types-6.13.0.tgz",
    "integrity":
"sha512-xtFJHudx8S2DSoujjMd1WeWvn7KKWFRESZTMeL1RptAYERu29D6jphMjjY+vn96j
vN3kVPDNxU/E13VTaXj6jg==",
    "dev": true
  },
  "node_modules/unicode-canonical-property-names-ecmascript": {
    "version": "2.0.0",
    "resolved":
"https://registry.npmjs.org/unicode-canonical-property-names-ecmascript/-/unicode-canonical
-property-names-ecmascript-2.0.0.tgz",
    "integrity":
"sha512-yY5PpDfVIU5+y/BSCxAJRBIS1Zc2dDG3Ujq+sR0U+JjUevW2JhocOF+soROYDSa
AezOzOKuyyixhD6mBknSmQ==",
    "dev": true,
    "engines": {
      "node": ">=4"
    }
  },
  "node_modules/unicode-match-property-ecmascript": {
    "version": "2.0.0",
    "resolved":
"https://registry.npmjs.org/unicode-match-property-ecmascript/-/unicode-match-property-ecm
ascript-2.0.0.tgz",
    "integrity":
"sha512-5kaZCrbp5mmbz5ulBkDkbY0SsPOjKqVS35VpL9ulIMPfSI0J0Xsm+9Evphv9ColZFW
re7aJoa94AY6seMKGvN5Q==",
    "dev": true,
    "dependencies": {
      "unicode-canonical-property-names-ecmascript": "^2.0.0",

```



```

    "unicode-property-aliases-ecmascript": "^2.0.0"
  },
  "engines": {
    "node": ">=4"
  }
},
"node_modules/unicode-match-property-value-ecmascript": {
  "version": "2.1.0",
  "resolved":
"https://registry.npmjs.org/unicode-match-property-value-ecmascript/-/unicode-match-propert
y-value-ecmascript-2.1.0.tgz",
  "integrity":
"sha512-qxkjQt6qjg/mYscYMC0XKRn3Rh0wFPIfxB0xkt9CfyTvpX1Ra0+rAmdX2QyAobptSE
vuy4RtpPRui6XkV+8wjA==",
  "dev": true,
  "engines": {
    "node": ">=4"
  }
},
"node_modules/unicode-property-aliases-ecmascript": {
  "version": "2.1.0",
  "resolved":
"https://registry.npmjs.org/unicode-property-aliases-ecmascript/-/unicode-property-aliases-ec
mascript-2.1.0.tgz",
  "integrity":
"sha512-6t3foTQI9qne+OZoVQB/8x8rk2k1eVy1gRXhV3oFQ5T6R1dqQ1xtin3XqSlx3+ATBkl
iTaR/hHyJBm+LVPNM8w==",
  "dev": true,
  "engines": {
    "node": ">=4"
  }
},
"node_modules/unicorn-magic": {
  "version": "0.1.0",
  "resolved": "https://registry.npmjs.org/unicorn-magic/-/unicorn-magic-0.1.0.tgz",
  "integrity":
"sha512-IRfVq8fE8gz6QMBuDM6a+LO3IAzTi05H6gCVaUpir2E1Rwpo4ZUog45KpNXXK/M
n3Yb9UDuHumeFTo9iV/D9FQ==",
  "dev": true,
  "engines": {
    "node": ">=18"
  },
  "funding": {
    "url": "https://github.com/sponsors/sindresorhus"
  }
},
"node_modules/unique-filename": {
  "version": "3.0.0",

```

```

    "resolved": "https://registry.npmjs.org/unique-filename/-/unique-filename-3.0.0.tgz",
    "integrity":
"sha512-afXhuC55wkAmZ0P18QsVE6kp8JaxrEokN2HGloIVv2ijHQd419H0+6EigAFclzXeMI
kclKNBpB3L/DXB3cTS/g==",
    "dev": true,
    "dependencies": {
      "unique-slug": "^4.0.0"
    },
    "engines": {
      "node": "^14.17.0 || ^16.13.0 || >=18.0.0"
    }
  },
  "node_modules/unique-slug": {
    "version": "4.0.0",
    "resolved": "https://registry.npmjs.org/unique-slug/-/unique-slug-4.0.0.tgz",
    "integrity":
"sha512-WrcA6AyEfqDX5bWige/4NQfPZMtASNVxdmWR76WESYQVAACSgWcR6e9i0mof
qqBxYFtL4oAxPIptY73/0YE1DQ==",
    "dev": true,
    "dependencies": {
      "imurmurhash": "^0.1.4"
    },
    "engines": {
      "node": "^14.17.0 || ^16.13.0 || >=18.0.0"
    }
  },
  "node_modules/universalify": {
    "version": "0.1.2",
    "resolved": "https://registry.npmjs.org/universalify/-/universalify-0.1.2.tgz",
    "integrity":
"sha512-rBJeI5CXAlmy1pV+617WB9J63U6XcazHHF2f2dbJix4XzpUF0RS3Zbj0FGIOCAva5
P/d/GBOYaACQ1w+0azUkg==",
    "dev": true,
    "engines": {
      "node": ">= 4.0.0"
    }
  },
  "node_modules/unpipe": {
    "version": "1.0.0",
    "resolved": "https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tgz",
    "integrity":
"sha512-rBJeI5CXAlmy1pV+617WB9J63U6XcazHHF2f2dbJix4XzpUF0RS3Zbj0FGIOCAva5
P/d/GBOYaACQ1w+0azUkg==",
    "dev": true,
    "engines": {
      "node": ">= 0.8"
    }
  },
  "node_modules/update-browserslist-db": {

```

```

"version": "1.1.0",
"resolved":
"https://registry.npmjs.org/update-browserslist-db/-/update-browserslist-db-1.1.0.tgz",
"integrity":
"sha512-EdRAaAyk2cUE1wOf2DkEhvxqOQvFOoRJFNS6NeyJ01Gp2beMRpBAINjM2iDXE
3KCuKhwnvHIQCJm6ThL2Z+HzQ==",
  "dev": true,
  "funding": [
    {
      "type": "opencollective",
      "url": "https://opencollective.com/browserslist"
    },
    {
      "type": "tidelift",
      "url": "https://tidelift.com/funding/github/npm/browserslist"
    },
    {
      "type": "github",
      "url": "https://github.com/sponsors/ai"
    }
  ],
  "dependencies": {
    "escalade": "^3.1.2",
    "picocolors": "^1.0.1"
  },
  "bin": {
    "update-browserslist-db": "cli.js"
  },
  "peerDependencies": {
    "browserslist": ">= 4.21.0"
  },
  "node_modules/uri-js": {
    "version": "4.4.1",
    "resolved": "https://registry.npmjs.org/uri-js/-/uri-js-4.4.1.tgz",
    "integrity":
"sha512-7rXuoRfMsSuBzksa3WjKybfM93zpuP6dWbQAgH3jY6z1qU/QsZn8Kbd3Eh2JhuJYKZ1oRl4gH+lcym0zZg=="
  },
  "node_modules/uri-js/node_modules/punycode": {
    "version": "2.1.0",
    "resolved": "https://registry.npmjs.org/punycode/-/punycode-2.1.1.tgz",

```

```

"integrity":
"sha512-vYt7UD1U9Wg6138shLtLOvdAu+8DsC/iIFtEVHcH+wydcSpNE20AfSOduf6MkRFa
hL5FY7X1oU7nKVZFtfq8Fg==",
  "dev": true,
  "engines": {
    "node": ">=6"
  }
},
"node_modules/util": {
  "version": "0.10.4",
  "resolved": "https://registry.npmjs.org/util/-/util-0.10.4.tgz",
"integrity":
"sha512-0Pm9hTQ3se5l1XihRic3FDIku70C+iHUdT/W926rSgHV5QgXsYbKZN8MSC3tJtSk
huROzvsQjAaFENRr+19A==",
  "dependencies": {
    "inherits": "2.0.3"
  }
},
"node_modules/util-deprecate": {
  "version": "1.0.2",
  "resolved": "https://registry.npmjs.org/util-deprecate/-/util-deprecate-1.0.2.tgz",
"integrity":
"sha512-EPD5q1uXyFxFjCrLnCc1nHnq3gOa6DZBocAII2TaSCA7VCJ1UJDMagCzIkXNsU
YfD1daK//LTEQ8xilbrHtcw==",
  "dev": true
},
"node_modules/util/node_modules/inherits": {
  "version": "2.0.3",
  "resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.3.tgz",
"integrity":
"sha512-x00IRNXNy63jwGkJmzPigoySHbaqpNuzKbBOMzK+g2OdZpQ9w+sxCN+VSB3ja7I
Age2OP2qpfXTjeNcyjmW1uw=="
},
"node_modules/utis-merge": {
  "version": "1.0.1",
  "resolved": "https://registry.npmjs.org/utis-merge/-/utis-merge-1.0.1.tgz",
"integrity":
"sha512-pMZTvIkT1d+TFGvDOqodOclx0QWkkgi6Tdoa8gC8ffGAAqz9pzPTZWAYbbsHHoE
D/ztMtkv/VoYTYyShUn81hA=="
},
  "engines": {
    "node": ">= 0.4.0"
  }
},
"node_modules/uuid": {
  "version": "8.3.2",
  "resolved": "https://registry.npmjs.org/uuid/-/uuid-8.3.2.tgz",

```

```

"integrity":
"sha512-+NYs2QeMWy+GWFOEm9xnn6HCDp0l7QBD7ml8zLUmJ+93Q5NF0NocErnwTk
XVFNiX3/fpC6afS8Dhb/gz7R7eg==",
  "dev": true,
  "bin": {
    "uuid": "dist/bin/uuid"
  }
},
"node_modules/validate-npm-package-license": {
  "version": "3.0.4",
  "resolved":
"https://registry.npmjs.org/validate-npm-package-license/-/validate-npm-package-license-3.0.
4.tgz",
  "integrity":
"sha512-DpKm2Ui/xN7/HQKcTpZxoRWBhZ9Z0kqtygG8XCgNQ8ZIDnxuQmWhj566j8fN4Cu
3/JmbhsDo7fcAJq4s9h27Ew==",
  "dev": true,
  "dependencies": {
    "spdx-correct": "^3.0.0",
    "spdx-expression-parse": "^3.0.0"
  }
},
"node_modules/validate-npm-package-name": {
  "version": "5.0.1",
  "resolved":
"https://registry.npmjs.org/validate-npm-package-name/-/validate-npm-package-name-5.0.1.t
gz",
  "integrity":
"sha512-OljlLrQ9SQdOUqTaQxqL5dEfZWrxEXyyWsozYIAWFawPVNuD83igl7uJD2RTkNMb
nilYgt8l81eCJGIQF7avLQ==",
  "dev": true,
  "engines": {
    "node": "^14.17.0 || ^16.13.0 || >=18.0.0"
  }
},
"node_modules/vary": {
  "version": "1.1.2",
  "resolved": "https://registry.npmjs.org/vary/-/vary-1.1.2.tgz",
  "integrity":
"sha512-BNGbWLfd0eUPabhkXUVm0j8uuvREyTh5ovRa/dyow/BqAbZJyC+5fU+lzQOzmAK
zYqYRAISoRhdQr3eIZ/PXqg==",
  "engines": {
    "node": ">= 0.8"
  }
},
"node_modules/vite": {
  "version": "5.3.2",
  "resolved": "https://registry.npmjs.org/vite/-/vite-5.3.2.tgz",

```

```

"integrity":
"sha512-6IA7OBHBIXUxiJxbO5aAY2fsHHzDr1q7DvXYnyZycRs2Dz+dXBWuhpWHvmljTRT
pQC2uvGmUFFkSHF2vGo90MA==",
  "dev": true,
  "dependencies": {
    "esbuild": "^0.21.3",
    "postcss": "^8.4.38",
    "rollup": "^4.13.0"
  },
  "bin": {
    "vite": "bin/vite.js"
  },
  "engines": {
    "node": "^18.0.0 || >=20.0.0"
  },
  "funding": {
    "url": "https://github.com/vitejs/vite?sponsor=1"
  },
  "optionalDependencies": {
    "fsevents": "~2.3.3"
  },
  "peerDependencies": {
    "@types/node": "^18.0.0 || >=20.0.0",
    "less": "*",
    "lightningcss": "^1.21.0",
    "sass": "*",
    "stylus": "*",
    "sugarss": "*",
    "terser": "^5.4.0"
  },
  "peerDependenciesMeta": {
    "@types/node": {
      "optional": true
    },
    "less": {
      "optional": true
    },
    "lightningcss": {
      "optional": true
    },
    "sass": {
      "optional": true
    },
    "stylus": {
      "optional": true
    },
    "sugarss": {
      "optional": true
    }
  }

```

```

    },
    "terser": {
      "optional": true
    }
  },
  "node_modules/void-elements": {
    "version": "2.0.1",
    "resolved": "https://registry.npmjs.org/void-elements/-/void-elements-2.0.1.tgz",
    "integrity":
"sha512-qZKX4RnBzH2ugr8Lxa7x+0V6XD9Sb/ouARTiasEQCHB1EVU4NXtmHsDDrx1dO4n
e5fc3J6EW05BP1DI0z0iung==",
    "dev": true,
    "engines": {
      "node": ">=0.10.0"
    }
  },
  "node_modules/watchpack": {
    "version": "2.4.1",
    "resolved": "https://registry.npmjs.org/watchpack/-/watchpack-2.4.1.tgz",
    "integrity":
"sha512-8wrBCMtVhqcXP2Sup1ctSkga6uc2Bx0IlvKyT7yTFier5AXHooSI+QyQQAtTb7+E0I
UCCKyTFmXqdqum2XWGg==",
    "dev": true,
    "dependencies": {
      "glob-to-regexp": "^0.4.1",
      "graceful-fs": "^4.1.2"
    },
    "engines": {
      "node": ">=10.13.0"
    }
  },
  "node_modules/wbuf": {
    "version": "1.7.3",
    "resolved": "https://registry.npmjs.org/wbuf/-/wbuf-1.7.3.tgz",
    "integrity":
"sha512-O84QOnr0icsbFGLS0O3bI5FswxzRr8/gHwWkDIQFskhSPryQXvrTMxjxGP4+iWYo
auLoBvfDpkrOauZ+0iZpDA==",
    "dev": true,
    "dependencies": {
      "minimalistic-assert": "^1.0.0"
    }
  },
  "node_modules/wcwidth": {
    "version": "1.0.1",
    "resolved": "https://registry.npmjs.org/wcwidth/-/wcwidth-1.0.1.tgz",

```

```

"integrity":
"sha512-XHPEwS0q6TaxcvG85+8EYkbiCux2XtWG2mkc47Ng2A77BQu9+DqI0JldST4HgP
kuea7dvKSj5VgX3P1d4rW8Tg==",
  "dev": true,
  "dependencies": {
    "defaults": "^1.0.3"
  }
},
"node_modules/weak-lru-cache": {
  "version": "1.2.2",
  "resolved": "https://registry.npmjs.org/weak-lru-cache/-/weak-lru-cache-1.2.2.tgz",
  "integrity":
"sha512-DEAoo25RfSYMtGc9vPJzZcZullwIqRDSI9LOy+fkCJPi6hykCnfKaXTuPBduXAUC
qHXyOgFtHNp/kB2FjYHbw==",
  "dev": true
},
"node_modules/webpack": {
  "version": "5.92.1",
  "resolved": "https://registry.npmjs.org/webpack/-/webpack-5.92.1.tgz",
  "integrity":
"sha512-JECQ7lwJb+7fgUFBlrJzbyu3GEuNBcdqr1LD7lbSzwkSmlevTm8PF+wej3Oxuz/JFB
UZ6O1o43zsPkwm1C4TmA==",
  "dev": true,
  "dependencies": {
    "@types/eslint-scope": "^3.7.3",
    "@types/estree": "^1.0.5",
    "@webassemblyjs/ast": "^1.12.1",
    "@webassemblyjs/wasm-edit": "^1.12.1",
    "@webassemblyjs/wasm-parser": "^1.12.1",
    "acorn": "^8.7.1",
    "acorn-import-attributes": "^1.9.5",
    "browserslist": "^4.21.10",
    "chrome-trace-event": "^1.0.2",
    "enhanced-resolve": "^5.17.0",
    "es-module-lexer": "^1.2.1",
    "eslint-scope": "5.1.1",
    "events": "^3.2.0",
    "glob-to-regexp": "^0.4.1",
    "graceful-fs": "^4.2.11",
    "json-parse-even-better-errors": "^2.3.1",
    "loader-runner": "^4.2.0",
    "mime-types": "^2.1.27",
    "neo-async": "^2.6.2",
    "schema-utils": "^3.2.0",
    "tapable": "^2.1.1",
    "terser-webpack-plugin": "^5.3.10",
    "watchpack": "^2.4.1",
    "webpack-sources": "^3.2.3"
  }
}

```



```

    },
    "bin": {
      "webpack": "bin/webpack.js"
    },
    "engines": {
      "node": ">=10.13.0"
    },
    "funding": {
      "type": "opencollective",
      "url": "https://opencollective.com/webpack"
    },
    "peerDependenciesMeta": {
      "webpack-cli": {
        "optional": true
      }
    }
  },
  "node_modules/webpack-dev-middleware": {
    "version": "7.2.1",
    "resolved":
    "https://registry.npmjs.org/webpack-dev-middleware/-/webpack-dev-middleware-7.2.1.tgz",
    "integrity":
    "sha512-hRLz+jPQXo999Nx9fXVdKlg/aeHsw1ajA9skAneGmT03xwmyuhvF93p6HUKKbWh
    XdcERtGTzUCtlQr+2lQegrA==",
    "dev": true,
    "dependencies": {
      "colorette": "^2.0.10",
      "memfs": "^4.6.0",
      "mime-types": "^2.1.31",
      "on-finished": "^2.4.1",
      "range-parser": "^1.2.1",
      "schema-utils": "^4.0.0"
    },
    "engines": {
      "node": ">= 18.12.0"
    },
    "funding": {
      "type": "opencollective",
      "url": "https://opencollective.com/webpack"
    },
    "peerDependencies": {
      "webpack": "^5.0.0"
    },
    "peerDependenciesMeta": {
      "webpack": {
        "optional": true
      }
    }
  }
}

```

```

},
"node_modules/webpack-dev-server": {
  "version": "5.0.4",
  "resolved":
    "https://registry.npmjs.org/webpack-dev-server/-/webpack-dev-server-5.0.4.tgz",
  "integrity":
    "sha512-dljXhUgx3HqKP2d8J/fUMvvhxGhzjeNVarDLcbO/EWMSgRizDkxHQDZQaLFL5VJY9
    tRBj2Gz+rvCEYYvvhbqPHNA==",
  "dev": true,
  "dependencies": {
    "@types/bonjour": "^3.5.13",
    "@types/connect-history-api-fallback": "^1.5.4",
    "@types/express": "^4.17.21",
    "@types/serve-index": "^1.9.4",
    "@types/serve-static": "^1.15.5",
    "@types/sockjs": "^0.3.36",
    "@types/ws": "^8.5.10",
    "ansi-html-community": "^0.0.8",
    "bonjour-service": "^1.2.1",
    "chokidar": "^3.6.0",
    "colorette": "^2.0.10",
    "compression": "^1.7.4",
    "connect-history-api-fallback": "^2.0.0",
    "default-gateway": "^6.0.3",
    "express": "^4.17.3",
    "graceful-fs": "^4.2.6",
    "html-entities": "^2.4.0",
    "http-proxy-middleware": "^2.0.3",
    "ipaddr.js": "^2.1.0",
    "launch-editor": "^2.6.1",
    "open": "^10.0.3",
    "p-retry": "^6.2.0",
    "rimraf": "^5.0.5",
    "schema-utils": "^4.2.0",
    "selfsigned": "^2.4.1",
    "serve-index": "^1.9.1",
    "sockjs": "^0.3.24",
    "spdy": "^4.0.2",
    "webpack-dev-middleware": "^7.1.0",
    "ws": "^8.16.0"
  },
  "bin": {
    "webpack-dev-server": "bin/webpack-dev-server.js"
  },
  "engines": {
    "node": ">= 18.12.0"
  },
  "funding": {

```

```

    "type": "opencollective",
    "url": "https://opencollective.com/webpack"
  },
  "peerDependencies": {
    "webpack": "^5.0.0"
  },
  "peerDependenciesMeta": {
    "webpack": {
      "optional": true
    },
    "webpack-cli": {
      "optional": true
    }
  }
},
"node_modules/webpack-dev-server/node_modules/brace-expansion": {
  "version": "2.0.1",
  "resolved": "https://registry.npmjs.org/brace-expansion/-/brace-expansion-2.0.1.tgz",
  "integrity":
"sha512-XnAlvQ8eM+kC6aULx6wuQiwVsnzsi9d3WxzV3FpWTGA19F621kwdbAcFKXgKU
HZWsy+mY6iL1sHTxWEFCytDA==",
  "dev": true,
  "dependencies": {
    "balanced-match": "^1.0.0"
  }
},
"node_modules/webpack-dev-server/node_modules/glob": {
  "version": "10.4.5",
  "resolved": "https://registry.npmjs.org/glob/-/glob-10.4.5.tgz",
  "integrity":
"sha512-7Bv8RF0k6xjo7d4A/PxYlBUCfb6c+Vpd2/mB2yRDlew7Jb5hEXiCD9ibfO7wpk8i4se
vK6DFny9h7EYbM3/sHg==",
  "dev": true,
  "dependencies": {
    "foreground-child": "^3.1.0",
    "jackspeak": "^3.1.2",
    "minimatch": "^9.0.4",
    "minipass": "^7.1.2",
    "package-json-from-dist": "^1.0.0",
    "path-scurry": "^1.11.1"
  },
  "bin": {
    "glob": "dist/esm/bin.mjs"
  },
  "funding": {
    "url": "https://github.com/sponsors/isaacs"
  }
},

```

```

"node_modules/webpack-dev-server/node_modules/http-proxy-middleware": {
  "version": "2.0.6",
  "resolved":
    "https://registry.npmjs.org/http-proxy-middleware/-/http-proxy-middleware-2.0.6.tgz",
  "integrity":
    "sha512-ya/UeJ6HVBYxrgYotAZo1KvPWlgB48kUJLDePFeneHsVujFaW5WNj2NgWCAE//B
    1DI02BIfYlpNgBy8Kf8Rjmw==",
  "dev": true,
  "dependencies": {
    "@types/http-proxy": "^1.17.8",
    "http-proxy": "^1.18.1",
    "is-glob": "^4.0.1",
    "is-plain-obj": "^3.0.0",
    "micromatch": "^4.0.2"
  },
  "engines": {
    "node": ">=12.0.0"
  },
  "peerDependencies": {
    "@types/express": "^4.17.13"
  },
  "peerDependenciesMeta": {
    "@types/express": {
      "optional": true
    }
  }
},
"node_modules/webpack-dev-server/node_modules/minimatch": {
  "version": "9.0.5",
  "resolved": "https://registry.npmjs.org/minimatch/-/minimatch-9.0.5.tgz",
  "integrity":
    "sha512-G6T0ZX48xgozx7587koeX9Ys2NYy6Gmv//P89sEte9V9whlapMNF4idKxnW2QtCc
    LiTWIb/wfCabAtAFWWhhBow==",
  "dev": true,
  "dependencies": {
    "brace-expansion": "^2.0.1"
  },
  "engines": {
    "node": ">=16 || 14 >=14.17"
  },
  "funding": {
    "url": "https://github.com/sponsors/isaacs"
  }
},
"node_modules/webpack-dev-server/node_modules/rimraf": {
  "version": "5.0.10",
  "resolved": "https://registry.npmjs.org/rimraf/-/rimraf-5.0.10.tgz",

```

```

"integrity":
"sha512-l0OE8wL34P4nJH/H2ffoaniAokM2qSmrtXHmlpvYr5AVVX8msAyW0I8NVJFDxISK4
u3Uh/f41cQheDVdnYijwQ==",
  "dev": true,
  "dependencies": {
    "glob": "^10.3.7"
  },
  "bin": {
    "rimraf": "dist/esm/bin.mjs"
  },
  "funding": {
    "url": "https://github.com/sponsors/isaacs"
  }
},
"node_modules/webpack-merge": {
  "version": "5.10.0",
  "resolved": "https://registry.npmjs.org/webpack-merge/-/webpack-merge-5.10.0.tgz",
  "integrity":
"sha512-+4zXKdx7UnO+1jaN4I2IHVD+mFvnlZQP/6ljaJVb4SZiwlKeUnrT5I0gkT8z+n4hKpC
+jpOv6O9R+gLtag7pSA==",
  "dev": true,
  "dependencies": {
    "clone-deep": "^4.0.1",
    "flat": "^5.0.2",
    "wildcard": "^2.0.0"
  },
  "engines": {
    "node": ">=10.0.0"
  }
},
"node_modules/webpack-sources": {
  "version": "3.2.3",
  "resolved": "https://registry.npmjs.org/webpack-sources/-/webpack-sources-3.2.3.tgz",
  "integrity":
"sha512-/DyMEOrDgLKKIG0fmvzt+4dUX/3Ghozwm6iPp8KRhvn+eQf9+Q7GWxVNMk3+u
CPWfdXYC4ExGBckIXdFEfH1w==",
  "dev": true,
  "engines": {
    "node": ">=10.13.0"
  }
},
"node_modules/webpack-subresource-integrity": {
  "version": "5.1.0",
  "resolved":
"https://registry.npmjs.org/webpack-subresource-integrity/-/webpack-subresource-integrity-5.
1.0.tgz",

```

```

"integrity":
"sha512-sacXoX+xd8r4WKsy9MvH/q/vBtEhr86cplmXwyg74pFipERKt6FmB8cXpeuh0ZLgcl
OIHI4WcIl7+R5L02xk9Q==",
  "dev": true,
  "dependencies": {
    "typed-assert": "^1.0.8"
  },
  "engines": {
    "node": ">= 12"
  },
  "peerDependencies": {
    "html-webpack-plugin": ">= 5.0.0-beta.1 < 6",
    "webpack": "^5.12.0"
  },
  "peerDependenciesMeta": {
    "html-webpack-plugin": {
      "optional": true
    }
  }
},
"node_modules/webpack/node_modules/ajv": {
  "version": "6.12.6",
  "resolved": "https://registry.npmjs.org/ajv/-/ajv-6.12.6.tgz",

```

```

"integrity":
"sha512-j3fVLgvTo527anyYyJOGTYJbG+vnnQYvE0m5mmkc1TK+nxAppkCLMIL0aZ4dbIVC
NoGShhm+kzE4ZUykBoMg4g==",
  "dev": true,
  "dependencies": {
    "fast-deep-equal": "^3.1.1",
    "fast-json-stable-stringify": "^2.0.0",
    "json-schema-traverse": "^0.4.1",
    "uri-js": "^4.2.2"
  },
  "funding": {
    "type": "github",
    "url": "https://github.com/sponsors/epoberezkin"
  }
},
"node_modules/webpack/node_modules/ajv-keywords": {
  "version": "3.5.2",
  "resolved": "https://registry.npmjs.org/ajv-keywords/-/ajv-keywords-3.5.2.tgz",

```

```

"integrity":
"sha512-5p6WTN0DdTGvQk6Vjceju19lgaHudalcfabD7yhDGeA6bcQnmL+CpveLJq/3hvfwd
1aof6L386Ougkx6RfyMIQ==",
  "dev": true,
  "peerDependencies": {
    "ajv": "^6.9.1"
  }
}

```

```

},
"node_modules/webpack/node_modules/json-parse-even-better-errors": {
  "version": "2.3.1",
  "resolved":
    "https://registry.npmjs.org/json-parse-even-better-errors/-/json-parse-even-better-errors-2.3.1
    .tgz",
  "integrity":
    "sha512-xyFwyhro/JEof6Ghe2iz2NcXoj2sloNsWr/XsERDK/oiPCfaNhl5ONfp+jQdAZRQQ0IJ
    WNzH9zIZF7li91kh2w==",
  "dev": true
},
"node_modules/webpack/node_modules/json-schema-traverse": {
  "version": "0.4.1",
  "resolved":
    "https://registry.npmjs.org/json-schema-traverse/-/json-schema-traverse-0.4.1.tgz",
  "integrity":
    "sha512-xbbCH5dCYU5T8LcEhhuh7HJ88HXuW3qsI3Y0zOZFKfZEHcpWiHU/Jxzk629Brsab
    /mMiHQti9wMP+845RPe3Vg==",
  "dev": true
},
"node_modules/webpack/node_modules/schema-utils": {
  "version": "3.3.0",
  "resolved": "https://registry.npmjs.org/schema-utils/-/schema-utils-3.3.0.tgz",
  "integrity":
    "sha512-pN/yOAvC+5rQ5nERGuwrjLIYvLTbCibnZ1I7B1LaiAz9BRBIE9GMgE/eqV30P7aJQ
    Uf7Ddimy/RsbYO/GrVGg==",
  "dev": true,
  "dependencies": {
    "@types/json-schema": "^7.0.8",
    "ajv": "^6.12.5",
    "ajv-keywords": "^3.5.2"
  },
  "engines": {
    "node": ">= 10.13.0"
  },
  "funding": {
    "type": "opencollective",
    "url": "https://opencollective.com/webpack"
  }
},
"node_modules/websocket-driver": {
  "version": "0.7.4",
  "resolved": "https://registry.npmjs.org/websocket-driver/-/websocket-driver-0.7.4.tgz",
  "integrity":
    "sha512-b17KeDIQVjvb0ssuSDF2cYXSg2iztliJ4B9WdsuB6J952qCPKmnVq4DyW5motlmX
    HDC1cBT/1UezrJVskw5zjg==",
  "dev": true,
  "dependencies": {

```

```

    "http-parser-js": ">=0.5.1",
    "safe-buffer": ">=5.1.0",
    "websocket-extensions": ">=0.1.1"
  },
  "engines": {
    "node": ">=0.8.0"
  }
},
"node_modules/websocket-extensions": {
  "version": "0.1.4",
  "resolved":
"https://registry.npmjs.org/websocket-extensions/-/websocket-extensions-0.1.4.tgz",
  "integrity":
"sha512-OqedPIGOfsDlo31UNwYbCFMSaO9m9G/0falHj5/dZFDMFqPTcx6UwqyOy3COEa
EOg/9VsGlpdqn62W5KhoKSpq==",
  "dev": true,
  "engines": {
    "node": ">=0.8.0"
  }
},
"node_modules/which": {
  "version": "1.3.1",
  "resolved": "https://registry.npmjs.org/which/-/which-1.3.1.tgz",
  "integrity":
"sha512-HxJdYWq1MTIQbJ3nw0cqssHoTNU267KlrDuGZ1WY1xDStUtKUhOaJmh112/TZm
HxxUfuJqPXSOM7tDyas0OSIQ==",
  "dev": true,
  "dependencies": {
    "isexe": "^2.0.0"
  },
  "bin": {
    "which": "bin/which"
  }
},
"node_modules/wildcard": {
  "version": "2.0.1",
  "resolved": "https://registry.npmjs.org/wildcard/-/wildcard-2.0.1.tgz",
  "integrity":
"sha512-CC1bOL87PIWSBhDcTrdeLo6eGT7mCFtrg0uJtqJUfYK+eJnzI8A1niH56uu7KMa5
XFrtiV+AQuHO3n7DsHnLQ==",
  "dev": true
},
"node_modules/wrap-ansi": {
  "version": "6.2.0",
  "resolved": "https://registry.npmjs.org/wrap-ansi/-/wrap-ansi-6.2.0.tgz",
  "integrity":
"sha512-r6lPGLs7gS9gWpKQWv6gpVcbK/q4G3Lw0DQi2XLlKS4I7uIwfqnPKvWo78Lg7541x4Q285WC340XUJYw==",

```



```

    "dev": true,
    "dependencies": {
      "ansi-styles": "^4.0.0",
      "string-width": "^4.1.0",
      "strip-ansi": "^6.0.0"
    },
    "engines": {
      "node": ">=8"
    }
  },
  "node_modules/wrap-ansi-cjs": {
    "name": "wrap-ansi",
    "version": "7.0.0",
    "resolved": "https://registry.npmjs.org/wrap-ansi/-/wrap-ansi-7.0.0.tgz",
    "integrity":
"sha512-YVGIj2kamLSTxw6NsZjoBxfSwsn0ycdesmc4p+Q21c5zPuZ1pl+NfxVdxPtdHvmNV
OQ6XSYG4AUtyt/Fi7D16Q==",
    "dev": true,
    "dependencies": {
      "ansi-styles": "^4.0.0",
      "string-width": "^4.1.0",
      "strip-ansi": "^6.0.0"
    },
    "engines": {
      "node": ">=10"
    },
    "funding": {
      "url": "https://github.com/chalk/wrap-ansi?sponsor=1"
    }
  },
  "node_modules/wrap-ansi-cjs/node_modules/ansi-styles": {
    "version": "4.3.0",
    "resolved": "https://registry.npmjs.org/ansi-styles/-/ansi-styles-4.3.0.tgz",
    "integrity":
"sha512-zbB9rCJAT1rbjiVDb2hqKFHNYLxgtk8NURxZ3IZwD3F6NtXBXZQCnnSi1Lkx+IDohd
PIFp222wVALIheZJQSEg==",
    "dev": true,
    "dependencies": {
      "color-convert": "^2.0.1"
    },
    "engines": {
      "node": ">=8"
    },
    "funding": {
      "url": "https://github.com/chalk/ansi-styles?sponsor=1"
    }
  },
  "node_modules/wrap-ansi-cjs/node_modules/color-convert": {

```

```

"version": "2.0.1",
"resolved": "https://registry.npmjs.org/color-convert/-/color-convert-2.0.1.tgz",
"integrity":
"sha512-RRECPsj7iu/xb5oKYcsFHSppFNnsj/52OVTRKb4zP5onXwVF3zVmmToNcOfGC+C
RDpfK/U584fMg38ZHCaElKQ==",
  "dev": true,
  "dependencies": {
    "color-name": "~1.1.4"
  },
  "engines": {
    "node": ">=7.0.0"
  }
},
"node_modules/wrap-ansi-cjs/node_modules/color-name": {
  "version": "1.1.4",
  "resolved": "https://registry.npmjs.org/color-name/-/color-name-1.1.4.tgz",
  "integrity":
"sha512-dOy+3AuW3a2wNbZHIuMZpTcgjGuLU/uBL/ubcZF9OXbDo8ff4O8yVp5Bf0efS8uEo
Yo5q4Fx7dY9OgQGxgAsQA==",
  "dev": true
},
"node_modules/wrap-ansi-cjs/node_modules/emoji-regex": {
  "version": "8.0.0",
  "resolved": "https://registry.npmjs.org/emoji-regex/-/emoji-regex-8.0.0.tgz",
  "integrity":
"sha512-MSjYzcWNOA0ewAHpz0MxpYFvwg6yjy1NG3xteoqz644VCo/RPgnr1/GGt+ic3iJTz
Q8Eu3TdM14SawnVUmGE6A==",
  "dev": true
},
"node_modules/wrap-ansi-cjs/node_modules/is-fullwidth-code-point": {
  "version": "3.0.0",
  "resolved":
"https://registry.npmjs.org/is-fullwidth-code-point/-/is-fullwidth-code-point-3.0.0.tgz",
  "integrity":
"sha512-zyymm5+u+sCsSWyD9qNaejV3DFvhCKclKdizYaJUuHA83RLjb7nSuGnddCHGv0hk
+KY7BMAlisWeK4Ueg6EV6XQg==",
  "dev": true,
  "engines": {
    "node": ">=8"
  }
},
"node_modules/wrap-ansi-cjs/node_modules/string-width": {
  "version": "4.2.3",
  "resolved": "https://registry.npmjs.org/string-width/-/string-width-4.2.3.tgz",
  "integrity":
"sha512-wKyQRQpjJ0sIp62ErSZdGsjMJWsap5oRNihHhu6G7JVO/9jIB6UyevL+tXuOqrng8j/
cxKTWyuWUwvSTriiZz/g==",
  "dev": true,

```

```

"dependencies": {
  "emoji-regex": "^8.0.0",
  "is-fullwidth-code-point": "^3.0.0",
  "strip-ansi": "^6.0.1"
},
"engines": {
  "node": ">=8"
}
},
"node_modules/wrap-ansi/node_modules/ansi-styles": {
  "version": "4.3.0",
  "resolved": "https://registry.npmjs.org/ansi-styles/-/ansi-styles-4.3.0.tgz",
  "integrity":
"sha512-zbB9rCJAT1rbjiVDb2hqKFHNYLxgtk8NURxZ3IZwD3F6NtxbXZQCnnSi1Lkx+IDohd
PIFp222wVALIheZJQSEg==",
  "dev": true,
  "dependencies": {
    "color-convert": "^2.0.1"
  },
  "engines": {
    "node": ">=8"
  },
  "funding": {
    "url": "https://github.com/chalk/ansi-styles?sponsor=1"
  }
},
"node_modules/wrap-ansi/node_modules/color-convert": {
  "version": "2.0.1",
  "resolved": "https://registry.npmjs.org/color-convert/-/color-convert-2.0.1.tgz",
  "integrity":
"sha512-RRECPsj7iu/xb5oKYcsFHSppFNnsj/52OVTRKb4zP5onXwVF3zVmmToNcOfGC+C
RDpfK/U584fMg38ZHCaElKQ==",
  "dev": true,
  "dependencies": {
    "color-name": "~1.1.4"
  },
  "engines": {
    "node": ">=7.0.0"
  }
},
"node_modules/wrap-ansi/node_modules/color-name": {
  "version": "1.1.4",
  "resolved": "https://registry.npmjs.org/color-name/-/color-name-1.1.4.tgz",
  "integrity":
"sha512-dOy+3AuW3a2wNbZHIuMZpTcgjGuLU/uBL/ubcZF9OXbDo8ff4O8yVp5Bf0efS8uEo
Yo5q4Fx7dY9OgQGXgAsQA==",
  "dev": true
},

```

```

"node_modules/wrap-ansi/node_modules/emoji-regex": {
  "version": "8.0.0",
  "resolved": "https://registry.npmjs.org/emoji-regex/-/emoji-regex-8.0.0.tgz",
  "integrity":
"sha512-MSjYzcWNOA0ewAHpz0MxpYFvwg6yjy1NG3xteoqz644VCo/RPgnr1/GGt+ic3iJTz
Q8Eu3TdM14SawnVUmGE6A==",
  "dev": true
},
"node_modules/wrap-ansi/node_modules/is-fullwidth-code-point": {
  "version": "3.0.0",
  "resolved":
"https://registry.npmjs.org/is-fullwidth-code-point/-/is-fullwidth-code-point-3.0.0.tgz",
  "integrity":
"sha512-zymm5+u+sCsSWyD9qNaejV3DFvhCKclKdizYaJUuHA83RLjb7nSuGnddCHGv0hk
+KY7BMAlIsWeK4Ueg6EV6XQg==",
  "dev": true,
  "engines": {
    "node": ">=8"
  }
},
"node_modules/wrap-ansi/node_modules/string-width": {
  "version": "4.2.3",
  "resolved": "https://registry.npmjs.org/string-width/-/string-width-4.2.3.tgz",
  "integrity":
"sha512-wKyQRQpjJ0sIp62ErSZdGsjMJWsap5oRNihHhu6G7JVO/9jIB6UyevL+tXuOqrng8j/
cxKTWYwUwvSTriiZz/g==",
  "dev": true,
  "dependencies": {
    "emoji-regex": "^8.0.0",
    "is-fullwidth-code-point": "^3.0.0",
    "strip-ansi": "^6.0.1"
  },
  "engines": {
    "node": ">=8"
  }
},
"node_modules/wrappy": {
  "version": "1.0.2",
  "resolved": "https://registry.npmjs.org/wrappy/-/wrappy-1.0.2.tgz",
  "integrity":
"sha512-l4Sp/DRseor9wL6EvV2+TuQn63dMkPjZ/sp9XkghTEbV9KIPS1xUsZ3u7/IQO4wxtc
FB4bgpQPRcR3QCvezPcQ==",
  "dev": true
},
"node_modules/ws": {
  "version": "8.17.1",
  "resolved": "https://registry.npmjs.org/ws/-/ws-8.17.1.tgz",

```

```

"integrity":
"sha512-6XQFvXTkbfUOZOKKILFG1PDK2NDQs4azKQI26T0YS5CxqWLGXajbPZ+h4gZekJ
yRqFU8pvnAbbs/3TgRPy+GQ==",
  "dev": true,
  "engines": {
    "node": ">=10.0.0"
  },
  "peerDependencies": {
    "bufferutil": "^4.0.1",
    "utf-8-validate": ">=5.0.2"
  },
  "peerDependenciesMeta": {
    "bufferutil": {
      "optional": true
    },
    "utf-8-validate": {
      "optional": true
    }
  }
},
"node_modules/y18n": {
  "version": "5.0.8",
  "resolved": "https://registry.npmjs.org/y18n/-/y18n-5.0.8.tgz",

```

```

"integrity":
"sha512-0pfFzegeDWJHJIAmTLRP2DwHjdF5s7jo9tuztdQxAhINCdvS+3nGINqPd00AphqJR
/0LhANUS6/+7SCb98YOfA==",
  "dev": true,
  "engines": {
    "node": ">=10"
  }
},
"node_modules/yallist": {
  "version": "3.1.1",
  "resolved": "https://registry.npmjs.org/yallist/-/yallist-3.1.1.tgz",

```

```

"integrity":
"sha512-a4UGQaWPH59mOXUYnAG2ewncQS4i4F43Tv3JoAM+s2VDAmS9NsK8GpDMLr
CHPksFT7h3K6TOoUNn2pb7RoXx4g==",
  "dev": true
},
"node_modules/yargs": {
  "version": "17.7.2",
  "resolved": "https://registry.npmjs.org/yargs/-/yargs-17.7.2.tgz",

```

```

"integrity":
"sha512-7dSzzRQ++CKnNI/krKnYRV7JKKPUXMEh61soaHKg9mrWEhzFWWhFnxPxGI+69c
D1Ou63C13NUPCnmlcrvqCuM6w==",
  "dev": true,
  "dependencies": {
    "cliui": "^8.0.1",

```

```

    "escalade": "^3.1.1",
    "get-caller-file": "^2.0.5",
    "require-directory": "^2.1.1",
    "string-width": "^4.2.3",
    "y18n": "^5.0.5",
    "yargs-parser": "^21.1.1"
  },
  "engines": {
    "node": ">=12"
  }
},
"node_modules/yargs-parser": {
  "version": "21.1.1",
  "resolved": "https://registry.npmjs.org/yargs-parser/-/yargs-parser-21.1.1.tgz",
  "integrity":
"sha512-tVpsJW7DdjecAiFpbIB1e3qxIQsE6NoPc5/eTdrbbIC4h0LVsWhnoa3g+m2HclBlujHz
sxZ4VJVA+GUuc2/LBw==",
  "dev": true,
  "engines": {
    "node": ">=12"
  }
},
"node_modules/yargs/node_modules/emoji-regex": {
  "version": "8.0.0",
  "resolved": "https://registry.npmjs.org/emoji-regex/-/emoji-regex-8.0.0.tgz",
  "integrity":
"sha512-MSjYzcWNOA0ewAHpz0MxpYFvwg6yjy1NG3xteoqz644VCo/RPgnr1/GGt+ic3iJTz
Q8Eu3TdM14SawnVUmGE6A==",
  "dev": true
},
"node_modules/yargs/node_modules/is-fullwidth-code-point": {
  "version": "3.0.0",
  "resolved":
"https://registry.npmjs.org/is-fullwidth-code-point/-/is-fullwidth-code-point-3.0.0.tgz",
  "integrity":
"sha512-zymm5+u+sCsSWyD9qNaejV3DFvhCKcKdizYaJUuHA83RLjb7nSuGnddCHGv0hk
+KY7BMAlsWeK4Ueg6EV6XQg==",
  "dev": true,
  "engines": {
    "node": ">=8"
  }
},
"node_modules/yargs/node_modules/string-width": {
  "version": "4.2.3",
  "resolved": "https://registry.npmjs.org/string-width/-/string-width-4.2.3.tgz",
  "integrity":
"sha512-wKyQRQpjJ0sIp62ErSZdGsjMJWsap5oRNihHhu6G7JVO/9jIB6UyevL+tXuOqrng8j/
cxKTWYwUwvSTriiZz/g==",

```

```

    "dev": true,
    "dependencies": {
      "emoji-regex": "^8.0.0",
      "is-fullwidth-code-point": "^3.0.0",
      "strip-ansi": "^6.0.1"
    },
    "engines": {
      "node": ">=8"
    }
  },
  "node_modules/yocto-queue": {
    "version": "1.1.1",
    "resolved": "https://registry.npmjs.org/yocto-queue/-/yocto-queue-1.1.1.tgz",
    "integrity":
"sha512-b4JR1PFR10y1mKjhHY9LaGo6tmrgjit7hxVleAmyMw3jegXR4dhYqLaQF5zMXZxY
7tLpMyJeLjr1C4rLmkVe8g==",
    "dev": true,
    "engines": {
      "node": ">=12.20"
    },
    "funding": {
      "url": "https://github.com/sponsors/sindresorhus"
    }
  },
  "node_modules/yoctocolors-cjs": {
    "version": "2.1.2",
    "resolved": "https://registry.npmjs.org/yoctocolors-cjs/-/yoctocolors-cjs-2.1.2.tgz",
    "integrity":
"sha512-cYVsTjKl8b+FrnidjibDWskAv7UKOfcwaVZdp/it9n1s9fU3lkgDbhdIRKCW4JDsAIEC
JY0ytoVPT3sK6kideA==",
    "dev": true,
    "engines": {
      "node": ">=18"
    },
    "funding": {
      "url": "https://github.com/sponsors/sindresorhus"
    }
  },
  "node_modules/zone.js": {
    "version": "0.14.10",
    "resolved": "https://registry.npmjs.org/zone.js/-/zone.js-0.14.10.tgz",
    "integrity":
"sha512-YGAhaO7J5ywOXW6lnXNILmfU194F8IVgu7bRntUF3TiG8Y3nBK0x1UJJuHUP/e8l
yihkjCYqhCScpSwnlaSRkQ=="
  }
}
}
}

```

lixo-no-ponto-front\package.json ----

```
{
  "name": "lixo-no-ponto-front",
  "version": "0.0.0",
  "scripts": {
    "ng": "ng",
    "start": "node load-env.js && ng serve ",
    "start-prod": "ng serve --configuration production",
    "start-with-ssl": "node load-env.js && ng serve --host 10.0.0.100 --ssl true --ssl-key
./ssl/server.key --ssl-cert ./ssl/server.crt --configuration production",
    "build": "node load-env.js && ng build --configuration production",
    "watch": "ng build --watch --configuration development",
    "test": "ng test"
  },
  "engines": {
    "node": "20.11.1",
    "npm": "10.4.0"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "^18.1.0",
    "@angular/cdk": "^18.1.3",
    "@angular/common": "^18.1.0",
    "@angular/compiler": "^18.1.0",
    "@angular/core": "^18.1.0",
    "@angular/forms": "^18.1.0",
    "@angular/google-maps": "^18.2.3",
    "@angular/material": "^18.1.3",
    "@angular/platform-browser": "^18.1.0",
    "@angular/platform-browser-dynamic": "^18.1.0",
    "@angular/router": "^18.1.0",
    "@angular/service-worker": "^18.1.0",
    "@auth0/angular-jwt": "^5.2.0",
    "crypto-js": "^4.2.0",
    "dexie": "^4.0.8",
    "dotenv": "^16.4.5",
    "express": "^4.21.0",
    "ngx-mask": "^18.0.0",
    "ngx-toastr": "^19.0.0",
    "path": "^0.12.7",
    "rxjs": "~7.8.0",
    "tslib": "^2.3.0",
    "zone.js": "~0.14.3"
  },
  "devDependencies": {
```



```

"@angular-devkit/build-angular": "^18.1.3",
"@angular/cli": "^18.1.3",
"@angular/compiler-cli": "^18.1.0",
"@types/crypto-js": "^4.2.2",
"@types/googlemaps": "^3.43.3",
"@types/jasmine": "~5.1.0",
"@types/node": "^22.1.0",
"angular-cli-ghpages": "^2.0.1",
"jasmine-core": "~5.1.0",
"karma": "~6.4.0",
"karma-chrome-launcher": "~3.2.0",
"karma-coverage": "~2.2.0",
"karma-jasmine": "~5.1.0",
"karma-jasmine-html-reporter": "~2.1.0",
"typescript": "~5.5.2"
}
}

```

lixo-no-ponto-front\public\manifest.webmanifest ----

```

{
  "name": "Lixo no Ponto",
  "short_name": "Lixo no Ponto",
  "theme_color": "#ffffff",
  "background_color": "#ffffff",
  "display": "standalone",
  "scope": "./",
  "start_url": "./",
  "description": "O Lixo no Ponto é um aplicativo que tem como objetivo facilitar a vida dos cidadãos, informando sobre a coleta de resíduos na cidade.",
  "id": "com.ajbb.lixonoponto",
  "icons": [
    {
      "src": "icons/android-icon-36x36.png",
      "sizes": "36x36",
      "type": "image/png",
      "density": "0.75"
    },
    {
      "src": "icons/android-icon-48x48.png",
      "sizes": "48x48",
      "type": "image/png",
      "density": "1.0"
    },
    {
      "src": "icons/android-icon-72x72.png",

```

```

"sizes": "72x72",
"type": "image/png",
"density": "1.5"
},
{
"src": "icons/android-icon-96x96.png",
"sizes": "96x96",
"type": "image/png",
"density": "2.0"
},
{
"src": "icons/android-icon-144x144.png",
"sizes": "144x144",
"type": "image/png",
"density": "3.0"
},
{
"src": "icons/android-icon-192x192.png",
"sizes": "192x192",
"type": "image/png",
"density": "4.0"
},
{
"src": "icons/android-chrome-512x512.png",
"sizes": "512x512",
"type": "image/png"
}
],
"screenshots": [
{
"src": "assets/tela_inicial.png",
"sizes": "1440x2960",
"type": "image/png",
"form_factor": "wide"
},
{
"src": "assets/tela_login.png",
"sizes": "1440x2960",
"type": "image/png",
"form_factor": "wide"
},
{
"src": "assets/tela_empresas_reciclagem.png",
"sizes": "1440x2960",
"type": "image/png",
"form_factor": "narrow"
},
{

```

```

    "src": "assets/tela_rota_e_horarios_coleta_de_lixo.png",
    "sizes": "1440x2960",
    "type": "image/png",
    "form_factor": "narrow"
  }
]
}

```

lixo-no-ponto-front\README.md ----

LixoNoPontoFront

This project was generated with [Angular CLI](<https://github.com/angular/angular-cli>) version 18.1.3.

Development server

Run `ng serve` for a dev server. Navigate to `http://localhost:4200/`. The application will automatically reload if you change any of the source files.

Code scaffolding

Run `ng generate component component-name` to generate a new component. You can also use `ng generate directive|pipe|service|class|guard|interface|enum|module`.

Build

Run `ng build` to build the project. The build artifacts will be stored in the `dist/` directory.

Running unit tests

Run `ng test` to execute the unit tests via [Karma](<https://karma-runner.github.io>).

Running end-to-end tests

Run `ng e2e` to execute the end-to-end tests via a platform of your choice. To use this command, you need to first add a package that implements end-to-end testing capabilities.

Further help

To get more help on the Angular CLI use `ng help` or go check out the [Angular CLI Overview and Command Reference](<https://angular.dev/tools/cli>) page.

lixo-no-ponto-front\src\app\account\account-routing.module.ts ----

```

import {NgModule} from '@angular/core';
import {RouterModule, Routes} from '@angular/router';
import {RegisterComponent} from "../register/register.component";
import {LoginComponent} from "../login/login.component";
import {AccountComponent} from "../account.component";
import {InfoComponent} from "../info/info.component";
import {AuthGuard} from "../shared/guard/AuthGuard";
import {AlreadyLoggedGuard} from "../shared/guard/AlreadyLoggedGuard";

const routes: Routes = [
  {
    path: "",
    component: AccountComponent,
    children: [
      {
        path: "",
        redirectTo: 'login',
        pathMatch: 'full'
      },
      {
        path: 'login',
        component: LoginComponent,
        canActivate: [AlreadyLoggedGuard]
      },
      {
        path: 'register',
        component: RegisterComponent,
        canActivate: [AlreadyLoggedGuard]
      },
      {
        path: 'info',
        component: InfoComponent,
        canActivate: [AuthGuard]
      }
    ]
  }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule]
})
export class AccountRoutingModule {
}

```

lixo-no-ponto-front\src\app\account\account.component.html ----

```
<section class="account-section">
  <router-outlet></router-outlet>
</section>
```

lixo-no-ponto-front\src\app\account\account.component.scss ----

```
.account-section {
  display: flex;
  justify-content: center;
  align-items: center;
  background-color: #CAD787;
  padding: 20px 28px;
  border-radius: 8px;
  margin: 18px;
}
```

lixo-no-ponto-front\src\app\account\account.component.ts ----

```
import {Component} from '@angular/core';

@Component({
  selector: 'app-account',
  templateUrl: './account.component.html',
  styleUrls: ['./account.component.scss']
})
export class AccountComponent {

}
```

lixo-no-ponto-front\src\app\account\account.module.ts ----

```
import {NgModule} from '@angular/core';
import {CommonModule} from '@angular/common';

import {AccountRoutingModule} from './account-routing.module';
import {AccountComponent} from './account.component';
import {RegisterComponent} from './register/register.component';
import {LoginComponent} from './login/login.component';
import {InfoComponent} from './info/info.component';
import {MatError, MatFormField, MatLabel} from '@angular/material/form-field';
```

```
import {MatInput} from "@angular/material/input";
import {MatButton} from "@angular/material/button";
import {FormsModule, ReactiveFormsModule} from "@angular/forms";
import {AccountService} from "../account.service";
```

```
@NgModule({
  declarations: [
    AccountComponent,
    RegisterComponent,
    LoginComponent,
    InfoComponent
  ],
  imports: [
    CommonModule,
    AccountRoutingModule,
    MatFormField,
    MatInput,
    MatButton,
    MatLabel,
    ReactiveFormsModule,
    FormsModule,
    MatError
  ],
  providers: [
    AccountService
  ]
})
export class AccountModule {
}
```

lixo-no-ponto-front\src\app\account\account.service.ts ----

```
import {Injectable} from '@angular/core';
import {UserLoginModel} from "../shared/interface/user-login.model";
import {map, Observable, ReplaySubject} from "rxjs";
import {API} from "../api/Endpoint";
import {HttpService} from "../shared/service/http/http.service";
import {User, UserClass} from "../shared/interface/user.model";
import {AuthService} from "../shared/service/auth/auth.service";
import {UserRegisterModel} from "../shared/interface/user-register.model";
```

```
@Injectable({
  providedIn: "root"
})
export class AccountService {
```

```

private user = new ReplaySubject<User>(1);

constructor(private httpService: HttpService, private authService: AuthService) {
  this.authService.isLoggedIn.subscribe(loggedIn => {
    if (!loggedIn) {
      this.user = new ReplaySubject<User>(1);
    }
  });
}

public login(request: UserLoginModel): Observable<string> {
  return this.httpService.post<string>({url: API.endpoints.user.login.url, body: request,
responseType: 'text'});
}

public register(request: UserRegisterModel): Observable<User> {
  return this.httpService.post<User>({url: API.endpoints.user.register.url, body: request});
}

public getUser(): Observable<User> {
  if (!this.user.observers.length) {
    this.httpService.get<User>({url: API.endpoints.user.info.url}).subscribe(user => {
      this.user.next(user);
    });
  }
  return this.user.asObservable();
}

public isAdmin(): Observable<boolean> {
  return this.getUser().pipe(map(user => user.classesDeUsuario.some(classe => classe.id
=== UserClass.ADMIN)));
}
}

```

lixo-no-ponto-front\src\app\account\info\info.component.html ----

```

<div class="container flex center-full column">
  <h1>Perfil</h1>
  <div class="form flex center-full column w-100">
    <mat-form-field appearance="fill" class="w-100">
      <mat-label>Nome:</mat-label>
      <input [ngModel]="user?.nome" disabled matInput placeholder="Nome do usuário">
    </mat-form-field>
    <mat-form-field appearance="fill" class="w-100">
      <mat-label>E-mail:</mat-label>
      <input [ngModel]="user?.email" disabled matInput placeholder="E-mail do usuário">

```

```

    </mat-form-field>
  </div>
  <div>
    <p>Quantidade de pontos verificados: {{ user?.pontosVerificados }}</p>
    <p>Data de cadastro: {{ user?.dataCadastro | date }}</p>
  </div>
  <button (click)="goBack()" color="primary" mat-raised-button>Voltar</button>
</div>

```

lixo-no-ponto-front\src\app\account\info\info.component.scss ----

```

.container {
  padding: 16px 8px;
}

```

lixo-no-ponto-front\src\app\account\info\info.component.ts ----

```

import {Component} from '@angular/core';
import {AccountService} from "../account.service";
import {Router} from "@angular/router";
import {User} from "../../shared/interface/user.model";

@Component({
  selector: 'app-info',
  templateUrl: './info.component.html',
  styleUrls: ['./info.component.scss']
})
export class InfoComponent {
  public user: User | undefined;

  constructor(private accountService: AccountService, private router: Router) {
  }

  ngOnInit(): void {
    this.accountService.getUser().subscribe(user => {
      this.user = user;
    });
  }

  public goBack(): void {
    this.router.navigate(['']);
  }
}

```


lixo-no-ponto-front\src\app\account\login\login.component.html ----

```

<div class="container">
  
  <h1>Login!</h1>
  <p>Bem-vindo de volta, você fez falta!</p>
  <form (ngSubmit)="onSubmit()" [formGroup]="form">
    <mat-form-field>
      <input formControlName="email" matInput placeholder="Email" type="email">
      <mat-error
        *ngIf="form.get('email')?.touched && (form.get('email')?.hasError('email') &&
!form.get('email')?.hasError('required'))">
        Revise e preencha corretamente.
      </mat-error>
      <mat-error
        *ngIf="form.get('email')?.touched &&
(form.get('email')?.hasError('required'))">
        O campo é <strong>obrigatório</strong>
      </mat-error>
    </mat-form-field>
    <mat-form-field>
      <input formControlName="password" matInput placeholder="Senha" type="password">
      <mat-error
        *ngIf="form.get('password')?.touched && (form.get('password')?.hasError('password')
&& !form.get('password')?.hasError('required'))">
        Revise e preencha corretamente.
      </mat-error>
      <mat-error
        *ngIf="form.get('password')?.touched &&
(form.get('password')?.hasError('required'))">
        O campo é <strong>obrigatório</strong>
      </mat-error>
    </mat-form-field>
    <button color="primary" mat-raised-button>Entrar</button>
    <button (click)="navigateToRegister()" color="secondary" mat-button type="button">Cria
nova conta</button>
  </form>
</div>

```

lixo-no-ponto-front\src\app\account\login\login.component.scss ----

```

.container, form {
  display: flex;
  flex-direction: column;
  align-items: center;

```

```

    justify-content: center;
  }

```

```

.container {
  padding: 16px 8px;
}

```

```

.logo {
  max-width: 230px;
  border-radius: 100%;
  margin-bottom: 18px;
}

```

lixo-no-ponto-front\src\app\account\login\login.component.ts ----

```

import {Component} from '@angular/core';
import {Router} from "@angular/router";
import {FormBuilder, FormGroup, Validators} from "@angular/forms";
import {AccountService} from "../account.service";
import {AuthService} from "../../shared/service/auth/auth.service";
import {HttpErrorResponse} from "@angular/common/http";
import {catchError} from "rxjs/operators";
import {EMPTY} from "rxjs";

```

```

@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.scss']
})
export class LoginComponent {
  public form: FormGroup;
  public error: HttpErrorResponse | undefined;

```

```

  constructor(private router: Router,
               private formBuilder: FormBuilder,
               private authService: AuthService,
               private accountService: AccountService) {
    this.form = this.formBuilder.group({
      email: ['', Validators.required],
      password: ['', Validators.required]
    });
  }

```

```

  public navigateToRegister(): void {
    this.router.navigate(['account/register']);
  }

```

```

public onSubmit(): void {
  if (this.form.valid) {
    this.accountService.login(this.form.value).pipe(
      catchError(err => {
        this.error = err;
        return EMPTY;
      })
    ).subscribe(response => {
      this.authService.login(response);
      this.router.navigate(['/']);
    });
  }
}
}
}

```

lixo-no-ponto-front\src\app\account\register\register.component.html ----

```

<div class="container">
  
  <h1>Registre-se!</h1>
  <p>Registre-se para começar a visualizar os pontos de descarte!</p>
  <form (ngSubmit)="onSubmit()" [formGroup]="form">
    <mat-form-field>
      <input formControlName="nome" matInput placeholder="Nome">
      <mat-error
        *ngIf="form.get('nome')?.touched && (form.get('nome')?.hasError('nome') &&
form.get('nome')?.hasError('required'))">
        Revise e preencha corretamente.
      </mat-error>
      <mat-error
        *ngIf="form.get('nome')?.touched &&
(form.get('nome')?.hasError('required'))">
        O campo é <strong>obrigatório</strong>
      </mat-error>
    </mat-form-field>
    <mat-form-field>
      <input formControlName="email" matInput placeholder="Email" type="email">
      <mat-error
        *ngIf="form.get('email')?.touched && (form.get('email')?.hasError('email') &&
form.get('email')?.hasError('required'))">
        Revise e preencha corretamente.
      </mat-error>
      <mat-error
        *ngIf="form.get('email')?.touched &&
(form.get('email')?.hasError('required'))">
        O campo é <strong>obrigatório</strong>

```

```

    </mat-error>
  </mat-form-field>
  <mat-form-field>
    <input formControlName="senha" matInput placeholder="Senha" type="password">
    <mat-error
      *ngIf="form.get('senha')?.touched && (form.get('senha')?.hasError('senha') &&
form.get('senha')?.hasError('required'))">
      Revise e preencha corretamente.
    </mat-error>
    <mat-error
      *ngIf="form.get('senha')?.touched &&
(form.get('senha')?.hasError('required'))">
      O campo é <strong>obrigatório</strong>
    </mat-error>
  </mat-form-field>
  <mat-form-field>
    <input formControlName="confirmarSenha" matInput placeholder="Confirmar Senha"
type="password">
    <mat-error
      *ngIf="form.get('confirmarSenha')?.touched &&
form.get('confirmarSenha')?.hasError('required'); else invalid">
      O campo é <strong>obrigatório</strong>
    </mat-error>
    <ng-template #invalid>
      <mat-error *ngIf="form.get('confirmarSenha')?.hasError('confirmarSenha'); else
notSame">
        Revise e preencha corretamente.
      </mat-error>
    </ng-template>
    <ng-template #notSame>
      <mat-error *ngIf="form.get('confirmarSenha')?.hasError('notSame')">
        As senhas não conferem.
      </mat-error>
    </ng-template>
  </mat-form-field>
  <button [disabled]="form.invalid" color="primary" mat-raised-button>Registrar</button>
  <button (click)="navigateToLogin()" color="secondary" mat-button type="button">Já tenho
uma conta</button>
</form>
</div>

```

lixo-no-ponto-front\src\app\account\register\register.component.scss ----

```

.container, form {
  display: flex;
  flex-direction: column;
  align-items: center;

```

```

    justify-content: center;
  }

```

```

.container {
  padding: 16px 8px;
}

```

```

.logo {
  max-width: 230px;
  border-radius: 100%;
  margin-bottom: 18px;
}

```

lixo-no-ponto-front\src\app\account\register\register.component.ts ----

```

import {Component} from '@angular/core';
import {Router} from "@angular/router";
import {AbstractControl, FormBuilder, FormGroup, ValidationErrors, Validators} from
"@angular/forms";
import {AccountService} from "../account.service";
import {ToastrService} from "ngx-toastr";

```

```

@Component({
  selector: 'app-register',
  templateUrl: './register.component.html',
  styleUrls: ['./register.component.scss']
})

```

```

export class RegisterComponent {
  public form: FormGroup;

```

```

    constructor(private toastrService: ToastrService, private router: Router, private
accountService: AccountService, private formBuilder: FormBuilder) {
      this.form = this.formBuilder.group({
        nome: ['', Validators.required],
        email: ['', Validators.required],
        senha: ['', Validators.required],
        confirmarSenha: ['', [Validators.required, this.checkPasswords.bind(this)]],
      });
    }
}

```

```

public onSubmit(): void {
  if (this.form.valid) {
    const {nome, email, senha} = this.form.value;
    const user = {nome, email, senha};

```

```

    this.accountService.register(user).subscribe(() => {
      this.router.navigate(['account/login']);
      this.toastrService.info(`Usuário cadastrado com sucesso!`, 'Sucesso!');
    });
  }
}

```

```

public navigateToLogin(): void {
  this.router.navigate(['account/login']);
}

```

```

private checkPasswords(control: AbstractControl): ValidationErrors | null {
  if (this.form) {
    const senha = this.form.get('senha')?.value;
    const confirmarSenha = control.value;
    return senha === confirmarSenha ? null : {notSame: true};
  }
  return null;
}
}

```

lixo-no-ponto-front\src\app\api\endpoint.interface.ts ----

```

import {EndpointService} from "../endpoint.service";

export interface ParamRouteProp {
  [key: string]: string | number | boolean | undefined;
}

export interface IEndpoints {
  user: {
    login: EndpointService;
    register: EndpointService;
    info: EndpointService;
  },
  urbanWaste: {
    list: EndpointService;
  },
  truckType: {
    list: EndpointService;
  }
  recyclingCompany: {
    list: EndpointService;
    create: EndpointService;
  },
}

```

```

trashCollectionRoutes: {
  list: EndpointService;
  create: EndpointService;
},
ibge: {
  states: EndpointService;
  cities: EndpointService;
},
disposalPoint: {
  list: EndpointService;
  create: EndpointService;
  update: EndpointService;
  verify: EndpointService;
  delete: EndpointService;
}
}

```

lixo-no-ponto-front\src\app\api\endpoint.service.ts ----

```

import {ParamRouteProp} from "./endpoint.interface";

export class EndpointService {
  public readonly _urlBase: string;
  private readonly _path: string;

  constructor(
    private urlBase: string,
    private path: string,
  ){
    this._urlBase = this.urlBase;
    this._path = this.path;
  }

  public get url(): string {
    return this._urlBase + this._path;
  }

  public paramRoute(path: string, params: ParamRouteProp): string {
    let concatenatedPath = path;

    Object.keys(params).forEach((attribute: string) => {
      concatenatedPath = concatenatedPath.replace(
        `{{${attribute}}}`,
        `${params[attribute]}`
      );
    });
  }
}

```

```

});

return `${this.url}${concatenatedPath}`;
}

public queryString(
  queries: ParamRouteProp[],
  paramRouteConfig?: { path: string; params: ParamRouteProp }
): string {
  let url: string = this.url;

  if (paramRouteConfig) {
    url = this.paramRoute(paramRouteConfig.path, paramRouteConfig.params);
  }

  const queryString = queries
    .map((query: ParamRouteProp) => Object.entries(query)
      .map((entry) => {
        if (entry.length > 0) {
          return `${entry[0]}=${entry[1]}`;
        }
      })
      .filter(Boolean)
    )
    .join('&');

  return `${url}${queryString ? '?' + queryString : ''}`;
}
}

```

lixo-no-ponto-front\src\app\api\Endpoint.ts ----

```

import {IEndpoints} from "./endpoint.interface";
import {EndpointService} from "./endpoint.service";
import {environment} from "../../environments/environment";

export class API {
  public static endpoints: IEndpoints;

  public static init(): void {
    API.endpoints = {
      user: {
        login: new EndpointService(environment.baseUrl, '/user/auth/login'),
        register: new EndpointService(environment.baseUrl, '/user/auth/register'),

```



```

    info: new EndpointService(environment.baseUrl, '/user/info')
  },
  urbanWaste: {
    list: new EndpointService(environment.baseUrl, '/waste-list')
  },
  recyclingCompany: {
    list: new EndpointService(environment.baseUrl, '/recycling-company'),
    create: new EndpointService(environment.baseUrl, '/recycling-company/create')
  },
  trashCollectionRoutes: {
    list: new EndpointService(environment.baseUrl, '/truck-routing'),
    create: new EndpointService(environment.baseUrl, '/truck-routing/create')
  },
  truckType: {
    list: new EndpointService(environment.baseUrl, '/truck-type'),
  },
  ibge: {
    states: new EndpointService(environment.ibgeUrl, '/estados'),
    cities: new EndpointService(environment.ibgeUrl, '/estados')
  },
  disposalPoint: {
    list: new EndpointService(environment.baseUrl, '/disposal-point'),
    create: new EndpointService(environment.baseUrl, '/disposal-point/create'),
    update: new EndpointService(environment.baseUrl, '/disposal-point'),
    verify: new EndpointService(environment.baseUrl, '/disposal-point/verify'),
    delete: new EndpointService(environment.baseUrl, '/disposal-point')
  }
}
}
}
}

```

lixo-no-ponto-front\src\app\app-routing.module.ts ----

```

import {NgModule} from '@angular/core';
import {RouterModule, Routes} from '@angular/router';
import {AuthGuard} from "../shared/guard/AuthGuard";

const routes: Routes = [
  {
    path: "",
    redirectTo: "",
    pathMatch: 'full'
  },
  {
    path: 'account',
    loadChildren: () => import('./account/account.module').then(m => m.AccountModule),

```

```

    },
    {
      path: "",
      loadChildren: () => import('./home/home.module').then(m => m.HomeModule),
      canActivate: [AuthGuard]
    },
    {
      path: 'recycling-companies',
      loadChildren: () => import('./recycling-companies/recycling-companies.module').then(m =>
m.RecyclingCompaniesModule),
      canActivate: [AuthGuard]
    },
    {
      path: 'trash-collection-routes',
      loadChildren: () => import('./trash-collection-routes/trash-collection-routes.module').then(m
=> m.TrashCollectionRoutesModule),
      canActivate: [AuthGuard]
    }
  ];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule {
}

```

lixo-no-ponto-front\src\app\app.component.html ----

```

<mat-toolbar *ngIf="isLogged" class="mat-elevation-z3">
  <button (click)="collapsed.set(!collapsed())" [matTooltip]="Abrir menu lateral"
mat-icon-button>
    <mat-icon>menu</mat-icon>
  </button>
  <span>Lixo no Ponto</span>
  <span class="spacer"></span>
  <a [matMenuTriggerFor]="userMenu" class="user-menu" mat-button>
    <mat-icon [fontSet]="rla.isActive ? 'material-icons' : 'material-icons-outlined'"
[inline]="true">account_circle
    </mat-icon>
  </a>
  <mat-menu #rla=routerLinkActive #userMenu="matMenu"
routerLinkActive="selected-menu-item">
    <a [routerLink]="/account/info" mat-menu-item>
      <mat-icon [fontSet]="rla.isActive ? 'material-icons' :
'material-icons-outlined'">person</mat-icon>

```

```

    <p class="mb-0">Perfil</p>
  </a>
  <a (click)="onLogout()" mat-menu-item>
    <mat-icon [fontSet]="rla.isActive ? 'material-icons' :
'material-icons-outlined'">logout</mat-icon>
    <p class="mb-0">Log-out</p>
  </a>
</mat-menu>
</mat-toolbar>
<mat-sidenav-container>
  <mat-sidenav *ngIf="isLoggedIn" [mode]="side" [style.width]="sidenavWidth()"
class="border-end" opened>
  <app-custom-sidenav [collapsed]="collapsed()"></app-custom-sidenav>
</mat-sidenav>
  <mat-sidenav-content [ngClass]="isLoggedIn ? 'logged' : 'notLogged'"
[style.margin-left]="isLoggedIn ? sidenavWidth() : 0"
class="content">
  <router-outlet></router-outlet>
</mat-sidenav-content>
</mat-sidenav-container>
<div *ngIf="modalVersion" class="modal-version-container w-100 position-absolute
bottom-0">
  <div class="flex space-btwn center-y-row">
    <p [style.margin-bottom]="0">
      Uma nova versão está disponível. <a (click)="updateVersion()" class="bg-white"
href="">Atualize agora clicando
      aqui.</a>
    </p>
    <button (click)="closeVersion()" [style.right]="6px" aria-label="Close"
class="btn-close"
mat-icon-button
type="button">
      <mat-icon>close</mat-icon>
    </button>
  </div>
</div>
<section *ngIf="loadingService.loading$ | async" class="spinner-container">
  <mat-spinner class="spinner" diameter="50"></mat-spinner>
</section>

```

lixo-no-ponto-front\src\app\app.component.scss ----

```

mat-toolbar {
  position: relative;
  z-index: 5;
}

```

```
//.content {  
// padding: 24px;  
//}  
  
mat-sidenav-container .logged {  
  height: calc(100vh - 64px);  
  
  @media (max-width: 600px) {  
    height: calc(100vh - 56px);  
  }  
}  
  
.border-end {  
  border-radius: 0;  
  border-right: 1px solid rgba(0, 0, 0, 0.12);  
}  
  
mat-sidenav,  
mat-sidenav-content {  
  transition: all 500ms ease-in-out;  
}  
  
.spacer {  
  flex: 1 1 auto;  
}  
  
.mb-0 {  
  margin-bottom: 0;  
}  
  
.user-menu {  
  margin: 0;  
  
  mat-icon {  
    font-size: 24px;  
    height: 24px;  
    width: 24px;  
    margin-right: 0;  
  }  
}  
  
.modal-version-container {  
  padding: 6px 12px 6px 6px;  
  background-color: beige;  
  border-bottom: 1px solid #e6e6ce;  
  z-index: 1;
```

```
width: calc(100% - 18px);
}
```

```
.pwa-modal {
  background: #acb772;
  border-top: 1px solid #dfe4d7;
  padding: 8px 4px;
  position: absolute;
  bottom: 0;
}
```

```
.spinner-container {
  background-color: rgba(0, 0, 0, .2);
  position: absolute;
  width: 100vw;
  height: 100vh;
  top: 0;
  z-index: 100;
  display: flex;
  justify-content: center;
  align-items: center;
  overflow: hidden;
}
```

lixo-no-ponto-front\src\app\app.component.ts ----

```
import {Component, computed, OnDestroy, OnInit, signal} from '@angular/core';
import {AuthService} from "../shared/service/auth/auth.service";
import {Subject, Subscription, switchMap, takeUntil} from "rxjs";
import {LocationService} from "../shared/service/location/location.service";
import {MatDialog} from "@angular/material/dialog";
import {LocationDialogComponent} from "../shared/dialog/location-dialog/location-dialog.component";
import {NavigationEnd, Router} from "@angular/router";
import {LocalStorageService} from "../shared/service/local-storage/local-storage.service";
import {LocalStorageKeys} from "../shared/service/local-storage/local-storage.enum";
import {LocationReadyService} from "../shared/service/location/location-ready.service";
import {tap} from "rxjs/operators";
import {SwUpdate} from "@angular/service-worker";
import {Platform} from "@angular/cdk/platform";
import {PwaDialogComponent} from "../shared/dialog/pwa-dialog/pwa-dialog.component";
import {ToastrService} from "ngx-toastr";
import {LoadingService} from "../shared/service/loading/loading.service";

@Component({
  selector: 'app-root',
```

```

    templateUrl: './app.component.html',
    styleUrls: ['./app.component.scss']
  })
  export class AppComponent implements OnInit, OnDestroy {
    public collapsed = signal(true);
    public sidenavWidth = computed(() => this.collapsed() ? '65px' : '250px');
    public isLoggedIn = false;
    public isOnline: boolean = false;
    public modalVersion: boolean = false;
    public modalPwaPlatform: string | undefined;
    private loginSubscription: Subscription | undefined;
    private modalPwaEvent: any;
    private destroy$ = new Subject<void>();

    constructor(public loadingService: LoadingService, private toastrService: ToastrService,
    public platform: Platform, public swUpdate: SwUpdate, private locationReadyService:
    LocationReadyService, private router: Router, private authService: AuthService, private
    locationService: LocationService, private matDialog: MatDialog) {
    }

    ngOnInit() {
      LocalStorageService.errorOccurred.pipe(takeUntil(this.destroy$)).subscribe(() => {
        this.authService.logout();
      });
      this.updateOnlineStatus();

      window.addEventListener('online', this.updateOnlineStatus.bind(this));
      window.addEventListener('offline', this.updateOnlineStatus.bind(this));

      if (this.swUpdate) {
        this.swUpdate.versionUpdates.subscribe((event) => {
          if (event.type === "VERSION_DETECTED") {
            this.modalVersion = true;
          }
        });
      }

      this.loadModalPwa();

      this.loginSubscription = this.authService.isLoggedIn.subscribe(isLoggedIn => {
        this.isLoggedIn = isLoggedIn
      });

      this.router.events.subscribe((event) => {
        if (event instanceof NavigationEnd && this.isLoggedIn) {
          this.checkLocation();
        }
      })
    }
  }

```

```

}

public updateVersion(): void {
  this.swUpdate.activateUpdate()
    .then(() => {
      this.toastrService.success('Versão atualizada com sucesso.', 'Atualização')
      this.modalVersion = false;
      document.location.reload()
    })
    .catch(() => this.toastrService.warning('Erro ao atualizar a versão. Por favor atualize a
página manualmente.', 'Atualização'));
}

public closeVersion(): void {
  this.modalVersion = false;
}

ngOnDestroy() {
  this.loginSubscription?.unsubscribe();
  this.destroy$.next();
  this.destroy$.complete();
}

public onLogout(): void {
  this.authService.logout();
}

private getCurrentLocation() {
  navigator.geolocation.getCurrentPosition((position) => {
    const location = {
      lat: position.coords.latitude,
      lng: position.coords.longitude
    };
    localStorageService.setJSON(localStorageKeys.latLng, location);
    this.locationReadyService.locationReady(true);
    this.locationService.geocodeLatLng(location).subscribe(({city, state}) => {
      this.locationService.saveLocation(state, city)
    });
  }, () => {
    this.openLocationDialog();
  })
}

private openLocationDialog(): void {
  const dialogRef = this.matDialog.open(LocationDialogComponent, {disableClose: true});
  dialogRef.afterClosed().pipe(
    switchMap((result: { city: string, state: { nome: string } }) => {
      return this.locationService.geocodeAddress(`${result.city}, ${result.state.nome}`)
    })
  )
}

```

```

        .pipe(tap(() => this.locationService.saveLocation(result.state.nome, result.city))
        );
    })
    ).subscribe(() => {
        this.locationReadyService.locationReady(true)
    });
}

private checkLocation(): void {
    const state = LocalStorageService.get(LocalStorageKeys.state);

    if (!state) {
        this.getCurrentLocation();
    }
}

private updateOnlineStatus(): void {
    this.isOnline = window.navigator.onLine;
    if (!this.isOnline) {
        this.toastrService.warning('Você está offline. Verifique sua conexão com a internet.',
'Offline')
    }
}

private loadModalPwa(): void {
    if (this.platform.ANDROID) {
        window.addEventListener('beforeinstallprompt', (event: any) => {
            event.preventDefault();
            this.modalPwaEvent = event;
            this.matDialog.open(PwaDialogComponent, {disableClose: true})
                .afterClosed()
                .subscribe((clicked: boolean) => {
                    if (clicked) {
                        this.modalPwaEvent.prompt();
                    }
                })
            this.modalPwaPlatform = 'ANDROID';
        });
    }

    if (this.platform.IOS && this.platform.SAFARI) {
        /* TODO: Implementar mensagem de aviso para usuário de iPhone */
        const isInStandaloneMode = ('standalone' in window.navigator) &&
((<any>window.navigator)['standalone']);
        if (!isInStandaloneMode) {
            this.modalPwaPlatform = 'IOS';
        }
    }
}

```



```

}
}

```

lixo-no-ponto-front\src\app\app.module.ts ----

```

import {APP_INITIALIZER, LOCALE_ID, NgModule} from '@angular/core';
import {BrowserModule} from '@angular/platform-browser';

import {AppRoutingModule} from './app-routing.module';
import {AppComponent} from './app.component';
import {ServiceWorkerModule} from '@angular/service-worker';
import {provideAnimationsAsync} from '@angular/platform-browser/animations/async';
import {HTTP_INTERCEPTORS, provideHttpClient, withInterceptorsFromDi} from
"@angular/common/http";
import {JwtModule} from "@auth0/angular-jwt";
import {LocalStorageService} from './shared/service/local-storage/local-storage.service';
import {jwtInterceptor} from './shared/interceptor/jwt.interceptor';
import localePt from '@angular/common/locales/pt';
import {registerLocaleData} from "@angular/common";
import {BrowserAnimationsModule} from "@angular/platform-browser/animations";
import {MatToolbarModule} from "@angular/material/toolbar";
import {MatButtonModule} from "@angular/material/button";
import {MatIconModule} from "@angular/material/icon";
import {MatSidenavModule} from "@angular/material/sidenav";
import {CustomSidenavComponent} from './shared/component/custom-sidenav/custom-sidenav.component';
import {MatListItem, MatListItemIcon, MatListItemTitle, MatNavList} from
"@angular/material/list";
import {MatTooltip} from "@angular/material/tooltip";
import {AuthService} from './shared/service/auth/auth.service';
import {GoogleMapsModule} from "@angular/google-maps";
import {LocationDialogComponent} from './shared/dialog/location-dialog/location-dialog.component';
import {MatDialogActions, MatDialogClose, MatDialogContent, MatDialogTitle} from
"@angular/material/dialog";
import {MatFormField, MatLabel} from "@angular/material/form-field";
import {MatInput} from "@angular/material/input";
import {FormsModule} from "@angular/forms";
import {MatOption, MatSelect} from "@angular/material/select";
import {MatAutocomplete, MatAutocompleteTrigger} from "@angular/material/autocomplete";
import {ToastrModule} from "ngx-toastr";
import {ErrorInterceptor} from './shared/interceptor/error.interceptor';
import {CreateSuccessInterceptor} from './shared/interceptor/create-success.interceptor';
import {provideNgxMask} from "ngx-mask";
import {ConfirmDialogComponent} from './shared/dialog/confirm-dialog/confirm-dialog.component';

```

```
import {MatMenu, MatMenuItem, MatMenuTrigger} from "@angular/material/menu";
import {environment} from "../environments/environment";
import {PwaDialogComponent} from './shared/dialog/pwa-dialog/pwa-dialog.component';
import {MatCard, MatCardContent} from "@angular/material/card";
import {LocalStorageKeys} from "./shared/service/local-storage/local-storage.enum";
import {MatProgressSpinner} from "@angular/material/progress-spinner";
import {LoadingInterceptor} from "./shared/interceptor/loading.interceptor";
```

```
registerLocaleData(localePt);
```

```
export function tokenGetter() {
  return LocalStorageService.get(LocalStorageKeys.jwt) as string;
}
```

```
@NgModule({
  declarations: [
    AppComponent,
    CustomSidenavComponent,
    LocationDialogComponent,
    ConfirmDialogComponent,
    PwaDialogComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    GoogleMapsModule,
    JwtModule.forRoot({
      config: {
        tokenGetter: tokenGetter,
        // allowedDomains: ["example.com"],
        // disallowedRoutes: ["http://example.com/examplebadroute/"],
      },
    }),
    ServiceWorkerModule.register('ngsw-worker.js', {
      enabled: environment.production,
      registrationStrategy: 'registerWhenStable:30000'
    }),
    BrowserAnimationsModule,
    MatToolbarModule,
    MatButtonModule,
    MatIconModule,
    MatSidenavModule,
    MatNavList,
    MatListItem,
    MatListItemIcon,
    MatListItemTitle,
    MatTooltip,
    MatDialogTitle,
```

```

MatDialogContent,
MatFormField,
MatInput,
MatDialogActions,
MatDialogClose,
FormsModule,
MatLabel,
MatSelect,
MatOption,
MatAutocompleteTrigger,
MatAutocomplete,
ToastrModule.forRoot({
  preventDuplicates: true,
}),
MatMenu,
MatMenuTrigger,
MatMenuItem,
MatCardContent,
MatCard,
MatProgressSpinner,
],
providers: [
  provideAnimationsAsync(),
  provideHttpClient(withInterceptorsFromDi()),
  {
    provide: APP_INITIALIZER,
    useFactory: (authService: AuthService) => function () {
      authService.restoreDataFromLocalStorage();
    },
    deps: [AuthService],
    multi: true
  },
  {provide: HTTP_INTERCEPTORS, useClass: ErrorInterceptor, multi: true},
  {provide: HTTP_INTERCEPTORS, useValue: jwtInterceptor, multi: true},
  {provide: HTTP_INTERCEPTORS, useClass: CreateSuccessInterceptor, multi: true},
  {provide: LOCALE_ID, useValue: 'pt-BR'},
  {provide: HTTP_INTERCEPTORS, useClass: LoadingInterceptor, multi: true},
  provideNgxMask()
],
bootstrap: [AppComponent]
})
export class AppModule {
}

```

lixo-no-ponto-front\src\app\home\dialog\disposal-point-create-dialog\disposal-point-create-dialog.component.html ----

```

<section class="container">
  <h2 class="txt-center" mat-dialog-title>Cadastrar ponto de descarte</h2>
  <p class="txt-center">Esta página utiliza sua localização para a marcação do ponto de
  descarte no mapa.</p>
  <mat-dialog-content>
    <form [formGroup]="form" class="flex center-full column">
      <div [style.margin-bottom]="16px">
        <button (click)="fileInput.click()" mat-flat-button type="button">Enviar imagem do
        ponto</button>
        <input #fileInput (change)="onFileSelected($event)" hidden id="file" type="file">
        <mat-error *ngIf="form.get('foto')?.touched && (form.get('foto')?.hasError('required'))">
          O campo é <strong>obrigatório</strong>
        </mat-error>
      </div>
      <mat-form-field>
        <mat-label>Nome</mat-label>
        <input [formControlName]="nome" matInput placeholder="Nome" required>
      </mat-form-field>
      <mat-form-field>
        <mat-label>Descrição</mat-label>
        <input [formControlName]="descricao" matInput placeholder="Descrição" required>
      </mat-form-field>
      <mat-form-field>
        <mat-label>Telefone</mat-label>
        <input [formControlName]="telefone" mask="(00) 0 0000-0000|((00) 0000-0000"
        matInput placeholder="Telefone">
        <mat-error
          *ngIf="form.get('telefone')?.touched && (form.get('telefone')?.hasError('telefone') &&
          !form.get('telefone')?.hasError('required'))">
          Revise e preencha corretamente.
        </mat-error>
        <mat-error *ngIf="form.get('telefone')?.touched &&
        (form.get('telefone')?.hasError('required'))">
          O campo é <strong>obrigatório</strong>
        </mat-error>
      </mat-form-field>
      <mat-form-field>
        <mat-label>Tipos de Resíduos Aceitos</mat-label>
        <mat-select [formControlName]="residuosIds" multiple>
          <mat-select-trigger>
            {{ selectedWasteItems[0]?.nome || " " }}
            <span *ngIf="(selectedWasteItems.length || 0) > 1">
              (+{{ (selectedWasteItems.length || 0) - 1 }} {{ selectedWasteItems.length === 2 ? 'outro' :
              'outros' }})
            </span>
          </mat-select-trigger>
        </mat-form-field>

```

```

        <mat-option *ngFor="let wasteltem of wasteList" [value]="wasteltem.id">{{
wasteltem.nome }}</mat-option>
    </mat-select>
</mat-form-field>
</form>
</mat-dialog-content>
<mat-dialog-actions>
    <button (click)="onClose(false)" mat-button>Cancelar</button>
        <button (click)="onSubmit()" [disabled]="form.invalid" cdkFocusInitial
mat-button>Cadastrar</button>
</mat-dialog-actions>
</section>

```

lixo-no-ponto-front\src\app\home\dialog\disposal-point-create-dialog\disposal-point-create-dialog.component.scss ----

```

.container {
  margin: 18px;
}

```

lixo-no-ponto-front\src\app\home\dialog\disposal-point-create-dialog\disposal-point-create-dialog.component.ts ----

```

import {Component, Inject, inject, OnInit} from '@angular/core';
import {MAT_DIALOG_DATA, MatDialogRef} from "@angular/material/dialog";
import {FormControl, FormGroup, Validators} from "@angular/forms";
import {UrbanWasteModel} from "../../shared/interface/waste-list.model";
import {UrbanWasteService} from "../../shared/service/urban-waste/urban-waste.service";
import {ToastrService} from "ngx-toastr";
import {DisposalPointService} from "../../disposal-point.service";

```

```

@Component({
  selector: 'app-disposal-point-create-dialog',
  templateUrl: './disposal-point-create-dialog.component.html',
  styleUrls: ['./disposal-point-create-dialog.component.scss']
})
export class DisposalPointCreateDialogComponent implements OnInit {
  public form: FormGroup;
  readonly dialogRef = inject(MatDialogRef<DisposalPointCreateDialogComponent>);
  public wasteList: UrbanWasteModel[] | undefined;
  public selectedWasteItems: UrbanWasteModel[] = [];

```

```

  constructor(private disposalPointService: DisposalPointService,

```

```

        private toastrService: ToastrService,
        private urbanWasteService: UrbanWasteService,
        @Inject(MAT_DIALOG_DATA) public data: {
            isViaButtonClick: boolean,
            location: { lat: number, lng: number }
        }
    ){
        this.form = new FormGroup({
            'nome': new FormControl(null, [Validators.required]),
            'descricao': new FormControl(null, []),
            'foto': new FormControl(null, []),
            'telefone': new FormControl(null, []),
            'residuosIds': new FormControl(null, [Validators.required]),
            'longitude': new FormControl(null, [Validators.required]),
            'latitude': new FormControl(null, [Validators.required]),
        })

        this.urbanWasteService.getWasteList().subscribe((data) => {
            this.wasteList = data;
        });

        if (this.data.isViaButtonClick) {
            this.form.get('latitude')?.setValue(data.location.lat);
            this.form.get('longitude')?.setValue(data.location.lng);
        } else {
            navigator.geolocation.getCurrentPosition((position) => {
                this.form.get('latitude')?.setValue(position.coords.latitude);
                this.form.get('longitude')?.setValue(position.coords.longitude);
            }, () => {
                this.toastrService.info('Precisamos do acesso à localização para o cadastro do ponto de coleta.');
```

de coleta.');

```

                this.onClose(false);
            })
        }
    }

    public onClose(customValue?: boolean): void {
        this.dialogRef.close(customValue);
    }

    public onSubmit() {
        this.disposalPointService.createDisposalPoint(this.form.value).subscribe((res) => {
            this.onClose(true);
        })
    }

    ngOnInit(): void {
        this.form.get('residuosIds')?.valueChanges.subscribe(ids => {

```

```

        this.selectedWasteItems = this.wasteList?.filter(item => ids.includes(item.id)) as
UrbanWasteModel[];
    });
}

```

```

public onFileSelected(event: any): void {
    if (event.target.files && event.target.files[0]) {
        const file = event.target.files[0];
        const reader = new FileReader();
        reader.onload = (e: any) => {
            this.form.get('foto')?.setValue(e.target.result);
        };
        reader.readAsDataURL(file);
    }
}
}
}

```

lixo-no-ponto-front\src\app\home\dialog\disposal-point-info-dialog\disposal-point-info-dialog.component.html ----

```

<section class="container">
  <mat-dialog-content>
    <div *ngIf="data.foto" [style.margin-bottom]="4px">
      <img [alt]="data.nome" [src]="data.foto" [style.height]="auto"
[style.max-height]="250px"
[style.width]="100%">
    </div>
    <div class="flex space-btwn">
      <div class="flex column">
        <h1>{{ data.nome }}</h1>
        <a *ngIf="data.telefone" [href]="tel:+55' + data.telefone">
          <h2>{{ data.telefone | mask: '(00) 0 0000-0000' || '(00) 0000-0000' }}</h2>
        </a>
        <p>{{ data.descricao }}</p>
        <p>Resíduos aceitos:
          <ng-container *ngFor="let waste of data.residuos; let i = index">
            {{ waste.nome }}{{ i < data.residuos.length - 1 ? ', ' : '' }}
          </ng-container>
        </p>
        <p>(Ponto {{ data.verificado ? 'Verificado' : 'Não Verificado' }})</p>
      </div>
      <div [style.margin-left]="0.8rem">
        <button (click)="copyToClipboard()" mat-icon-button>
          <mat-icon>share</mat-icon>
        </button>
      </div>
    </div>
  </mat-dialog-content>
</section>

```

```

</div>
<div [style.gap]="1.5rem" [style.margin-bottom]="8px" class="flex column">
  <button (click)="onVerifyPoint()" *ngIf="!data.verificado" mat-stroked-button>O ponto
está correto? Verifique
  aqui.
  </button>
  <button (click)="openRouteInMaps()" mat-stroked-button>Traçar rota até o
local</button>
  <button (click)="onUpdatePoint()" *ngIf="isAdmin" mat-flat-button>Atualizar
ponto</button>
</div>
</mat-dialog-content>
<mat-dialog-actions class="flex space-btwn">
  <button (click)="onDeletePoint()" *ngIf="isAdmin" mat-button>Excluir ponto</button>
  <button (click)="onClose()" mat-button>Fechar</button>
</mat-dialog-actions>
</section>

```

lixo-no-ponto-front\src\app\home\dialog\disposal-point-info-dialog\disposal-point-info-dialog.component.scss ----

```

.container {
  margin: 18px;
}

h1 {
  font-size: 1.8rem !important;
  line-height: 1.8rem !important;
}

h2 {
  font-size: 1.3rem !important;
  line-height: 1.3rem !important;
}

```

lixo-no-ponto-front\src\app\home\dialog\disposal-point-info-dialog\disposal-point-info-dialog.component.ts ----

```

import {Component, inject} from '@angular/core';
import {MAT_DIALOG_DATA, MatDialog, MatDialogRef} from "@angular/material/dialog";
import {DisposalPointDtoModel} from "../../shared/interface/disposal-point-dto.model";
import {ToastrService} from "ngx-toastr";
import {AccountService} from "../../account/account.service";
import {DisposalPointService} from "../../disposal-point.service";

```



```

import                                     {ConfirmDialogComponent}
"../../shared/dialog/confirm-dialog/confirm-dialog.component";
import {
  DisposalPointUpdateDialogComponent
} from "../../disposal-point-update-dialog/disposal-point-update-dialog.component";

@Component({
  selector: 'app-disposal-point-info-dialog',
  templateUrl: './disposal-point-info-dialog.component.html',
  styleUrls: ['./disposal-point-info-dialog.component.scss']
})
export class DisposalPointInfoDialogComponent {
  readonly dialogRef = inject(MatDialogRef<DisposalPointInfoDialogComponent>);
  readonly data = inject<DisposalPointDtoModel>(MAT_DIALOG_DATA);
  public isAdmin = false;

  constructor(private matDialog: MatDialog, private toastrService: ToastrService, private
accountService: AccountService, private disposalPointService: DisposalPointService) {
    this.accountService.isAdmin().subscribe(isAdmin => {
      this.isAdmin = isAdmin;
    });
  }

  public onClose(reloadDisposalPoints?: boolean): void {
    this.dialogRef.close(reloadDisposalPoints);
  }

  public copyToClipboard(): void {
    const lat = this.data.latitude;
    const lng = this.data.longitude;
    const link = `https://www.google.com/maps/search/?api=1&query=${lat},${lng}`;
    navigator.clipboard.writeText(link).then(() => {
      this.toastrService.info('O link foi copiado para a área de transferência.', 'Link copiado');
    }).catch(err => {
      console.error('Could not copy text: ', err);
    });
  }

  public openRouteInMaps(): void {
    const lat = this.data.latitude;
    const lng = this.data.longitude;
    const link = `https://www.google.com/maps/dir/?api=1&destination=${lat},${lng}`;
    window.open(link, '_blank');
  }

  public onVerifyPoint(): void {
    this.disposalPointService.verifyDisposalPoint(this.data.id).subscribe(() => {
      this.toastrService.info('O ponto de descarte foi verificado.', 'Ponto verificado');
    });
  }
}

```

```

    })
  }

  public onDeletePoint(): void {
    const dialogRef = this.matDialog.open(ConfirmDialogComponent);

    dialogRef.afterClosed()
      .subscribe((confirm: boolean) => {
        if (confirm) {
          this.disposalPointService.deleteDisposalPoint(this.data.id).subscribe(() => {
            this.toastrService.info('O ponto de descarte foi removido com sucesso.', 'Ponto
removido');
            this.dialogRef.close(confirm);
          })
        }
      })
  }

  public onUpdatePoint(): void {
    const dialogRef = this.matDialog.open(DisposalPointUpdateDialogComponent, {
      data: this.data
    });

    dialogRef.afterClosed()
      .subscribe((updated: boolean) => {
        if (updated) {
          this.onClose(updated);
        }
      })
  }
}

```

lixo-no-ponto-front\src\app\home\dialog\disposal-point-update-dialog\disposal-point-update-dialog.component.html ----

```

<section class="container">
  <h2 class="txt-center" mat-dialog-title>Cadastrar ponto de descarte</h2>
  <p class="txt-center">Esta página utiliza sua localização para a marcação do ponto de
descarte no mapa.</p>
  <mat-dialog-content>
    <form [formGroup]="form" class="flex center-full column">
      <div [style.margin-bottom]="16px">
        <button (click)="fileInput.click()" mat-flat-button
          type="button">{{ form.get('foto')?.value ? 'Alterar foto do ponto de descarte' :
'Enviar imagem do ponto' }}
        </button>

```

```

<input #fileInput (change)="onFileSelected($event)" hidden id="file" type="file">
<mat-error *ngIf="form.get('foto')?.touched && (form.get('foto')?.hasError('required'))">
  O campo é <strong>obrigatório</strong>
</mat-error>
</div>
<mat-form-field>
  <mat-label>Nome</mat-label>
  <input [formControlName]="nome" matInput placeholder="Nome" required>
</mat-form-field>
<mat-form-field>
  <mat-label>Descrição</mat-label>
  <input [formControlName]="descricao" matInput placeholder="Descrição" required>
</mat-form-field>
<mat-form-field>
  <mat-label>Telefone</mat-label>
  <input [formControlName]="telefone" mask="(00) 0 0000-0000|((00) 0000-0000"
matInput placeholder="Telefone">
  <mat-error
    *ngIf="form.get('telefone')?.touched && (form.get('telefone')?.hasError('telefone') &&
!form.get('telefone')?.hasError('required'))">
    Revise e preencha corretamente.
  </mat-error>
  <mat-error *ngIf="form.get('telefone')?.touched &&
(form.get('telefone')?.hasError('required'))">
    O campo é <strong>obrigatório</strong>
  </mat-error>
</mat-form-field>
<mat-form-field>
  <mat-label>Tipos de Resíduos Aceitos</mat-label>
  <mat-select [formControlName]="residuosIds" multiple>
    <mat-select-trigger>
      {{ selectedWasteItems[0]?.nome || " }}
      <span *ngIf="(selectedWasteItems.length || 0) > 1">
        (+{{ (selectedWasteItems.length || 0) - 1 }} {{ selectedWasteItems.length === 2 ? 'outro' :
'outros' }})
      </span>
    </mat-select-trigger>
    <mat-option *ngFor="let wasteltem of wasteList" [value]="wasteltem.id">{{
wasteltem.nome }}</mat-option>
  </mat-select>
</mat-form-field>
</form>
</mat-dialog-content>
<mat-dialog-actions>
  <button mat-button mat-dialog-close>Cancelar</button>
  <button (click)="onSubmit()" cdkFocusInitial mat-button>Atualizar</button>
</mat-dialog-actions>
</section>

```

lixo-no-ponto-front\src\app\home\dialog\disposal-point-update-dialog\disposal-point-update-dialog.component.scss ----

```
.container {
  margin: 18px;
}
```

lixo-no-ponto-front\src\app\home\dialog\disposal-point-update-dialog\disposal-point-update-dialog.component.ts ----

```
import {Component, Inject, inject, OnInit} from '@angular/core';
import {FormControl, FormGroup, Validators} from "@angular/forms";
import {MAT_DIALOG_DATA, MatDialogRef} from "@angular/material/dialog";
import {UrbanWasteModel} from "../../shared/interface/waste-list.model";
import {DisposalPointService} from "../../disposal-point.service";
import {UrbanWasteService} from "../../shared/service/urban-waste/urban-waste.service";
import {DisposalPointDtoModel} from "../../shared/interface/disposal-point-dto.model";
```

```
@Component({
  selector: 'app-disposal-point-update-dialog',
  templateUrl: './disposal-point-update-dialog.component.html',
  styleUrls: ['./disposal-point-update-dialog.component.scss']
})
export class DisposalPointUpdateDialogComponent implements OnInit {
  public form: FormGroup;
  readonly dialogRef = inject(MatDialogRef<DisposalPointUpdateDialogComponent>);
  public wasteList: UrbanWasteModel[] | undefined;
  public selectedWasteItems: UrbanWasteModel[] = [];

  constructor(private disposalPointService: DisposalPointService,
               private urbanWasteService: UrbanWasteService,
               @Inject(MAT_DIALOG_DATA) private data: DisposalPointDtoModel) {
    this.urbanWasteService.getWasteList().subscribe((data) => {
      this.wasteList = data;
    });

    const residuosIds = data.residuos.map(residuo => residuo.id);
    this.selectedWasteItems = data.residuos;
    this.form = new FormGroup({
      'nome': new FormControl(data.nome, [Validators.required]),
      'descricao': new FormControl(data.descricao, []),
      'foto': new FormControl(data.foto, []),
    });
  }
}
```

```

    'telefone': new FormControl(data.telefone, [Validators.required]),
    'residuosIds': new FormControl(residuosIds, [Validators.required]),
    'longitude': new FormControl(data.longitude, [Validators.required]),
    'latitude': new FormControl(data.latitude, [Validators.required]),
  })
}

public onClose(customValue?: boolean): void {
  this.dialogRef.close(customValue);
}

public onSubmit(): void {
  this.disposalPointService.updateDisposalPoint(this.data.id, this.form.value)
    .subscribe(() => {
      this.onClose(true);
    })
}

ngOnInit(): void {
  this.form.get('residuosIds')?.valueChanges.subscribe(ids => {
    this.selectedWasteItems = this.wasteList?.filter(item => ids.includes(item.id)) as
UrbanWasteModel[];
  });
}

public onFileSelected(event: any): void {
  if (event.target.files && event.target.files[0]) {
    const file = event.target.files[0];
    const reader = new FileReader();
    reader.onload = (e: any) => {
      this.form.get('foto')?.setValue(e.target.result);
    };
    reader.readAsDataURL(file);
  }
}
}
}

```

lixo-no-ponto-front\src\app\home\dialog\filter-points-dialog\filter-points-dialog.component.htm
| ----

```

<section class="container">
  <h2 class="txt-center">Selecione os resíduos desejados:</h2>
  <mat-dialog-content>
    <div class="flex space-btwn">
      <button (click)="wastes.deselectAll()" mat-button>Limpar</button>
      <button (click)="wastes.selectAll()" mat-button>Selecionar todos</button>
    </div>
  </mat-dialog-content>
</section>

```

```

</div>
<mat-selection-list #wastes [(ngModel)]="selectedWasteItems">
  @for (wasteItem of wasteList; track wasteItem) {
    <mat-list-option togglePosition="before" [value]="wasteItem.id.toString()">{{
wasteItem.nome }}
    </mat-list-option>
  }
</mat-selection-list>
</mat-dialog-content>
<mat-dialog-actions>
  <button (click)="onClose()" mat-button>Cancelar</button>
  <button [matDialogClose]="selectedWasteItems" mat-flat-button type="submit">Pronto!
  </button>
</mat-dialog-actions>
</section>

```

lixo-no-ponto-front\src\app\home\dialog\filter-points-dialog\filter-points-dialog.component.scss
 s ----

```

.container {
  margin: 18px;
}

```

lixo-no-ponto-front\src\app\home\dialog\filter-points-dialog\filter-points-dialog.component.ts

```

import {Component, inject} from '@angular/core';
import {UrbanWasteModel} from "../../shared/interface/waste-list.model";
import {UrbanWasteService} from "../../shared/service/urban-waste/urban-waste.service";
import {MatDialogRef} from "@angular/material/dialog";

```

```

@Component({
  selector: 'app-filter-points-dialog',
  templateUrl: './filter-points-dialog.component.html',
  styleUrls: ['./filter-points-dialog.component.scss']
})
export class FilterPointsDialogComponent {
  public wasteList: UrbanWasteModel[] | undefined;
  public selectedWasteItems: number[] | undefined;
  readonly dialogRef = inject(MatDialogRef<FilterPointsDialogComponent>);

  constructor(private urbanWasteService: UrbanWasteService) {
    this.urbanWasteService.getWasteList().subscribe((data) => {

```

```

        this.wasteList = data;
    });
}

public onClose(customResult?: string[]): void {
    this.dialogRef.close(customResult);
}
}

```

lixo-no-ponto-front\src\app\home\disposal-point.service.ts ----

```

import {Injectable} from '@angular/core';
import {DisposalPointModel} from "../shared/interface/disposal-point.model";
import {Observable} from "rxjs";
import {HttpService} from "../shared/service/http/http.service";
import {API} from "../api/Endpoint";
import {DisposalPointDtoModel} from "../shared/interface/disposal-point-dto.model";

@Injectable({
    providedIn: 'root'
})
export class DisposalPointService {

    constructor(private httpService: HttpService) {
    }

    public createDisposalPoint(disposalPoint: DisposalPointModel):
    Observable<DisposalPointDtoModel> {
        return this.httpService.post<DisposalPointDtoModel>({
            url: API.endpoints.disposalPoint.create.url,
            body: disposalPoint
        });
    }

    public verifyDisposalPoint<DisposalPointDtoModel>(disposalPointId: number):
    Observable<DisposalPointDtoModel> {
        return this.httpService.post<DisposalPointDtoModel>({
            url: API.endpoints.disposalPoint.verify.url + '/' + disposalPointId,
        });
    }

    public updateDisposalPoint<DisposalPointDtoModel>(disposalPointId: number,
    disposalPoint: DisposalPointModel): Observable<DisposalPointDtoModel> {
        return this.httpService.put<DisposalPointDtoModel>({
            url: API.endpoints.disposalPoint.update.url + '/' + disposalPointId,
            body: disposalPoint
        });
    }
}

```

```
});
}
```

```
public getDisposalPointList(boundsInLatLng: {
  south: string,
  west: string,
  north: string,
  east: string
}, wastesId?: string[]): Observable<DisposalPointDtoModel[]> {
  let params: {} = {...boundsInLatLng};
  if (wastesId && wastesId.length > 0) {
    const wastesIdString = wastesId.join(',');
    params = {...params, wastesId: wastesIdString};
  }
  return this.httpService.get<DisposalPointDtoModel[]>({
    url: API.endpoints.disposalPoint.list.url,
    params
  });
}
```

```
public deleteDisposalPoint(disposalPointId: number): Observable<void> {
  return this.httpService.delete<void>({
    url: API.endpoints.disposalPoint.delete.url + '/' + disposalPointId,
  });
}
}
```

lixo-no-ponto-front\src\app\home\home-routing.module.ts ----

```
import {NgModule} from '@angular/core';
import {RouterModule, Routes} from '@angular/router';
import {HomeComponent} from './home.component';
```

```
const routes: Routes = [
  {
    path: "",
    component: HomeComponent
  }
];
```

```
@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule]
})
export class HomeRoutingModule {
}
```


lixo-no-ponto-front\src\app\home\home.component.html ----

```

<section class="map-container">
    <google-map (boundsChanged)="onMapChange()"
    (mapClick)="openDisposalPointCreateDialog(true, $event)"
    (tilesloaded)="onMapLoad()"
    *ngIf="isLocationReady"
    [height]="100%"
    [options]="options" [width]="100%">
    <map-advanced-marker
    *ngIf="this.userLocation"
    [options]="{
    content: customIconElement,
    zIndex: -1
    }"
    [position]="this.userLocation"
    [title]="Você está aqui!">
    </map-advanced-marker>
    <map-advanced-marker #markerElem="mapAdvancedMarker"
    (mapClick)="onPointClick(disposalPoint)"
    *ngFor="let disposalPoint of disposalPointList"
    [position]="{ lat: disposalPoint.latitude, lng: disposalPoint.longitude }"
    [title]="disposalPoint.nome">
    </map-advanced-marker>
    </google-map>
</section>
<button (click)="openFilterDialog()" *ngIf="showLoadButton" class="load-button"
mat-raised-button>
    Carregar pontos
</button>
<button (click)="openDisposalPointCreateDialog(false)" [matTooltip]="Cadastrar um ponto
de descarte"
    class="fab-button"
    color="secondary"
    mat-mini-fab>
    <mat-icon>add</mat-icon>
</button>

```

lixo-no-ponto-front\src\app\home\home.component.scss ----

```

.fab-button {
    position: fixed;
    right: 12px;

```

```

    bottom: 12px;
}

```

```

.map-container {
  padding: -24px;
  display: block;
  position: relative;
  height: 100%;
  width: 100%;
}

```

```

.load-button {
  position: absolute;
  bottom: 20px;
  left: 50%;
  transform: translateX(-50%);
}

```

```

:host ::ng-deep google-map iframe + div {
  border: none !important;
}

```

lixo-no-ponto-front\src\app\home\home.component.ts ----

```

import {Component, ViewChild} from '@angular/core';
import {MatDialog} from "@angular/material/dialog";
import {
  DisposalPointCreateDialogComponent
} from "../dialog/disposal-point-create-dialog/disposal-point-create-dialog.component";
import {environment} from "../../environments/environment";
import {LocalStorageService} from "../shared/service/local-storage/local-storage.service";
import {LocalStorageKeys} from "../shared/service/local-storage/local-storage.enum";
import {LocationReadyService} from "../shared/service/location/location-ready.service";
import {DisposalPointDtoModel} from "../shared/interface/disposal-point-dto.model";
import {DisposalPointService} from "../disposal-point.service";
import {GoogleMap} from "@angular/google-maps";
import
    {FilterPointsDialogComponent}
    from
    "../dialog/filter-points-dialog/filter-points-dialog.component";
import {
  DisposalPointInfoDialogComponent
} from "../dialog/disposal-point-info-dialog/disposal-point-info-dialog.component";
import {ToastrService} from "ngx-toastr";
import {Platform} from "@angular/cdk/platform";
import {AccountService} from "../account/account.service";

@Component({

```

```

    selector: 'app-home',
    templateUrl: './home.component.html',
    styleUrls: ['./home.component.scss']
  })
  export class HomeComponent {
    @ViewChild(GoogleMap) googleMap: GoogleMap | undefined;

    public options: google.maps.MapOptions = {
      mapId: environment.googleMapsMapID,
      fullscreenControl: false,
      mapTypeControl: false,
      zoomControlOptions: {
        position: 7.0
      },
      streetViewControl: false,
      zoom: 15,
      maxZoom: 18,
      minZoom: 15
    };
    public disposalPointList: DisposalPointDtoModel[] | undefined;
    public isLocationReady = false;
    public showLoadButton = false;
    public customIconElement: Node | undefined;
    public isAdmin = false;
    private firstRequest = true;
    private updateInterval = 15000;

    constructor(private accountService: AccountService, public platform: Platform, private
    toastrService: ToastrService, private disposalPointService: DisposalPointService, private
    locationReadyService: LocationReadyService, public matDialog: MatDialog) {
      this.accountService.isAdmin().subscribe(isAdmin => {
        this.isAdmin = isAdmin;
      });
      this.locationReadyService.locationReady$.subscribe((ready) => {
        if (ready) {
          this.generateUserPin();
          this.options.center = this.userLocation;
        }
        this.isLocationReady = ready;
      });

      if (navigator.geolocation && (this.platform.ANDROID || this.platform.IOS)) {
        setInterval(() => {
          this.updateUserLocation();
        }, this.updateInterval);
      }
    }
  }

```

```

public get userLocation(): { lat: number, lng: number } {
    return LocalStorageService.getJSON(LocalStorageKeys.latLng) as { lat: number, lng:
number };
}

```

```

    public openDisposalPointCreateDialog(isViaButtonClick: boolean, $event?:
google.maps.MapMouseEvent | google.maps.IconMouseEvent): void {
    if (isViaButtonClick && !this.isAdmin) {
        return;
    }
    this.updateUserLocation();
    this.matDialog.open(DisposalPointCreateDialogComponent, {
        disableClose: true,
        data: {
            isViaButtonClick,
            location: {
                lat: $event?.latLng?.lat(),
                lng: $event?.latLng?.lng()
            }
        }
    })
    .afterClosed()
    .subscribe((created) => {
        if (created) {
            this.getDisposalListByBounds();
        }
    })
}

```

```

public onPointClick(disposalPoint: DisposalPointDtoModel) {
    const dialogRef = this.matDialog.open(DisposalPointInfoDialogComponent, {
        data: disposalPoint
    })

```

```

    dialogRef.afterClosed()
        .subscribe((result) => {
            if (result) {
                this.getDisposalListByBounds();
            }
        })
}

```

```

public openFilterDialog(): void {
    this.updateUserLocation();
    const dialogRef = this.matDialog.open(FilterPointsDialogComponent);
    dialogRef.afterClosed().subscribe((result: string[]) => {
        this.onLoadDisposalPoints(result);
    })
}

```

```

}

public onLoadDisposalPoints(wastesId?: string[]) {
  if (!this.disposalPointList) {
    return;
  }
  this.getDisposalListByBounds(wastesId);
  this.showLoadButton = false;
}

public onMapLoad() {
  if (!this.disposalPointList) {
    this.getDisposalListByBounds()
  }
}

public onMapChange(): void {
  if (this.disposalPointList) {
    if (!this.showLoadButton) {
      this.showLoadButton = true;
    }
  }
}

deg2rad(deg: number) {
  return deg * (Math.PI / 180);
}

public distanceBetween(coord1: { lat: number, lng: number }, coord2: { lat: number, lng:
number }) {
  const R = 6371; // Radius of the earth in km
  const dLat = this.deg2rad(coord2.lat - coord1.lat);
  const dLng = this.deg2rad(coord2.lng - coord1.lng);
  const a = Math.sin(dLat / 2) * Math.sin(dLat / 2) +
    Math.cos(this.deg2rad(coord1.lat)) * Math.cos(this.deg2rad(coord2.lat)) *
    Math.sin(dLng / 2) * Math.sin(dLng / 2);
  const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
  const distance = R * c; // Distance in km
  return distance;
}

teste($event: google.maps.MapMouseEvent | google.maps.IconMouseEvent) {
  const position = {
    lat: $event.latLng?.lat(),
    lng: $event.latLng?.lng()
  };
  console.log('teste', position)
}

```

```

private updateUserLocation() {
  navigator.geolocation.getCurrentPosition((position) => {
    const newPosition = {
      lat: position.coords.latitude,
      lng: position.coords.longitude
    };
    localStorageService.setJSON(localStorageKeys.latLng, newPosition)

    if (this.distanceBetween(this.userLocation, newPosition) > 0.1) {
      this.googleMap?.googleMap?.setCenter(this.userLocation);
    }
  });
}

private generateUserPin() {
  const img = document.createElement('img');
  img.src = 'icons/user-location.png';
  this.customIconElement = img;
}

private getDisposalListByBounds(wastesId?: string[]) {
  const bounds = this.googleMap?.getBounds();
  if (!bounds) {
    return;
  }
  const southWest = bounds.getSouthWest();
  const northEast = bounds.getNorthEast();

  const boundsInLatLng = {
    south: southWest.lat().toString(),
    west: southWest.lng().toString(),
    north: northEast.lat().toString(),
    east: northEast.lng().toString(),
  };

  this.disposalPointService.getDisposalPointList(boundsInLatLng, wastesId)
    .subscribe((disposalPointList) => {
      this.disposalPointList = disposalPointList;
      this.showLoadButton = false;
      if (disposalPointList.length === 0 && !this.firstRequest) {
        this.toastrService.warning('Nenhum ponto de coleta encontrado para essa área.');
      }
      this.firstRequest = false;
    });
}
}
}

```

lixo-no-ponto-front\src\app\home\home.module.ts ----

```

import {NgModule} from '@angular/core';
import {CommonModule} from '@angular/common';

import {HomeRoutingModule} from './home-routing.module';
import {HomeComponent} from './home.component';
import {MatIcon} from "@angular/material/icon";
import {MatButton, MatFabButton, MatIconButton, MatMiniFabButton} from
"@angular/material/button";
import {MatTooltip} from "@angular/material/tooltip";
import {
  DisposalPointCreateDialogComponent
} from './dialog/disposal-point-create-dialog/disposal-point-create-dialog.component';
import {MatError, MatFormField, MatInput, MatLabel} from "@angular/material/input";
import {MatDialogActions, MatDialogClose, MatDialogContent, MatDialogTitle} from
"@angular/material/dialog";
import {MatOption} from "@angular/material/core";
import {MatSelect, MatSelectTrigger} from "@angular/material/select";
import {FormsModule, ReactiveFormsModule} from "@angular/forms";
import {NgxMaskDirective, NgxMaskPipe} from "ngx-mask";
import {GoogleMap, MapAdvancedMarker, MapInfoWindow} from "@angular/google-maps";
import
  {FilterPointsDialogComponent}
  from
  './dialog/filter-points-dialog/filter-points-dialog.component';
import {MatListOption, MatSelectionList} from "@angular/material/list";
import {MatCheckbox} from "@angular/material/checkbox";
import {
  DisposalPointInfoDialogComponent
} from './dialog/disposal-point-info-dialog/disposal-point-info-dialog.component';
import {MatCard, MatCardContent, MatCardHeader, MatCardSubtitle, MatCardTitle} from
"@angular/material/card";
import {
  DisposalPointUpdateDialogComponent
} from './dialog/disposal-point-update-dialog/disposal-point-update-dialog.component';

@NgModule({
  declarations: [
    HomeComponent,
    DisposalPointCreateDialogComponent,
    FilterPointsDialogComponent,
    DisposalPointInfoDialogComponent,
    DisposalPointUpdateDialogComponent
  ],
  imports: [

```

```

CommonModule,
HomeRoutingModule,
MatIcon,
MatFabButton,
MatMiniFabButton,
MatTooltip,
MatButton,
MatInput,
MatDialogTitle,
MatDialogContent,
MatFormField,
MatDialogActions,
MatDialogClose,
MatLabel,
MatOption,
MatSelect,
MatSelectTrigger,
ReactiveFormsModule,
MatError,
NgxMaskDirective,
GoogleMap,
MapAdvancedMarker,
MapInfoWindow,
MatSelectionList,
MatListOption,
FormsModule,
MatCheckbox,
MatCard,
MatCardHeader,
MatCardContent,
NgxMaskPipe,
MatCardTitle,
MatCardSubtitle,
MatIconButton,
]
})
export class HomeModule {
}

```

lixo-no-ponto-front\src\app\recycling-companies\recycling-companies-routing.module.ts ----

```

import {NgModule} from '@angular/core';
import {RouterModule, Routes} from '@angular/router';
import {RecyclingCompaniesComponent} from "../view/recycling-companies.component";
import {RecyclingCompaniesRegisterComponent} from
"../register/recycling-companies-register.component";

```



```

import {AuthGuard} from "../shared/guard/AdminGuard";

const routes: Routes = [
  {
    path: "",
    children: [
      {
        path: "",
        component: RecyclingCompaniesComponent,
      },
      {
        path: 'register',
        component: RecyclingCompaniesRegisterComponent,
        canActivate: [AuthGuard]
      }
    ]
  },
  // {
  //   path: 'register',
  //   component: RecyclingCompaniesRegisterComponent,
  // }
];

@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule]
})
export class RecyclingCompaniesRoutingModule {
}

```

lixo-no-ponto-front\src\app\recycling-companies\recycling-companies.module.ts ----

```

import {NgModule} from '@angular/core';
import {CommonModule} from '@angular/common';
import {RecyclingCompaniesRoutingModule} from "../recycling-companies-routing.module";
import {RecyclingCompaniesComponent} from "../view/recycling-companies.component";
import {MatError, MatFormField, MatLabel} from "@angular/material/form-field";
import {MatInput} from "@angular/material/input";
import {ReactiveFormsModule} from "@angular/forms";
import {RecyclingCompaniesRegisterComponent} from
"../register/recycling-companies-register.component";
import {MatButton} from "@angular/material/button";
import {NgxMaskDirective, NgxMaskPipe} from "ngx-mask";
import {MatOption, MatSelect, MatSelectTrigger} from "@angular/material/select";
import {MatCheckbox} from "@angular/material/checkbox";

```

```
import {MatCard, MatCardContent, MatCardImage, MatCardTitle} from
"@angular/material/card";
```

```
@NgModule({
  declarations: [
    RecyclingCompaniesComponent,
    RecyclingCompaniesRegisterComponent
  ],
  imports: [
    CommonModule,
    RecyclingCompaniesRoutingModule,
    MatFormField,
    MatInput,
    ReactiveFormsModule,
    MatError,
    MatLabel,
    MatButtonModule,
    NgxMaskDirective,
    MatSelect,
    MatOption,
    MatSelectTrigger,
    MatCheckbox,
    MatCard,
    MatCardImage,
    MatCardTitle,
    MatCardContent,
    NgxMaskPipe,
  ]
})
export class RecyclingCompaniesModule {
}
```

lixo-no-ponto-front\src\app\recycling-companies\recycling-companies.service.ts ----

```
import {Injectable} from '@angular/core';
import {HttpService} from "../shared/service/http/http.service";
import {RecyclingCompanyModel} from "../shared/interface/recycling-company.model";
import {API} from "../api/Endpoint";
import {Observable} from "rxjs";
import {RecyclingCompanyDtoModel} from
"../shared/interface/recycling-company-dto.model";

@Injectable({
  providedIn: 'root'
})
```

```

export class RecyclingCompaniesService {

  constructor(private httpService: HttpService) {
  }

  public getCompanies(): Observable<RecyclingCompanyDtoModel[]> {
    return this.httpService.get<RecyclingCompanyDtoModel[]>({url:
API.endpoints.recyclingCompany.list.url})
  }

  public createCompany(company: RecyclingCompanyModel):
Observable<RecyclingCompanyDtoModel> {
    return this.httpService.post<RecyclingCompanyDtoModel>({
      url: API.endpoints.recyclingCompany.create.url,
      body: company
    })
  }
}

```

lixo-no-ponto-front\src\app\recycling-companies\register\recycling-companies-register.component.html ----

```

<section class="container flex column center-full txt-center">
  <h1>Cadastrar Empresas de Reciclagem!</h1>
  <form (ngSubmit)="onSubmit()" [formGroup]="form" class="flex column w-100">
    <div [style.margin-bottom]="16px">
      <button (click)="fileInput.click()" mat-flat-button type="button">Enviar imagem da
empresa</button>
      <input #fileInput (change)="onFileSelected($event)" hidden id="file" type="file">
      <mat-error *ngIf="form.get('foto')?.touched && (form.get('foto')?.hasError('required'))">
        O campo é <strong>obrigatório</strong>
      </mat-error>
    </div>
    <mat-form-field>
      <mat-label>Nome</mat-label>
      <input [formControlName]="nome" matInput placeholder="Nome">
      <mat-error
        *ngIf="form.get('nome')?.touched && (form.get('nome')?.hasError('nome') &&
form.get('nome')?.hasError('required'))">
        Revise e preencha corretamente.
      </mat-error>
      <mat-error
        *ngIf="form.get('nome')?.touched &&
(form.get('nome')?.hasError('required'))">
        O campo é <strong>obrigatório</strong>
      </mat-error>
    </mat-form-field>

```

```

<mat-form-field>
  <mat-label>Telefone</mat-label>
  <input [formControlName]="contato" mask="(00) 0 0000-0000||(00) 0000-0000"
matInput placeholder="Telefone">
  <mat-error
    *ngIf="form.get('contato')?.touched && (form.get('contato')?.hasError('contato') &&
!form.get('contato')?.hasError('required'))">
    Revise e preencha corretamente.
  </mat-error>
  <mat-error *ngIf="form.get('contato')?.touched &&
(form.get('contato')?.hasError('required'))">
    O campo é <strong>obrigatório</strong>
  </mat-error>
</mat-form-field>
<mat-form-field>
  <mat-label>Site</mat-label>
  <input [formControlName]="site" [mask]="*" [pattern]="websiteRegex" matInput
placeholder="Site">
  <mat-error
    *ngIf="form.get('site')?.touched && (form.get('site')?.hasError('site') &&
!form.get('site')?.hasError('required'))">
    Revise e preencha corretamente.
  </mat-error>
  <mat-error *ngIf="form.get('site')?.touched && (form.get('site')?.hasError('required'))">
    O campo é <strong>obrigatório</strong>
  </mat-error>
</mat-form-field>
<mat-form-field>
  <mat-label>Tipos de Resíduos Aceitos</mat-label>
  <mat-select [formControlName]="residuosIds" multiple>
    <mat-select-trigger>
      {{ selectedWasteItems[0]?.nome || " }}
      <span *ngIf="(selectedWasteItems.length || 0) > 1">
        (+{{ (selectedWasteItems.length || 0) - 1 }} {{ selectedWasteItems.length === 2 ? 'outro' :
'outros' }})
      </span>
    </mat-select-trigger>
    <mat-option *ngFor="let wasteItem of wasteList" [value]="wasteItem.id">{{
wasteItem.nome }}</mat-option>
  </mat-select>
  <mat-error
    *ngIf="form.get('residuosIds')?.touched &&
(form.get('residuosIds')?.hasError('required'))">
    O campo é <strong>obrigatório</strong>
  </mat-error>
</mat-form-field>
<mat-form-field>
  <mat-label>Horário de Funcionamento</mat-label>

```

```

    <input [formControlName]="horarioFuncionamento" matInput placeholder="Horário de
Funcionamento">
    <mat-error
        *ngIf="form.get('horarioFuncionamento')?.touched
        (form.get('horarioFuncionamento')?.hasError('horarioFuncionamento')
        !form.get('horarioFuncionamento')?.hasError('required'))">
        Revise e preencha corretamente.
    </mat-error>
    <mat-error
        *ngIf="form.get('horarioFuncionamento')?.touched
        (form.get('horarioFuncionamento')?.hasError('required'))">
        O campo é <strong>obrigatório</strong>
    </mat-error>
</mat-form-field>
</div>
    <mat-checkbox formControlName="patrocinador">Patrocinador?</mat-checkbox>
</div>

    <button [disabled]="form.invalid" color="primary" mat-raised-button>Cadastrar</button>
</form>
</section>

```

lixo-no-ponto-front\src\app\recycling-companies\register\recycling-companies-register.component.scss ----

```

.container {
  max-width: 600px;
  margin: 18px auto;
  padding: 0 18px;
}

```

lixo-no-ponto-front\src\app\recycling-companies\register\recycling-companies-register.component.ts ----

```

import {Component, OnInit} from '@angular/core';
import {FormControl, FormGroup, Validators} from "@angular/forms";
import {UrbanWasteService} from "../../shared/service/urban-waste/urban-waste.service";
import {UrbanWasteModel} from "../../shared/interface/waste-list.model";
import {RecyclingCompaniesService} from "../../recycling-companies.service";
import {Router} from "@angular/router";

```

```

@Component({
  selector: 'app-recycling-companies-register',
  templateUrl: './recycling-companies-register.component.html',

```

```

    styleUrls: ['./recycling-companies-register.component.scss']
  })
  export class RecyclingCompaniesRegisterComponent implements OnInit {
    public form: FormGroup;
    public websiteRegex: string = '^((http|https):\\|\\|www\\.\\.)[^ "]+$';
    public wasteList: UrbanWasteModel[] | undefined;
    public selectedWasteItems: UrbanWasteModel[] = [];

    constructor(private router: Router, private urbanWasteService: UrbanWasteService, private
recyclingCompaniesService: RecyclingCompaniesService) {
      this.form = new FormGroup({
        'foto': new FormControl(null, [Validators.required]),
        'nome': new FormControl(null, [Validators.required]),
        'contato': new FormControl(null, [Validators.required]),
        'site': new FormControl(null, [Validators.required]),
        'residuosIds': new FormControl(null, [Validators.required]),
        'horarioFuncionamento': new FormControl(null, [Validators.required]),
        'patrocinador': new FormControl(false, [Validators.required]),
      });

      this.urbanWasteService.getWasteList().subscribe((data) => {
        this.wasteList = data;
      });
    }

    ngOnInit(): void {
      this.form.get('residuosIds')?.valueChanges.subscribe(ids => {
        this.selectedWasteItems = this.wasteList?.filter(item => ids.includes(item.id)) as
UrbanWasteModel[];
      });
    }

    onFileSelected(event: any) {
      if (event.target.files && event.target.files[0]) {
        const file = event.target.files[0];
        const reader = new FileReader();
        reader.onload = (e: any) => {
          this.form.get('foto')?.setValue(e.target.result);
        };
        reader.readAsDataURL(file);
      }
    }

    onSubmit() {
      if (this.form.valid) {
        this.recyclingCompaniesService.createCompany(this.form.value).subscribe((data) => {
          this.router.navigate(['/recycling-companies'])
        });
      }
    }
  }

```

```

}
}
}

```

lixo-no-ponto-front\src\app\recycling-companies\view\recycling-companies.component.html

```

<section class="container">
  <h1 class="txt-center">Empresas de Reciclagem</h1>
  <div [style.gap]="16px" class="companies-container flex wrap">
    <mat-card *ngFor="let company of recyclingCompanyList" class="company-card">
      <img [alt]="Foto da empresa ' + company.nome" [src]="company.foto" class="image"
mat-card-image>
      <mat-card-content>
        <mat-card-title>{{ company.nome }}</mat-card-title>
        <p>Telefone: <a [href]="tel:+55' + company.contato">{{ company.contato | mask: '(00) 0
0000-0000' }}</a></p>
        <p>Site: <a [href]="company.site">{{ company.site }}</a></p>
        <p>Tipos de Resíduos Aceitos:
          <span
            *ngFor="let residuo of company.tipoResiduos; let isLast=last">{{ residuo.nome }}{{
isLast ? " : ', ' }}</span>
          </p>
        <p>Horário de Funcionamento: {{ company.horarioFuncionamento }}</p>
      </mat-card-content>
    </mat-card>
  </div>
</section>

```

lixo-no-ponto-front\src\app\recycling-companies\view\recycling-companies.component.scss

```

.container {
  margin: 18px;
}

.company-card {
  width: 100%;
  position: relative;
  margin-bottom: 8px;
  @media(min-width: 600px) {
    max-width: 500px;
  }
}

```

```

p {
  word-break: break-all;
}

.image {
  width: 100%;
  max-height: 275px;
  object-fit: cover;
}
}

```

lixo-no-ponto-front\src\app\recycling-companies\view\recycling-companies.component.ts ----

```

import {Component, OnInit} from '@angular/core';
import {RecyclingCompaniesService} from "../recycling-companies.service";
import {RecyclingCompanyDtoModel} from "../shared/interface/recycling-company-dto.model";

@Component({
  selector: 'app-recycling-companies',
  templateUrl: './recycling-companies.component.html',
  styleUrls: ['./recycling-companies.component.scss']
})
export class RecyclingCompaniesComponent implements OnInit {
  public recyclingCompanyList: RecyclingCompanyDtoModel[] | undefined;

  constructor(private recyclingCompaniesService: RecyclingCompaniesService) {
  }

  ngOnInit(): void {
    this.getCompanies();
  }

  private getCompanies() {
    this.recyclingCompaniesService.getCompanies().subscribe((list) => {
      this.recyclingCompanyList = list;
    });
  }
}

```

lixo-no-ponto-front\src\app\shared\component\custom-sidenav\custom-sidenav.component.h
tml ----


```

<mat-nav-list>
  <ng-container *ngFor="let item of menuItems()">
    <a #rla=routerLinkActive *ngIf="canSeeMenuItem(item)" [activated]=rla.isActive"
[matTooltip]="item.name"
  [routerLinkActiveOptions]="{exact: true}"
  [routerLink]="item.route"
  class="menu-item"
  mat-list-item routerLinkActive="selected-menu-item">
      <mat-icon [fontSet]=rla.isActive ? 'material-icons' : 'material-icons-outlined'"
matListItemIcon>{{ item.icon }}
    </mat-icon>
    <span *ngIf="!sidenavCollapsed()" matListItemTitle>{{ item.name }}</span>
  </a>
</ng-container>
</mat-nav-list>

```

lixo-no-ponto-front\src\app\shared\component\custom-sidenav\custom-sidenav.component.s
css ----

```

:host * {
  transition: all 500ms ease-in-out;
}

.menu-item {
  border-radius: 0 !important;
  border-left: 5px solid transparent;
}

.selected-menu-item {
  border-left-color: var(--primary-color);
}

```

lixo-no-ponto-front\src\app\shared\component\custom-sidenav\custom-sidenav.component.ts

```

import {Component, Input, OnDestroy, OnInit, signal} from '@angular/core';
import {User, UserClass} from "../../interface/user.model";
import {Subscription} from "rxjs";
import {AccountService} from "../../account/account.service";

export type MenuItem = {
  name: string;
  icon: string;
  route: string;
}

```

```

    requiredRole?: UserClass;
  }

@Component({
  selector: 'app-custom-sidenav',
  templateUrl: './custom-sidenav.component.html',
  styleUrls: ['./custom-sidenav.component.scss']
})
export class CustomSidenavComponent implements OnInit, OnDestroy {
  public sidenavCollapsed = signal(false);
  public menuItems = signal<MenuItem[]>([
    {name: 'Início', icon: 'home', route: '/'},
    {name: 'Empresas de Reciclagem', icon: 'store', route: '/recycling-companies'},
    {
      name: 'Cadastro de Empresas de Reciclagem',
      icon: 'add_business',
      route: '/recycling-companies/register',
      requiredRole: UserClass.ADMIN
    },
    {name: 'Rotas de coleta', icon: 'routes', route: '/trash-collection-routes'},
    {
      name: 'Cadastro de Rotas de Coleta',
      icon: 'add_location',
      route: '/trash-collection-routes/register',
      requiredRole: UserClass.ADMIN
    },
  ]),
  public user: User | undefined;
  private userSubscription: Subscription | undefined;

  constructor(private accountService: AccountService) {
  }

  @Input() set collapsed(value: boolean) {
    this.sidenavCollapsed.set(value);
  }

  ngOnInit() {
    this.accountService.getUser().subscribe(user => {
      this.user = user;
    });
  }

  ngOnDestroy() {
    this.userSubscription?.unsubscribe();
  }

  public canSeeMenuItem(menuItem: MenuItem): boolean {

```

```

    return !menuItem.requiredRole || (this.user?.classesDeUsuario.some(classe => classe.id
=== menuItem.requiredRole) ?? false);
  }
}

```

lixo-no-ponto-front\src\app\shared\dialog\confirm-dialog\confirm-dialog.component.html ----

```

<mat-dialog-content>
  <p>
    Você tem certeza?
  </p>
</mat-dialog-content>
<mat-dialog-actions align="end">
  <button mat-dialog-close mat-raised-button tabindex="-1">Não</button>
    <button (click)="onConfirmClick()" color="primary" mat-raised-button
tabindex="1">Sim</button>
</mat-dialog-actions>

```

lixo-no-ponto-front\src\app\shared\dialog\confirm-dialog\confirm-dialog.component.scss ----

lixo-no-ponto-front\src\app\shared\dialog\confirm-dialog\confirm-dialog.component.ts ----

```

import {Component} from '@angular/core';
import {MatDialogRef} from "@angular/material/dialog";

@Component({
  selector: 'app-confirm-dialog',
  templateUrl: './confirm-dialog.component.html',
  styleUrls: ['./confirm-dialog.component.scss']
})
export class ConfirmDialogComponent {
  constructor(
    private dialogRef: MatDialogRef<ConfirmDialogComponent>) {
  }

  onConfirmClick(): void {
    this.dialogRef.close(true);
  }
}

```

lixo-no-ponto-front\src\app\shared\dialog\location-dialog\location-dialog.component.html ----

```

<section class="container">
  <h2 class="txt-center" mat-dialog-title>Escolha a cidade que você deseja buscar
informações:</h2>
  <mat-dialog-content>
    <div class="flex column center-full">
      <mat-form-field>
        <mat-label>Estado</mat-label>
        <mat-select (selectionChange)="loadCities()" [(ngModel)]="state" name="Estado"
required>
          @for (state of stateList; track state) {
            <mat-option [value]="state">{{ state.nome }}</mat-option>
          }
        </mat-select>
      </mat-form-field>
      <mat-form-field>
        <mat-label>Cidade</mat-label>
        <input
          (input)="filterCities($event)"
          [(ngModel)]="city"
          [disabled]="!cityList"
          [matAutocomplete]="auto"
          aria-label="Cidade"
          matInput
          placeholder="Cidade"
          required
          type="text">
        <mat-autocomplete #auto="matAutocomplete" requireSelection>
          @for (city of filteredCityOptions; track city) {
            <mat-option [value]="city.nome">{{ city.nome }}</mat-option>
          }
        </mat-autocomplete>
      </mat-form-field>
    </div>
  </mat-dialog-content>
  <mat-dialog-actions>
    <button [disabled]="!state" [matDialogClose]="{ city, state }" mat-button
type="submit">Pronto!</button>
  </mat-dialog-actions>
</section>

```

lixo-no-ponto-front\src\app\shared\dialog\location-dialog\location-dialog.component.scss ----

```
.container {
```

```
margin: 18px;
}
```

lixo-no-ponto-front\src\app\shared\dialog\location-dialog\location-dialog.component.ts ----

```
import {Component, OnInit} from '@angular/core';
import {LocationService} from "../../service/location/location.service";

@Component({
  selector: 'app-location-dialog',
  templateUrl: './location-dialog.component.html',
  styleUrls: ['./location-dialog.component.scss']
})
export class LocationDialogComponent implements OnInit {
  public city: { id: number, nome: string } | undefined;
  public state: { id: number, nome: string } | undefined;
  public cityList: { id: number, nome: string }[] | undefined;
  public stateList: { id: number, nome: string }[] | undefined;
  filteredCityOptions: { id: number, nome: string }[] | undefined;

  constructor(private locationService: LocationService) {
  }

  ngOnInit(): void {
    this.locationService.getStateList()
      .subscribe(states => {
        this.stateList = states;
      })
  }

  public loadCities(): void {
    if (!this.state) return;
    this.locationService.getCitiesList((this.state.id).toString())
      .subscribe(cities => {
        this.cityList = cities;
        this.filteredCityOptions = this.cityList.slice();
      })
  }

  public filterCities(event: Event): void {
    const filterValue = (event.target as HTMLInputElement).value.toLowerCase();
    this.filteredCityOptions = this.cityList?.filter(o => {
      const cityName = o.nome.toLowerCase();
      return cityName.includes(filterValue);
    });
  }
}
```

```

}
}

```

lixo-no-ponto-front\src\app\shared\dialog\pwa-dialog\pwa-dialog.component.html ----

```

<mat-dialog-content class="dialog-content">
  <div class="header-content flex space-btwn">
    <p class="dialog-title" mat-dialog-title>Instale nosso aplicativo</p>
    <button (click)="closePwa()" class="close-button" mat-dialog-close mat-icon-button
tabindex="-1">
      <mat-icon>close</mat-icon>
    </button>
  </div>
  <mat-card (click)="addToHomeScreen()" class="app-card">
    <mat-card-content class="flex w-100 flex center-y-row ">
      <div class="image-container">
        
      </div>
      <div class="content-container flex column">
        <h3>Lixo no Ponto</h3>
        <p>lixonoponto.live</p>
      </div>
    </mat-card-content>
  </mat-card>
</mat-dialog-content>

```

lixo-no-ponto-front\src\app\shared\dialog\pwa-dialog\pwa-dialog.component.scss ----

```

.dialog-content {
  position: relative;
}

.header-content {
  align-items: start;
  margin-bottom: 6px;

  .dialog-title {
    padding: 0 !important;
    margin: 0 !important;
  }
}

mat-dialog-content {
  overflow: hidden;

```

```

}

.close-button {
  position: absolute;
  top: 0;
  right: 0;
}

.app-card {
  width: 100%;
  display: flex;
  flex-direction: row;
}

.app-card mat-card-content {
  display: flex !important;
  gap: 18px;

  h3 {
    margin-bottom: 4px;
    font-size: 2rem;
    line-height: 1.8rem;
  }
}

.image-container {
  width: 30%;

  img {
    border: 1px solid #CAD787;
    border-radius: 18px;
    width: 100%;
    height: auto;
  }
}

.content-container {
  width: 70%;
}

```

lixo-no-ponto-front\src\app\shared\dialog\pwa-dialog\pwa-dialog.component.spec.ts ----

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
```

```
import { PwaDialogComponent } from './pwa-dialog.component';
```

```

describe('PwaDialogComponent', () => {
  let component: PwaDialogComponent;
  let fixture: ComponentFixture<PwaDialogComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [PwaDialogComponent]
    })
    .compileComponents();

    fixture = TestBed.createComponent(PwaDialogComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});

```

lixo-no-ponto-front\src\app\shared\dialog\pwa-dialog\pwa-dialog.component.ts ----

```

import {Component} from '@angular/core';
import {MatDialogRef} from '@angular/material/dialog';

@Component({
  selector: 'app-pwa-dialog',
  templateUrl: './pwa-dialog.component.html',
  styleUrls: ['./pwa-dialog.component.scss']
})
export class PwaDialogComponent {
  constructor(private dialogRef: MatDialogRef<PwaDialogComponent>) {
  }

  closePwa(): void {
    this.dialogRef.close(false);
  }

  addToHomeScreen(): void {
    this.dialogRef.close(true);
  }
}

```

lixo-no-ponto-front\src\app\shared\guard\AdminGuard.ts ----


```
import {Injectable} from '@angular/core';
import {ActivatedRouteSnapshot, CanActivate, Router, RouterStateSnapshot, UrlTree} from
 '@angular/router';
import {map, Observable} from 'rxjs';
import {AccountService} from "../../account/account.service";
```

```
@Injectable({
  providedIn: 'root'
})
export class AdminGuard implements CanActivate {

  constructor(private accountService: AccountService, private router: Router) {
  }

  canActivate(
    route: ActivatedRouteSnapshot,
    state: RouterStateSnapshot): Observable<boolean | UrlTree> {
    return this.accountService.isAdmin().pipe(map(isAdmin => {
      if (!isAdmin) {
        this.router.navigate(['/']);
      }
      return isAdmin;
    }));
  }
}
```

lixo-no-ponto-front\src\app\shared\guard\AlreadyLoggedGuard.ts ----

```
import {Injectable} from '@angular/core';
import {CanActivate, Router} from '@angular/router';
import {AuthService} from '../service/auth/auth.service';
import {map, Observable, take} from "rxjs";

@Injectable({
  providedIn: 'root'
})
export class AlreadyLoggedGuard implements CanActivate {

  constructor(private authService: AuthService, private router: Router) {
  }

  canActivate(): Observable<boolean> | Promise<boolean> | boolean {
    return this.authService.isLoggedIn.pipe(
      take(1),
```

```

    map(isLoggedIn => {
      if (isLoggedIn) {
        this.router.navigate(['/']);
        return false;
      }
      return true;
    })
  );
}
}

```

lixo-no-ponto-front\src\app\shared\guard\AuthGuard.ts ----

```

import {Injectable} from "@angular/core";
import {ActivatedRouteSnapshot, CanActivate, Router, RouterStateSnapshot, UrlTree} from
"@angular/router";
import {AuthService} from "../service/auth/auth.service";
import {map, Observable, take} from "rxjs";

```

```

@Injectable({
  providedIn: 'root'
})
export class AuthGuard implements CanActivate {

  constructor(private authService: AuthService, private router: Router) {
  }

  canActivate(
    route: ActivatedRouteSnapshot,
    state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean |
UrlTree> | boolean | UrlTree {
    return this.authService.isLoggedIn.pipe(
      take(1),
      map(isLoggedIn => {
        if (!isLoggedIn) {
          this.router.navigate(['/account/login']);
          return false;
        }
        return true;
      })
    )
  }
}

```

lixo-no-ponto-front\src\app\shared\interceptor\create-success.interceptor.ts ----

```
import {HttpEvent, HttpHandler, HttpInterceptor, HttpRequest, HttpResponse} from
'@angular/common/http';
import {Observable} from 'rxjs';
import {tap} from 'rxjs/operators';
import {ToastrService} from 'ngx-toastr';
import {Injectable} from "@angular/core";
```

```
@Injectable()
```

```
export class CreateSuccessInterceptor implements HttpInterceptor {
  constructor(private toastr: ToastrService) {
  }
}
```

```
  intercept(req: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {
    return next.handle(req).pipe(
      tap((event: HttpEvent<any>) => {
        if (event instanceof HttpResponse && event.status === 201 &&
req.url.endsWith('/create')) {
          this.toastr.success('Dados cadastrados com sucesso!');
        }
      })
    );
  }
}
```

lixo-no-ponto-front\src\app\shared\interceptor\error.interceptor.ts ----

```
import {HttpErrorResponse, HttpEvent, HttpHandler, HttpInterceptor, HttpRequest} from
'@angular/common/http';
import {Observable, throwError} from 'rxjs';
import {catchError} from 'rxjs/operators';
import {ToastrService} from 'ngx-toastr';
import {Injectable} from "@angular/core";
import {AuthService} from "../service/auth/auth.service";
```

```
@Injectable()
```

```
export class ErrorInterceptor implements HttpInterceptor {
  constructor(private authService: AuthService, private toastr: ToastrService) {
  }
}
```

```
  intercept(req: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {
    return next.handle(req).pipe(
```

```

catchError((error: HttpResponse) => {
  if (error.status !== 200) {
    if (error.status === 403) {
      this.toastr.error('Você não deveria estar aqui :)', 'Ops! Ocorreu um erro.');
```

```

      this.authService.logout();
    } else {
      this.toastr.error(error.error, 'Ops! Ocorreu um erro.');
```

```

    }
  }
  return throwError(() => error);
})
);
}
}

```

lixo-no-ponto-front\src\app\shared\interceptor\jwt.interceptor.ts ----

```

import {HttpEvent, HttpHandler, HttpInterceptor, HttpRequest} from '@angular/common/http';
import {Observable} from 'rxjs';
import {LocalStorageService} from "../service/local-storage/local-storage.service";
import {LocalStorageKeys} from "../service/local-storage/local-storage.enum";

```

```

export const jwtInterceptor: HttpInterceptor = {
  intercept(req: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {
    const jwtToken = LocalStorageService.get(LocalStorageKeys.jwt)

    if (jwtToken) {
      req = req.clone({
        setHeaders: {
          Authorization: `Bearer ${jwtToken}`,
          'Content-Type': 'application/json',
          'Access-Control-Allow-Origin': '*'
        }
      });
    }

    return next.handle(req);
  }
};

```

lixo-no-ponto-front\src\app\shared\interceptor\loading.interceptor.ts ----

```

import {Injectable} from '@angular/core';
import {HttpEvent, HttpHandler, HttpInterceptor, HttpRequest} from '@angular/common/http';

```

```

import {Observable} from 'rxjs';
import {finalize} from 'rxjs/operators';
import {LoadingService} from "../service/loading/loading.service";

@Injectable()
export class LoadingInterceptor implements HttpInterceptor {
  constructor(private loadingService: LoadingService) {
  }

  intercept(request: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>> {
    this.loadingService.startLoading();

    return next.handle(request).pipe(
      finalize(() => this.loadingService.stopLoading())
    );
  }
}

```

lixo-no-ponto-front\src\app\shared\interface\disposal-point-dto.model.ts ----

```

import {UrbanWasteModel} from "../waste-list.model";

export interface DisposalPointDtoModel {
  id: number;
  nome: string;
  foto?: string;
  descricao: string;
  telefone?: string;
  verificado: boolean;
  residuos: UrbanWasteModel[];
  longitude: number;
  latitude: number;
}

```

lixo-no-ponto-front\src\app\shared\interface\disposal-point.model.ts ----

```

export interface DisposalPointModel {
  nome: string;
  descricao: string;
  residuosIds: number[];
  longitude: number;
  latitude: number;
}

```

lixo-no-ponto-front\src\app\shared\interface\recycling-company-dto.model.ts ----

```
import {UrbanWasteModel} from "../waste-list.model";

export interface RecyclingCompanyDtoModel {
  id: number;
  nome: string;
  contato: string;
  site: string;
  horarioFuncionamento: string;
  patrocinador: boolean;
  foto: string;
  tipoResiduos: UrbanWasteModel[];
}
```

lixo-no-ponto-front\src\app\shared\interface\recycling-company.model.ts ----

```
export interface RecyclingCompanyModel {
  nome: string;
  contato: string;
  site: string;
  horarioFuncionamento: string;
  patrocinador: boolean;
  foto: string;
  residuosIds: number[];
}
```

lixo-no-ponto-front\src\app\shared\interface\trash-collection-routes-dto.model.ts ----

```
export interface TrashCollectionRoutesDtoModel {
  idTipoCaminhao: number;
  horarioInicio: string;
  horarioFim: string;
  cidade: string;
  idEmpresaReciclagem: number;
  ruasPassagem: string;
}
```

lixo-no-ponto-front\src\app\shared\interface\trash-collection-routes.model.ts ----

```
import {TruckTypeModel} from "../truck-type.model";
```

```
export interface TrashCollectionRoutesModel {  
  idTipoCaminhao: TruckTypeModel;  
  horarioInicio: string;  
  horarioFim: string;  
  cidade: string;  
  idEmpresaReciclagem: number;  
  ruasPassagem: string;  
}
```

```
lixo-no-ponto-front\src\app\shared\interface\truck-type.model.ts ----
```

```
export enum TruckTypeClass {  
  NORMAL = 1,  
  RECICLAVEL = 2  
}
```

```
export interface TruckTypeModel {  
  id: number;  
  nome: string;  
}
```

```
lixo-no-ponto-front\src\app\shared\interface\user-login.model.ts ----
```

```
export interface UserLoginModel {  
  email: string;  
  password: string;  
}
```

```
lixo-no-ponto-front\src\app\shared\interface\user-register.model.ts ----
```

```
export interface UserRegisterModel {  
  nome: string;  
  email: string;  
  senha: string;  
}
```

```
lixo-no-ponto-front\src\app\shared\interface\user.model.ts ----
```

```

export enum UserClass {
  ADMIN = 1,
  USER = 2
}

export interface UserClassObject {
  id: UserClass;
  name: string;
}

export interface User {
  dataCadastro: string;
  nome: string;
  email: string;
  pontosVerificados: number;
  classesDeUsuario: UserClassObject[];
}

```

lixo-no-ponto-front\src\app\shared\interface\waste-list.model.ts ----

```

export interface UrbanWasteModel {
  id: number;
  nome: string;
}

```

lixo-no-ponto-front\src\app\shared\service\auth\auth.service.ts ----

```

import {Injectable} from '@angular/core';
import {LocalStorageService} from "../../local-storage/local-storage.service";
import {LocalStorageKeys} from "../../local-storage/local-storage.enum";
import {BehaviorSubject} from "rxjs";
import {Router} from "@angular/router";

@Injectable({
  providedIn: 'root'
})
export class AuthService {
  private loggedIn = new BehaviorSubject<boolean>(this.hasToken());

  constructor(private router: Router) {
  }

  get isLoggedIn() {
    return this.loggedIn.asObservable();
  }
}

```



```

}

private get isLoggedIn(): boolean {
  return !!LocalStorageService.get(LocalStorageKeys.jwt);
}

public login(token: string): void {
  LocalStorageService.set(LocalStorageKeys.jwt, token);
  this.loggedIn.next(true);
}

public logout(): void {
  LocalStorageService.clear()
  this.loggedIn.next(false);
  this.router.navigate(['/account/login'])
}

public restoreDataFromLocalStorage(): void {
  this.validateToken();
}

private hasToken(): boolean {
  return !!LocalStorageService.get(LocalStorageKeys.jwt);
}

private validateToken() {
  try {
    const token = LocalStorageService.get(LocalStorageKeys.jwt);

    if (token) {
      this.login(token);
    } else {
      this.logout();
    }
  } catch (e) {
    this.logout();
  }
}
}

```

lixo-no-ponto-front\src\app\shared\service\cryptography\cryptography.service.ts ----

```

import {Injectable} from '@angular/core';
import {AES, enc} from 'crypto-js';
import {environment} from "../../environments/environment";

```

```

@Injectable({
  providedIn: 'root',
})
export class CryptographyService {

  private static key = environment.googleMapsMapID + environment.googleMapsApiKey;

  constructor() {
  }

  public static encrypt(message: string): string {
    return AES.encrypt(message, this.key).toString();
  }

  public static decrypt(message: string): string {
    const bytes = AES.decrypt(message, this.key);
    return bytes.toString(enc.Utf8);
  }
}

```

lixo-no-ponto-front\src\app\shared\service\http\http.service.ts ----

```

import {Injectable} from "@angular/core";
import {HttpClient, HttpHeaders, HttpParams} from "@angular/common/http";
import {Observable} from "rxjs";

interface HttpConfig {
  url: string;
  params?: {
    [param: string]: string | string[];
  };
  headers?: {
    [header: string]: string | string[];
  };
  responseType?: HttpResponseTypes;
}

interface HttpBodyConfig extends HttpConfig {
  body?: unknown;
  responseType?: HttpResponseTypes;
}

type HttpResponseTypes = 'arraybuffer' | 'blob' | 'text' | 'json';

@Injectable({
  providedIn: 'root'

```

```

}))

export class HttpService {

  constructor(
    private http: HttpClient
  ){
  }

  public get<T>({url, params, headers, responseType}: HttpConfig): Observable<T> {
    let httpParams = new HttpParams();
    if (params) {
      Object.keys(params).forEach(key => {
        const value = params[key];
        if (Array.isArray(value)) {
          value.forEach(v => httpParams = httpParams.append(key, v));
        } else {
          httpParams = httpParams.set(key, String(value));
        }
      });
    }

    const options: Record<string, HttpHeaders | HttpParams | HttpResponseTypes |
unknown> = {
      headers: new HttpHeaders({
        ...headers
      }),
      params: httpParams,
      responseType
    };

    return this.http.get<T>(url, options);
  }

  public post<T>({url, headers, body, params, responseType}: HttpBodyConfig):
Observable<T> {
    const options: Record<string, HttpHeaders | HttpParams | HttpResponseTypes |
unknown> = {
      headers: new HttpHeaders({
        'Content-Type': 'application/json; charset=UTF-8',
        ...headers
      }),
      params: new HttpParams({
        ...params
      }),
      responseType
    };
  }
}

```

```

    return this.http.post<T>(url, body, options);
}

```

```

public put<T>({url, headers, body, params}: HttpBodyConfig): Observable<T> {
    const options: Record<string, HttpHeaders | HttpParams | HttpResponseTypes |
unknown> = {
    headers: new HttpHeaders({
        'Content-Type': 'application/json; charset=UTF-8',
        ...headers
    }),
    params: new HttpParams({
        ...params
    }),
};

```

```

    return this.http.put<T>(url, body, options);
}

```

```

public patch<T>({url, headers, body, params}: HttpBodyConfig): Observable<T> {
    const options: Record<string, HttpHeaders | HttpParams | HttpResponseTypes |
unknown> = {
    headers: new HttpHeaders({
        'Content-Type': 'application/json; charset=UTF-8',
        ...headers
    }),
    params: new HttpParams({
        ...params
    }),
};

```

```

    return this.http.patch<T>(url, body, options);
}

```

```

public delete<T>({url, headers, body, params}: HttpBodyConfig): Observable<T> {
    const options: Record<string, HttpHeaders | HttpParams | HttpResponseTypes |
unknown> = {
    headers: new HttpHeaders({
        'Content-Type': 'application/json; charset=UTF-8',
        ...headers
    }),
    params: new HttpParams({
        ...params
    }),
    body,
};

```

```

    return this.http.delete<T>(url, options);
}

```

}

lixo-no-ponto-front\src\app\shared\service\jwt\jwt.service.ts ----

```
import {Injectable} from '@angular/core';
import {localStorageTypes} from "../local-storage/local-storage.type";
import {LocalStorageService} from "../local-storage/local-storage.service";
import {JwtHelperService} from "@auth0/angular-jwt";
```

```
@Injectable({
  providedIn: 'root'
})
```

```
export class JwtService {
  public jwtHelper = new JwtHelperService();
```

```
  constructor() {
  }
```

```
  public get(key: localStorageTypes): string | undefined {
    return LocalStorageService.get(key);
  }
```

```
  public set(key: localStorageTypes, token: string): void {
    LocalStorageService.set(key, token);
  }
```

```
  public decodeToken<T>(token: string): T | null {
    return this.jwtHelper.decodeToken<T>(token);
  }
```

```
  public isTokenExpired(token: string): boolean {
    return this.jwtHelper.isTokenExpired(token);
  }
```

```
  public isTokenInvalid(key: localStorageTypes): boolean {
    if (!key) {
      return true;
    }
  }
```

```
  const token = this.get(key);
```

```
  return !token ? true : this.isTokenExpired(token);
}
```

```
  public getTokenExpirationDate(token: string): Date | null {
    return this.jwtHelper.getTokenExpirationDate(token);
  }
```

```

}

public removeToken(key: localStorageTypes): void {
  LocalStorageService.remove(key);
}
}

```

lixo-no-ponto-front\src\app\shared\service\loading\loading.service.spec.ts ----

```

import { TestBed } from '@angular/core/testing';

import { LoadingService } from './loading.service';

describe('LoadingService', () => {
  let service: LoadingService;

  beforeEach(() => {
    TestBed.configureTestingModule({});
    service = TestBed.inject(LoadingService);
  });

  it('should be created', () => {
    expect(service).toBeTruthy();
  });
});

```

lixo-no-ponto-front\src\app\shared\service\loading\loading.service.ts ----

```

import { Injectable } from '@angular/core';
import { BehaviorSubject } from "rxjs";

@Injectable({
  providedIn: 'root'
})
export class LoadingService {
  public loading$ = new BehaviorSubject<boolean>(false);

  startLoading() {
    this.loading$.next(true);
  }

  stopLoading() {
    this.loading$.next(false);
  }
}

```

```
}
```

lixo-no-ponto-front\src\app\shared\service\local-storage\local-storage.enum.ts ----

```
export enum LocalStorageKeys {
  jwt = 'axdbom',
  state = 'czgyk',
  city = 'actw',
  latLng = 'afwcc'
}
```

lixo-no-ponto-front\src\app\shared\service\local-storage\local-storage.service.ts ----

```
import {Injectable} from '@angular/core';
import {LocalStorageTypes} from "../local-storage.type";
import {CryptographyService} from "../cryptography/cryptography.service";
import {Subject} from "rxjs";
```

```
@Injectable({
  providedIn: 'root',
})
```

```
export class LocalStorageService {
  public static errorOccurred = new Subject<void>();
```

```
  constructor() {
  }

```

```
  public static get(key: LocalStorageTypes): string | undefined {
    let item = localStorage.getItem(key);
    try {
      if (item) {
        item = CryptographyService.decrypt(item);
      }
    } catch (error) {
      LocalStorageService.errorOccurred.next();
      return;
    }

```

```
    return item && item !== 'undefined' ? item : undefined;
  }

```

```
  public static set(key: LocalStorageTypes, value: string): void {
    if (!value) {
      return;
    }

```

```

    }

    localStorage.setItem(key, CryptographyService.encrypt(value));
  }

  public static getJSON<T>(key: localStorageTypes): T | undefined {
    const content: string | undefined = LocalStorageService.get(key);

    return content ? JSON.parse(content) : undefined;
  }

  public static setJSON(key: localStorageTypes, value: unknown): void {
    const content: string = JSON.stringify(value);

    LocalStorageService.set(key, content);
  }

  public static remove(key: localStorageTypes): void {
    localStorage.removeItem(key);
  }

  public static clear(): void {
    localStorage.clear();
  }

  public static hasKey(key: localStorageTypes): boolean {
    return key in localStorage;
  }
}

```

lixo-no-ponto-front\src\app\shared\service\local-storage\local-storage.type.ts ----

```

export type localStorageTypes =
  | 'axdbom' | 'czgyk' | 'actw' | 'afwcç'

```

lixo-no-ponto-front\src\app\shared\service\location\location-ready.service.ts ----

```

import {Injectable} from '@angular/core';
import {BehaviorSubject} from "rxjs";
import {LocalStorageService} from "../local-storage/local-storage.service";
import {LocalStorageKeys} from "../local-storage/local-storage.enum";

```

```

@Injectable({
  providedIn: 'root'

```



```

})
export class LocationReadyService {
    private locationReadySource = new
BehaviorSubject<boolean>(LocalStorageService.hasKey(LocalStorageKeys.latLng));
    locationReady$ = this.locationReadySource.asObservable();

    locationReady(value: boolean): void {
        this.locationReadySource.next(value);
    }
}

```

lixo-no-ponto-front\src\app\shared\service\location\location.service.ts ----

```

import {Injectable} from '@angular/core';
import {HttpService} from '../http/http.service';
import {Observable, ReplaySubject} from 'rxjs';
import {MapGeocoder} from '@angular/google-maps';
import {LocalStorageService} from '../local-storage/local-storage.service';
import {LocalStorageKeys} from '../local-storage/local-storage.enum';
import {API} from '../../api/Endpoint';

@Injectable({
    providedIn: 'root'
})
export class LocationService {
    private stateList = new ReplaySubject<{ id: number, nome: string }[]>(1);
    private citiesListCache: { [key: string]: ReplaySubject<{ id: number, nome: string }[]> } = {};

    constructor(private httpService: HttpService, private mapGeocoder: MapGeocoder) {
    }

    public geocodeAddress(address: string): Observable<any> {
        return new Observable(observer => {
            this.mapGeocoder.geocode({address})
                .subscribe(response => {
                    const location = {
                        lat: response.results[0].geometry.location.lat(),
                        lng: response.results[0].geometry.location.lng()
                    };
                    LocalStorageService.setJSON(LocalStorageKeys.latLng, location);
                    observer.next(response);
                    observer.complete();
                })
        });
    }
}

```

```

public geocodeLatLng(location: google.maps.LatLngLiteral): Observable<{
  city: string | undefined,
  state: string | undefined
}> {
  return new Observable<{ city: string | undefined, state: string | undefined }>(observer => {
    this.mapGeocoder.geocode({location}).subscribe(response => {
      const results = response['results'];
      if (results[0]) {
        const addressComponents = results[0].address_components;
        const cityComponent = addressComponents.find(component =>
component.types.includes('administrative_area_level_2'));
        const stateComponent = addressComponents.find(component =>
component.types.includes('administrative_area_level_1'));

        const city = cityComponent ? cityComponent.long_name : undefined;
        const state = stateComponent ? stateComponent.long_name : undefined;

        observer.next({city, state});
        observer.complete();
      } else {
        observer.error('No results found');
      }
    }, error => {
      observer.error(error);
    });
  });
}

public saveLocation(state?: string, city?: string): void {
  if (state) {
    LocalStorageService.set(LocalStorageKeys.state, state);
  }
  if (city) {
    LocalStorageService.set(LocalStorageKeys.city, city);
  }
}

public getStateList(): Observable<{ id: number, nome: string }[]> {
  if (!this.stateList.observers.length) {
    this.httpService.get<{ id: number, nome: string }[]>({url:
API.endpoints.ibge.states.url}).subscribe(states => {
    this.stateList.next(states);
  });
}
return this.stateList.asObservable();
}

```

```

public getCitiesList(stateId: string): Observable<{ id: number, nome: string }[]> {
  if (!this.citiesListCache[stateId] || !this.citiesListCache[stateId].observers.length) {
    this.citiesListCache[stateId] = new ReplaySubject<{ id: number, nome: string }[]>(1);
    this.httpService.get<{
      id: number,
      nome: string
    }[]>({url: API.endpoints.ibge.cities.url + '/' + stateId + '/distritos'}).subscribe(cities => {
      this.citiesListCache[stateId].next(cities);
    });
  }
  return this.citiesListCache[stateId].asObservable();
}
}

```

lixo-no-ponto-front\src\app\shared\service\truck-type\truck-type.service.ts ----

```

import {Injectable} from '@angular/core';
import {Observable, ReplaySubject} from "rxjs";
import {UrbanWasteModel} from "../../interface/waste-list.model";
import {HttpService} from "../../http/http.service";
import {API} from "../../api/Endpoint";
import {TruckTypeModel} from "../../interface/truck-type.model";

@Injectable({
  providedIn: 'root'
})
export class TruckTypeService {
  private truckTypeList = new ReplaySubject<TruckTypeModel[]>(1);

  constructor(private httpService: HttpService) {
  }

  public getTruckTypeList(): Observable<TruckTypeModel[]> {
    if (!this.truckTypeList.observers.length) {
      this.httpService.get<UrbanWasteModel[]>({url: API.endpoints.truckType.list.url})
        .subscribe(typeList => {
          this.truckTypeList.next(typeList);
        })
    }
    return this.truckTypeList.asObservable();
  }
}

```

lixo-no-ponto-front\src\app\shared\service\urban-waste\urban-waste.service.ts ----

```
import {Injectable} from '@angular/core';
import {HttpService} from "../http/http.service";
import {UrbanWasteModel} from "../../interface/waste-list.model";
import {Observable, ReplaySubject} from "rxjs";
import {API} from "../../api/Endpoint";

@Injectable({
  providedIn: 'root'
})
export class UrbanWasteService {
  private wasteList = new ReplaySubject<UrbanWasteModel[]>(1);

  constructor(private httpService: HttpService) {
  }

  public getWasteList(): Observable<UrbanWasteModel[]> {
    if (!this.wasteList.observers.length) {
      this.httpService.get<UrbanWasteModel[]>({url: API.endpoints.urbanWaste.list.url})
        .subscribe(wasteList => {
          this.wasteList.next(wasteList);
        })
    }
    return this.wasteList.asObservable();
  }
}
```

lixo-no-ponto-front\src\app\trash-collection-routes\register\trash-collection-routes-register.component.html ----

```
<section class="container flex column center-full txt-center">
  <h1>Cadastrar rotas de coleta</h1>
  <form (ngSubmit)="onSubmit()" [formGroup]="form" class="w-100 flex column">
    <!-- TODO: Utilizar padrão do IBGE. Será implementado lista após a entrega. -->
    <mat-form-field>
      <mat-label>Cidade</mat-label>
      <input [formControlName]="cidade" matInput placeholder="Cidade">
      <mat-error
        *ngIf="form.get('cidade')?.touched && (form.get('cidade')?.hasError('cidade') &&
        !form.get('cidade')?.hasError('required'))">
        Revise e preencha corretamente.
      </mat-error>
      <mat-error
        *ngIf="form.get('cidade')?.touched &&
        (form.get('cidade')?.hasError('required'))">
```

```

    O campo é <strong>obrigatório</strong>
  </mat-error>
</mat-form-field>
<mat-form-field>
  <mat-label>Tipo do Caminhão</mat-label>
  <mat-select [formControlName]="idTipoCaminhao" required>
    @for (type of truckTypeList; track type) {
      <mat-option [value]="type.id">{{ type.nome }}</mat-option>
    }
  </mat-select>
  <mat-error
    *ngIf="form.get('idTipoCaminhao')?.touched
(form.get('idTipoCaminhao')?.hasError('idTipoCaminhao')
!form.get('idTipoCaminhao')?.hasError('required'))">
    Revise e preencha corretamente.
  </mat-error>
    <mat-error *ngIf="form.get('idTipoCaminhao')?.touched
(form.get('idTipoCaminhao')?.hasError('required'))">
    O campo é <strong>obrigatório</strong>
  </mat-error>
</mat-form-field>
<mat-form-field [style.margin-bottom]="16px">
  <mat-label>Ruas de Passagem</mat-label>
  <textarea [formControlName]="ruasPassagem" matInput placeholder="Ruas de
Passagem"></textarea>
  <mat-hint align="start">Separar as ruas utilizado o caracter ','</mat-hint>
  <mat-error
    *ngIf="form.get('ruasPassagem')?.touched
(form.get('ruasPassagem')?.hasError('ruasPassagem')
!form.get('ruasPassagem')?.hasError('required'))">
    Revise e preencha corretamente.
  </mat-error>
    <mat-error *ngIf="form.get('ruasPassagem')?.touched
(form.get('ruasPassagem')?.hasError('required'))">
    O campo é <strong>obrigatório</strong>
  </mat-error>
</mat-form-field>

<mat-form-field>
  <mat-label>Horário Início</mat-label>
  <input [dropSpecialCharacters]="false" [formControlName]="horarioInicio"
[mask]="00:00" [showMaskTyped]="true"
  matInput
  placeholder="Horário Início">
  <mat-hint>Formato: 00:00</mat-hint>
  <mat-error

```

```

                                *ngIf="form.get('horarioInicio')?.touched    &&
(form.get('horarioInicio')?.hasError('horarioInicio'))                &&
!form.get('horarioInicio')?.hasError('required'))">
  Revise e preencha corretamente.
</mat-error>
                                <mat-error    *ngIf="form.get('horarioInicio')?.touched    &&
(form.get('horarioInicio')?.hasError('required'))">
  O campo é <strong>obrigatório</strong>
</mat-error>
</mat-form-field>
<mat-form-field>
  <mat-label>Horário Fim</mat-label>
    <input  [dropSpecialCharacters]="false"  [formControlName]="horarioFim"
[mask]="00:00" [showMaskTyped]="true"
  matInput
  placeholder="Horário Fim">
  <mat-hint>Formato: 00:00</mat-hint>
  <mat-error
                                *ngIf="form.get('horarioFim')?.touched    &&
(form.get('horarioFim')?.hasError('horarioFim'))                    &&
!form.get('horarioFim')?.hasError('required'))">
  Revise e preencha corretamente.
  </mat-error>
                                <mat-error    *ngIf="form.get('horarioFim')?.touched    &&
(form.get('horarioFim')?.hasError('required'))">
  O campo é <strong>obrigatório</strong>
  </mat-error>
</mat-form-field>
<mat-form-field>
  <mat-label>Empresa Relacionada</mat-label>
  <mat-select [formControlName]="idEmpresaReciclagem" required>
    @for (company of recyclingCompanies; track company) {
      <mat-option [value]="company.id">{{ company.nome }}</mat-option>
    }
  </mat-select>
  <mat-error
                                *ngIf="form.get('idEmpresaReciclagem')?.touched    &&
(form.get('idEmpresaReciclagem')?.hasError('idEmpresaReciclagem'))  &&
!form.get('idEmpresaReciclagem')?.hasError('required'))">
  Revise e preencha corretamente.
  </mat-error>
  <mat-error
                                *ngIf="form.get('idEmpresaReciclagem')?.touched    &&
(form.get('idEmpresaReciclagem')?.hasError('required'))">
  O campo é <strong>obrigatório</strong>
  </mat-error>
</mat-form-field>

```

```

    <button [disabled]="form.invalid" color="primary" mat-raised-button>Cadastrar</button>
  </form>
</section>

```

lixo-no-ponto-front\src\app\trash-collection-routes\register\trash-collection-routes-register.component.scss ----

```

.container {
  max-width: 600px;
  margin: 18px auto;
  padding: 0 18px;
}

```

lixo-no-ponto-front\src\app\trash-collection-routes\register\trash-collection-routes-register.component.ts ----

```

import {Component, OnInit} from '@angular/core';
import {FormControl, FormGroup, Validators} from "@angular/forms";
import {TrashCollectionRoutesService} from "../trash-collection-routes.service";
import {TruckTypeModel} from "../../shared/interface/truck-type.model";
import {TruckTypeService} from "../../shared/service/truck-type/truck-type.service";
import
    {RecyclingCompaniesService}
    from
    "../../recycling-companies/recycling-companies.service";
import
    {RecyclingCompanyDtoModel}
    from
    "../../shared/interface/recycling-company-dto.model";
import {Router} from "@angular/router";

@Component({
  selector: 'app-trash-collection-routes-register',
  templateUrl: './trash-collection-routes-register.component.html',
  styleUrls: ['./trash-collection-routes-register.component.scss']
})
export class TrashCollectionRoutesRegisterComponent implements OnInit {
  public form: FormGroup;
  public truckTypeList: TruckTypeModel[] | undefined;
  public recyclingCompanies: RecyclingCompanyDtoModel[] | undefined;

  /* TODO: Na refatoração pegar api de cidades e pegar o estado com base na localização */
  constructor(private router: Router, private recyclingCompaniesService:
    RecyclingCompaniesService, private trashCollectionRoutesService:
    TrashCollectionRoutesService, private vehicleTypeService: TruckTypeService) {
    this.form = new FormGroup({
      'idTipoCaminhao': new FormControl(null, [Validators.required]),

```

```

    'horarioInicio': new FormControl(null, [Validators.required]),
    'horarioFim': new FormControl(null, [Validators.required]),
    'cidade': new FormControl(null, [Validators.required]),
    'idEmpresaReciclagem': new FormControl(null, [Validators.required]),
    'ruasPassagem': new FormControl(null, [Validators.required]),
  });

  this.vehicleTypeService.getTruckTypeList().subscribe((data) => {
    this.truckTypeList = data;
  });

  this.recyclingCompaniesService.getCompanies().subscribe((data) => {
    this.recyclingCompanies = data;
  });
}

ngOnInit(): void {
}

onSubmit() {
  if (this.form.valid) {
    const formValues = this.form.value;
    const transformTime = (time: string) => {
      const [hours, minutes] = time.split(':');
      return new Date(1970, 0, 1, Number(hours), Number(minutes));
    };

    formValues.horarioInicio = transformTime(formValues.horarioInicio);
    formValues.horarioFim = transformTime(formValues.horarioFim);
    this.trashCollectionRoutesService.createRoute(this.form.value).subscribe((data) => {
      this.router.navigate(['/trash-collection-routes'])
    });
  }
}
}

```

lixo-no-ponto-front\src\app\trash-collection-routes\trash-collection-routes-routing.module.ts

```

import {NgModule} from '@angular/core';
import {RouterModule, Routes} from '@angular/router';
import {AuthGuard} from "../../shared/guard/AdminGuard";
import {TrashCollectionRoutesComponent} from "../../view/trash-collection-routes.component";

```



```
import {TrashCollectionRoutesRegisterComponent} from
"./register/trash-collection-routes-register.component";
```

```
const routes: Routes = [
  {
    path: "",
    children: [
      {
        path: "",
        component: TrashCollectionRoutesComponent,
      },
      {
        path: 'register',
        component: TrashCollectionRoutesRegisterComponent,
        canActivate: [AuthGuard]
      }
    ]
  }
];
```

```
@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule]
})
export class TrashCollectionRoutesRoutingModule {
}
```

lixo-no-ponto-front\src\app\trash-collection-routes\trash-collection-routes.module.ts ----

```
import {NgModule} from '@angular/core';
import {CommonModule} from '@angular/common';
import {TrashCollectionRoutesRoutingModule} from
"./trash-collection-routes-routing.module";
import {TrashCollectionRoutesComponent} from './view/trash-collection-routes.component';
import {TrashCollectionRoutesRegisterComponent} from
"./register/trash-collection-routes-register.component";
import {FormsModule, ReactiveFormsModule} from "@angular/forms";
import {MatError, MatFormField, MatHint, MatLabel} from "@angular/material/form-field";
import {MatInput} from "@angular/material/input";
import {MatButton} from "@angular/material/button";
import {MatOption} from "@angular/material/core";
import {MatSelect} from "@angular/material/select";
import {NgxMaskDirective} from "ngx-mask";
import {MatCard, MatCardContent, MatCardTitle} from "@angular/material/card";
```

```

@NgModule({
  declarations: [
    TrashCollectionRoutesComponent,
    TrashCollectionRoutesRegisterComponent
  ],
  imports: [
    CommonModule,
    TrashCollectionRoutesRoutingModule,
    FormsModule,
    ReactiveFormsModule,
    MatError,
    MatFormField,
    MatInput,
    MatLabel,
    MatButtonModule,
    MatOption,
    MatSelect,
    MatHint,
    NgxMaskDirective,
    MatCard,
    MatCardTitle,
    MatCardContent
  ]
})
export class TrashCollectionRoutesModule {
}

```

lixo-no-ponto-front\src\app\trash-collection-routes\trash-collection-routes.service.ts ----

```

import {Injectable} from '@angular/core';
import {TrashCollectionRoutesModel} from './shared/interface/trash-collection-routes.model';
import {Observable} from "rxjs";
import {API} from "../api/Endpoint";
import {HttpService} from "../shared/service/http/http.service";

@Injectable({
  providedIn: 'root'
})
export class TrashCollectionRoutesService {

  constructor(private httpService: HttpService) {

  }

  public getTrashCollectionRoutes(city: string, truckType?: string):
  Observable<TrashCollectionRoutesModel[]> {

```

```

let params: { [key: string]: string } = {
  cidade: city
};
if (truckType !== undefined) {
  params['idTipoCaminhao'] = truckType;
}
return this.httpService.get<TrashCollectionRoutesModel[]>({
  url: API.endpoints.trashCollectionRoutes.list.url,
  params
})
}

```

```

        public      createRoute(route:      TrashCollectionRoutesModel):
Observable<TrashCollectionRoutesModel> {
  return this.httpService.post<TrashCollectionRoutesModel>({
    url: API.endpoints.trashCollectionRoutes.create.url,
    body: route
  })
}
}

```

lixo-no-ponto-front\src\app\trash-collection-routes\view\trash-collection-routes.component.html ----

```

<section class="container">
  <h1 class="txt-center">Rotas e Horários de Coleta de Lixo</h1>
  <div class="filter-container flex space-btwn">
    <mat-form-field class="filter-field-name">
      <mat-label>Filtrar por nome da rua</mat-label>
      <input [(ngModel)]="filterRua" matInput>
    </mat-form-field>

    <mat-form-field class="filter-field-type">
      <mat-label>Tipo de coleta</mat-label>
      <mat-select [(ngModel)]="filterTipoColeta">
        <mat-option value="1">Coleta Comum</mat-option>
        <mat-option value="2">Coleta Seletiva</mat-option>
      </mat-select>
    </mat-form-field>
  </div>

  <div class="routes-container">
    <ng-container *ngFor="let route of filteredRoutes">
      <mat-card *ngFor="let rua of route.ruasPassagem.split(';')" [style.margin-bottom]="16px"
class="route-card">
        <div class="route-info flex">

```

```

    <div class="flex route-type center-full">
      <span [innerHTML]="route.idTipoCaminhao.id === 1 ? 'Coleta<br>Comum' :
'Coleta<br>Seletiva'"
        [ngClass]="route.idTipoCaminhao.id === 1 ? 'comum' : 'seletiva'"
        class="txt-center route-type-circle"></span>
    </div>
    <div class="route-details">
      <mat-card-content>
        <mat-card-title>{{ rua }}</mat-card-title>
        <p>Horário da Rota: {{ route.horarioInicio | date: 'HH:mm' }} - {{ route.horarioFim |
date: 'HH:mm' }}</p>
      </mat-card-content>
    </div>
  </div>
</mat-card>
</ng-container>
<ng-container *ngIf="isEmpty">
  <p>Nenhuma rua encontrada com os critérios de busca fornecidos.</p>
</ng-container>
</div>
</section>

```

lixo-no-ponto-front\src\app\trash-collection-routes\view\trash-collection-routes.component.sc
ss ----

```

.container {
  margin: 18px;
}

.route-info {
  @media (max-width: 600px) {
    flex-direction: column;
  }
}

.route-type {
  width: 10%;
  @media (max-width: 600px) {
    width: 100%;
  }
}

.route-details {
  width: 90%;
  @media (max-width: 600px) {
    width: 100%;
  }
}

```

```
    text-align: center;
  }
}

.route-type-circle {
  border-radius: 50%;
  padding: 14px 8px;
  color: white;

  &.comum {
    background-color: #CAD787;
  }

  &.seletiva {
    background-color: #026e00;
  }
}

.filter-container {
  @media (max-width: 600px) {
    flex-wrap: wrap;
  }
}

.filter-field-name {
  width: 75%;
  margin-right: 8px;
  @media (max-width: 600px) {
    width: 100%;
    margin-right: 0;
  }
}

.filter-field-type {
  width: 25%;
  margin-left: 8px;
  @media (max-width: 600px) {
    width: 100%;
    margin-left: 0;
  }
}
}
```

```

import {Component, OnInit} from '@angular/core';
import {TrashCollectionRoutesModel} from
"../../shared/interface/trash-collection-routes.model";
import {TrashCollectionRoutesService} from "../../trash-collection-routes.service";
import {LocalStorageService} from "../../shared/service/local-storage/local-storage.service";
import {LocalStorageKeys} from "../../shared/service/local-storage/local-storage.enum";

@Component({
  selector: 'app-trash-collection-routes',
  templateUrl: './trash-collection-routes.component.html',
  styleUrls: ['./trash-collection-routes.component.scss']
})
export class TrashCollectionRoutesComponent implements OnInit {
  public trashCollectionRoutesList: TrashCollectionRoutesModel[] | undefined;
  public filterRua: string = "";
  public filterTipoColeta: number | undefined;
  public isEmpty = false;

  constructor(private trashCollectionRoutesService: TrashCollectionRoutesService) {
  }

  get filteredRoutes(): TrashCollectionRoutesModel[] | [] {
    if (!this.trashCollectionRoutesList) {
      return [];
    }

    this.isEmpty = false;

    const filteredRoutes = this.trashCollectionRoutesList.flatMap(route => {
      return route.ruasPassagem.split(';')
        .map(rua => rua.trim())
        .filter(rua => rua !== "")
        .map(rua => ({...route, ruasPassagem: rua}));
    }).filter(route => {
      const ruaPassesFilter = this.filterRua ?
route.ruasPassagem.toLowerCase().includes(this.filterRua.toLowerCase()) : true;
      const tipoColetaPassesFilter = this.filterTipoColeta ? route.idTipoCaminhao.id ===
Number(this.filterTipoColeta) : true;
      return ruaPassesFilter && tipoColetaPassesFilter;
    });

    if (filteredRoutes.length === 0) {
      this.isEmpty = true;
    }

    return filteredRoutes;
  }
}

```

```

ngOnInit() {
  /* TODO: Pegar localização via gps ou fazer modal para selecionar estado e cidade. */
  this.getTrashCollectionRoutes(localStorageService.get(localStorageKeys.city) as string);
}

/* TODO: Faltar por o model DTO, porque o retorno do back-end é diferente.*/
private getTrashCollectionRoutes(city: string, truckType?: string) {
  this.trashCollectionRoutesService.getTrashCollectionRoutes(city,
truckType).subscribe((list) => {
  this.trashCollectionRoutesList = list;
  });
}
}
}

```

lixo-no-ponto-front\src\environments\environment.prod.ts ----

```

export const environment = {
  production: true,
  googleMapsApiKey: 'AlzaSyC4ELJbjNTTtwgXmLjjEqQ8RzHv5E2pTWRg',
  googleMapsMapID: 'ef54da838b6f0cba',
  baseUrl: 'https://lixo-no-ponto-5de457800c7a.herokuapp.com/api',
  ibgeUrl: 'https://servicodados.ibge.gov.br/api/v1/localidades',
};

```

lixo-no-ponto-front\src\environments\environment.ts ----

```

export const environment = {
  production: true,
  googleMapsApiKey: 'AlzaSyC4ELJbjNTTtwgXmLjjEqQ8RzHv5E2pTWRg',
  googleMapsMapID: 'ef54da838b6f0cba',
  baseUrl: 'https://lixo-no-ponto-5de457800c7a.herokuapp.com/api',
  ibgeUrl: 'https://servicodados.ibge.gov.br/api/v1/localidades',
};

```

lixo-no-ponto-front\src\index.html ----

```

<!doctype html>
<html lang="pt-BR">
<head>
  <meta charset="utf-8">
  <title>Lixo no Ponto</title>

```

```

<base href="/">
<meta content="width=device-width, initial-scale=1" name="viewport">
  <meta content="Lixo no Ponto é um aplicativo que ajuda você a encontrar pontos de
descarte de resíduos na sua cidade."
  name="description">
<meta content="lixo, ponto, descarte, cidade, aplicativo, resíduos" name="keywords">
<link href="icons/apple-icon-57x57.png" rel="apple-touch-icon" sizes="57x57">
<link href="icons/apple-icon-60x60.png" rel="apple-touch-icon" sizes="60x60">
<link href="icons/apple-icon-72x72.png" rel="apple-touch-icon" sizes="72x72">
<link href="icons/apple-icon-76x76.png" rel="apple-touch-icon" sizes="76x76">
<link href="icons/apple-icon-114x114.png" rel="apple-touch-icon" sizes="114x114">
<link href="icons/apple-icon-120x120.png" rel="apple-touch-icon" sizes="120x120">
<link href="icons/apple-icon-144x144.png" rel="apple-touch-icon" sizes="144x144">
<link href="icons/apple-icon-152x152.png" rel="apple-touch-icon" sizes="152x152">
<link href="icons/apple-icon-180x180.png" rel="apple-touch-icon" sizes="180x180">
  <link href="icons/android-icon-192x192.png" rel="icon" sizes="192x192"
type="image/png">
<link href="icons/favicon-32x32.png" rel="icon" sizes="32x32" type="image/png">
<link href="icons/favicon-96x96.png" rel="icon" sizes="96x96" type="image/png">
<link href="icons/favicon-16x16.png" rel="icon" sizes="16x16" type="image/png">
<link href="favicon.ico" rel="icon" type="image/x-icon">
<link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@300;400;500&amp;display=s
wap" rel="stylesheet">
<link
href="https://fonts.googleapis.com/icon?family=Material+Icons|Material+Icons+Outlined"
rel="stylesheet">
<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
<meta content="#CAD787" name="theme-color">
<link href="manifest.webmanifest" rel="manifest">
</head>
<body class="mat-typography">
<app-root></app-root>
<noscript>Please enable JavaScript to continue using this application.</noscript>
<script>
(g => {
  var h, a, k, p = "The Google Maps JavaScript API", c = "google", l = "importLibrary", q =
"__ib__", m = document,
  b = window;
  b = b[c] || (b[c] = {});
  var d = b.maps || (b.maps = {}), r = new Set, e = new URLSearchParams,
  u = () => h || (h = new Promise(async (f, n) => {
    await (a = m.createElement("script"));
    e.set("libraries", [...r] + "");
    for (k in g) e.set(k.replace(/[A-Z]/g, t => "_" + t[0].toLowerCase()), g[k]);
    e.set("callback", c + ".maps." + q);
    a.src = `https://maps.${c}apis.com/maps/api/js?` + e;
    d[q] = f;

```



```

    a.onerror = () => h = n(Error(p + " could not load."));
    a.nonce = m.querySelector("script[nonce]")?.nonce || "";
    m.head.append(a)
  });
  d[] ? console.warn(p + " only loads once. Ignoring:", g) : d[] = (f, ...n) => r.add(f) &&
u().then(() => d[](f, ...n))
}({
  key: "AlzaSyC4ELJbjNTTtwgXmLjjEqQ8RzHv5E2pTWRg",
  v: "weekly",
});
</script>
</body>
</html>

```

lixo-no-ponto-front\src\main.ts ----

```

import {platformBrowserDynamic} from '@angular/platform-browser-dynamic';

import {AppModule} from './app/app.module';
import {API} from './app/api/Endpoint';

API.init()

platformBrowserDynamic().bootstrapModule(AppModule, {
  ngZoneEventCoalescing: true
})
.catch(err => console.error(err));

```

lixo-no-ponto-front\src\styles.scss ----

```

@use '@angular/material' as mat;

@include mat.core();

$lixo-no-ponto-front-theme: mat.define-theme((
  color: (
    theme-type: light,
    primary: mat.$green-palette,
    tertiary: mat.$blue-palette,
  ),
  density: (
    scale: 0,
  )
));

```

```
:root {
  @include mat.all-component-themes($lixo-no-ponto-front-theme);
}

html, body {
  height: 100%;
}

body {
  margin: 0;
  font-family: Roboto, "Helvetica Neue", sans-serif;
  --primary-color: #a4bd26
}

/*
Framework v1.1
Data de atualização: 29/06/2021
Autor: André J. Bez Birolo
*/

/* Gerais */
.margin-auto {
  margin: 0 auto;
}

.transition {
  -webkit-transition: all .2s linear;
  transition: all .2s linear;
}

/* Disposição dos itens */
.flex {
  display: -webkit-box;
  display: -ms-flexbox;
  display: flex;
}

.column {
  -webkit-box-orient: vertical;
  -webkit-box-direction: normal;
  -ms-flex-direction: column;
  flex-direction: column;
}

.flex-auto-3 {
```

```
flex: 1 0 290px;  
}
```

```
.flex-auto-4 {  
flex: 1 0 250px;  
}
```

```
.flex-auto-5 {  
flex: 1 0 200px;  
}
```

```
.wrap {  
-ms-flex-wrap: wrap;  
flex-wrap: wrap;  
}
```

```
.space-btwn {  
-webkit-box-pack: justify;  
-ms-flex-pack: justify;  
justify-content: space-between !important;  
}
```

```
.center-full {  
-webkit-box-pack: center;  
-ms-flex-pack: center;  
justify-content: center;  
-webkit-box-align: center;  
-ms-flex-align: center;  
align-items: center;  
}
```

```
.center-x-row {  
-webkit-box-pack: center;  
-ms-flex-pack: center;  
justify-content: center;  
}
```

```
.center-y-row,  
.center-x-column {  
-webkit-box-align: center;  
-ms-flex-align: center;  
align-items: center;  
}
```

```
.justify-start {  
-webkit-box-pack: start;  
-ms-flex-pack: start;  
justify-content: start;
```

```
}

/* Textos */
.txt-center {
  text-align: center;
}

.font-light {
  font-weight: 300;
}

.font-bold {
  font-weight: bold;
}

.font-italic {
  font-style: italic;
}

.font-uppercase {
  text-transform: uppercase;
}

.font-capitalize {
  text-transform: capitalize;
}

/* Responsivo */
@media (min-width: 1023px) {
  .w-50-t {
    width: 50%;
  }
}

@media (min-width: 767px) {
  .w-50-m {
    width: 50%;
  }
}

@media (max-width: 1024px) {
  .hide-from-tablet {
    display: none;
  }
}

@media (max-width: 768px) {
  .hide-from-phone {
```

```
    display: none;
  }
}

.w-100 {
  width: 100%;
}

.position-absolute {
  position: absolute;
}

.position-relative {
  position: relative;
}

.top-0 {
  top: 0;
}

.bottom-0 {
  bottom: 0;
}

.end-0 {
  right: 0;
}

html, body {
  height: 100%;
}

body {
  margin: 0;
  font-family: Roboto, "Helvetica Neue", sans-serif;
}

h1 {
  font-size: 2.5rem !important;
  line-height: 2.5rem !important;
}

h2 {
  font-size: 2rem !important;
}

a.bg-white {
```

```

color: #026e00;
}

```

lixo-no-ponto-front\TCC (respostas) - Respostas ao formulário 1.csv ----

Carimbo de data/hora,Endereço de e-mail,Agradecemos sua participação. Você concorda em participar dessa pesquisa?,Qual seu nome?,Qual sua idade? (Apenas números),Qual seu sexo?,Qual plataforma você pretende utilizar para testar nosso projeto?,Você tem familiaridade com o uso de aplicativos em dispositivos móveis e/ou computador/notebook?,Quantos pontos de descarte você conhecia antes de usar o sistema?,Quantos pontos de descarte novos você conseguiu identificar após usar o sistema? ,Quantas rotas de coleta de lixo você conhecia antes de usar o sistema?,Quantas rotas de coleta de lixo novas você conseguiu identificar após usar o sistema?,As informações sobre os pontos de descarte são de fácil compreensão.,As informações sobre as rotas de coleta de lixo são de fácil compreensão,"As informações sobre os pontos de descarte são adequadas (suficientes, claras e corretas).","As informações sobre as rotas de coleta de lixo são adequadas (suficientes, claras e corretas)",Eu acho que gostaria de usar esse sistema com frequência.,Eu acho o sistema desnecessariamente complexo.,Eu achei o sistema fácil de usar.,Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.,Eu acho que as várias funções do sistema estão muito bem integradas.,Eu acho que o sistema apresenta muita inconsistência.,Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.,Eu achei o sistema atrapalhado de usar.,Eu me senti confiante ao usar o sistema.,Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

30/09/2024 08:57:40,pedro_silva@gmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Pedro,29,Masculino,Computador/Notebook,Muita,2,4,1,3,Sim,Sim,5,5,3,2,5,2,5,1,4,1,4,1

30/09/2024 15:49:42,victorm.araujo@hotmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Victor Moura de Araújo,25,Masculino,Computador/Notebook,Muita,2,1,0,9,Sim,Sim,4,4,2,1,4,1,2,2,5,1,5,1

01/10/2024 13:00:17,pedrorhamon16@gmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Pedro Rhamon,26,Masculino,Computador/Notebook,Muita,0,3,1,2,Sim,Sim,5,5,4,2,5,1,5,2,5,1,5,3

01/10/2024 19:20:53,rique.soaresc@gmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Henrique,23,Masculino,Computador/Notebook,Muita,2,2,2,2,Sim,Sim,4,3,2,1,4,2,5,1,5,2,5,1

02/10/2024 22:25:45,francieleeesamu@gmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Franciele Rodrigues de Queiros,28,Feminino,Dispositivos Móveis,Mediana,2,3,2,1,Sim,Sim,5,4,5,4,5,2,3,2,5,1,4,1

02/10/2024 22:32:10,anne.fbittencourt@gmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Anne Freitas Bittencourt,21,Feminino,Dispositivos Móveis,Mediana,0,1,0,2,Sim,Sim,5,5,4,1,4,1,5,1,5,1,5,1

03/10/2024 00:02:08,marcocesardasilva@gmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Marco Cesar da Silva,38,Masculino,Dispositivos Móveis,Muita,2,7,1,10,Sim,Sim,5,5,3,1,5,1,5,1,5,1,4,1

03/10/2024 00:18:30,mmkantikas@gmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Mayane,32,Feminino,Dispositivos Móveis,Muita,1,2,2,2,Sim,Sim,4,4,5,1,4,4,4,2,4,1,5,4

03/10/2024 05:35:59,rodrigoquerodecastro@hotmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Rodrigo Quero de Castro,26,Masculino,Dispositivos Móveis,Mediana,0,2,1,5,Sim,Sim,5,5,4,1,5,1,5,1,5,1,5,1

03/10/2024 08:12:36,jardelgalvao1@gmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Jardel Galvão Rodrigues,31,Masculino,Computador/Notebook,Muita,0,6,0,2,Sim,Sim,5,5,5,1,4,1,5,1,5,1,5,1

03/10/2024 17:21:57,vihvw1802@hotmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Vitória Valio Weiss,21,Feminino,Dispositivos Móveis,Muita,0,1,0,2,Sim,Sim,5,3,3,1,5,1,4,1,5,2,4,1

04/10/2024 10:54:18,jpzmartinez@outlook.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",João Pedro Ziliotto Martinez,23,Masculino,Dispositivos Móveis,Muita,0,0,0,9,Sim,Sim,4,5,5,1,5,1,5,1,5,1,5,1

04/10/2024 17:52:17,cleantoneto2010@hotmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",José Cleanto Feitosa Arrais Neto,24,Masculino,Computador/Notebook,Muita,0,6,0,6,Sim,Sim,4,5,3,1,5,1,1,1,5,1,5,1

06/10/2024 11:49:41,beatriz.valio@gmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Beatriz Valio Weiss,25,Feminino,Dispositivos Móveis,Muita,0,4,0,9,Sim,Sim,5,5,4,1,5,1,5,1,5,1,5,1

06/10/2024 12:28:08,estevam.ufsc@gmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Cristian Nunes Estevam,31,Masculino,Dispositivos Móveis,Muita,1,3,0,3,Não,Sim,4,3,5,1,4,1,3,2,4,2,5,1

06/10/2024 16:16:16,cahfvale@gmail.com,"Sim, li o Termo de Consentimento Livre e Esclarecido (TCLE) e aceito participar da avaliação.",Carolina Nau França Vale,22,Feminino,Dispositivos Móveis,Muita,0,2,0,9,Sim,Não,4,2,2,1,4,1,3,4,4,2,4,1

lixo-no-ponto-front\ttsconfig.app.json ----

```

/* To learn more about Typescript configuration file:
https://www.typescriptlang.org/docs/handbook/tsconfig-json.html. */
/* To learn more about Angular compiler options:
https://angular.dev/reference/configs/angular-compiler-options. */
{
  "extends": "./tsconfig.json",
  "compilerOptions": {
    "outDir": "./out-tsc/app",
    "typeRoots": [
      "node_modules/@types"
    ],
    "types": [
      "googlemaps"
    ]
  }
}

```

```

    ]
  },
  "files": [
    "src/main.ts"
  ],
  "include": [
    "src/**/*.d.ts"
  ]
}

```

lixo-no-ponto-front\tscconfig.json ----

```

/*      To      learn      more      about      Typescript      configuration      file:
https://www.typescriptlang.org/docs/handbook/tsconfig-json.html. */
/*      To      learn      more      about      Angular      compiler      options:
https://angular.dev/reference/configs/angular-compiler-options. */
{
  "compileOnSave": false,
  "compilerOptions": {
    "outDir": "./dist/out-tsc",
    "strict": true,
    "noImplicitOverride": true,
    "noPropertyAccessFromIndexSignature": true,
    "noImplicitReturns": true,
    "noFallthroughCasesInSwitch": true,
    "skipLibCheck": true,
    "esModuleInterop": true,
    "sourceMap": true,
    "declaration": false,
    "experimentalDecorators": true,
    "moduleResolution": "bundler",
    "importHelpers": true,
    "target": "ES2022",
    "module": "ES2022",
    "lib": [
      "ES2022",
      "dom"
    ]
  },
  "angularCompilerOptions": {
    "enableI18nLegacyMessageIdFormat": false,
    "strictInjectionParameters": true,
    "strictInputAccessModifiers": true,
    "strictTemplates": true
  }
}

```


lixo-no-ponto-front\tscconfig.spec.json ----

```

/*      To      learn      more      about      Typescript      configuration      file:
https://www.typescriptlang.org/docs/handbook/tsconfig-json.html. */
/*      To      learn      more      about      Angular      compiler      options:
https://angular.dev/reference/configs/angular-compiler-options. */
{
  "extends": "./tsconfig.json",
  "compilerOptions": {
    "outDir": "./out-tsc/spec",
    "types": [
      "jasmine"
    ]
  },
  "include": [
    "src/**/*.spec.ts",
    "src/**/*.d.ts"
  ]
}

```

Apêndice E - Código Fonte Back End:

Arquivo: lixo-no-ponto\concatenet.js ----

```

const fs = require('fs');
const path = require('path');

const outputFilePath = 'output.txt';
const ignoreDirs = ['.mvn', '.gitignore', 'target', 'node_modules', 'dist', '.idea', '.angular', '.git', '.env'];
const ignoreExtensions = ['.txt', '.gitignore', '.ico', '.svg', '.png', '.jpg', '.jpeg', '.gif', '.webp', '.mp4', '.mp3', '.wav', '.flac', '.ogg', '.webm', '.pdf', '.doc', '.docx', '.xls', '.xlsx', '.ppt', '.pptx', '.zip', '.rar', '.tar', '.gz', '.7z', '.exe', '.dll', '.so', '.dylib', '.app', '.dmg', '.pkg', '.deb', '.rpm', '.apk', '.ipa', '.jar', '.war', '.ear', '.class', '.pyc', '.o', '.obj', '.lib', '.a', '.dll', '.so', '.dylib', '.app', '.dmg', '.pkg', '.deb', '.rpm', '.apk', '.ipa', '.jar', '.war', '.ear', '.class', '.pyc', '.o', '.obj', '.lib', '.a', '.dll', '.so', '.dylib', '.app', '.dmg', '.pkg', '.deb', '.rpm', '.apk', '.ipa', '.jar', '.war', '.ear', '.class', '.pyc', '.o', '.obj', '.lib', '.a', '.dll', '.so', '.dylib', '.app', '.dmg', '.pkg', '.deb', '.rpm', '.apk', '.ipa', '.jar', '.war', '.ear', '.class', '.pyc', '.o', '.obj', '.lib', '.a', '.dll', '.so', '.dylib', '.app', '.dmg', '.pkg', '.deb', '.rpm', '.apk', '.ipa', '.jar', '.war', '.ear', '.class', '.pyc', '.o', '.obj', '.lib', '.a', '.dll', '.so', '.dylib', '.app', '.dmg', '.pkg', '.deb', '.rpm', '.apk', '.ipa', '.jar', '.war', '.ear', '.class', '.pyc', '.o', '.obj', '.lib'];
const projectDir = path.basename(__dirname);

```

```

function isIgnored(filePath) {
  const ext = path.extname(filePath);
  return ignoreDirs.some(dir => filePath.includes(dir)) || ignoreExtensions.includes(ext);
}

function readFilesRecursively(dir, fileList = []) {
  const files = fs.readdirSync(dir);
  files.forEach(file => {
    const filePath = path.join(dir, file);
    const stat = fs.statSync(filePath);
    if (stat.isDirectory()) {
      if (!isIgnored(filePath)) {
        readFilesRecursively(filePath, fileList);
      }
    } else {
      if (!isIgnored(filePath)) {
        fileList.push(filePath);
      }
    }
  });
  return fileList;
}

function concatenateFiles(fileList, outputFilePath) {
  const writeStream = fs.createWriteStream(outputFilePath);
  const totalFiles = fileList.length;

  fileList.forEach((file, index) => {
    const fileContent = fs.readFileSync(file, 'utf-8');
    const relativeFilePath = path.relative(__dirname, file);
    const displayFilePath = path.join(projectDir, relativeFilePath);
    const progress = ((index + 1) / totalFiles * 100).toFixed(2);

    console.log(`Processing ${displayFilePath} (${progress}%)`);

    writeStream.write(`\nArquivo: ${displayFilePath} ----\n\n`);
    writeStream.write(fileContent);
    writeStream.write("\n\n");
  });

  writeStream.end();
}

const allFiles = readFilesRecursively(__dirname);
concatenateFiles(allFiles, outputFilePath);

console.log(`All files have been concatenated into ${outputFilePath}`);

```

Arquivo: lixo-no-ponto\HELP.md ----

Getting Started

Reference Documentation

For further reference, please consider the following sections:

- * [Official Apache Maven documentation](https://maven.apache.org/guides/index.html)
- * [Spring Boot Maven Plugin Reference Guide](https://docs.spring.io/spring-boot/docs/3.3.1/maven-plugin/reference/html/)
- * [Create an OCI image](https://docs.spring.io/spring-boot/docs/3.3.1/maven-plugin/reference/html/#build-image)
- * [Spring Data JPA](https://docs.spring.io/spring-boot/docs/3.3.1/reference/htmlsingle/index.html#data.jpa-and-spring-data)
- * [Spring Boot DevTools](https://docs.spring.io/spring-boot/docs/3.3.1/reference/htmlsingle/index.html#using.devtools)
- * [Validation](https://docs.spring.io/spring-boot/docs/3.3.1/reference/htmlsingle/index.html#io.validation)

Guides

The following guides illustrate how to use some features concretely:

- * [Accessing Data with JPA](https://spring.io/guides/gs/accessing-data-jpa/)
- * [Validation](https://spring.io/guides/gs/validating-form-input/)

Maven Parent overrides

Due to Maven's design, elements are inherited from the parent POM to the project POM. While most of the inheritance is fine, it also inherits unwanted elements like ``<license>`` and ``<developers>`` from the parent.

To prevent this, the project POM contains empty overrides for these elements.

If you manually switch to a different parent and actually want the inheritance, you need to remove those overrides.

Arquivo: lixo-no-ponto\lixo-no-ponto.iml ----

```

<?xml version="1.0" encoding="UTF-8"?>
<module version="4">
  <component name="FacetManager">
    <facet type="jpa" name="JPA">
      <configuration>
        <setting name="validation-enabled" value="true" />
        <setting name="provider-name" value="Hibernate" />
        <datasource-mapping>
          <factory-entry name="entityManagerFactory"
value="b52f50d5-fd4e-4287-9bd0-7afae9d06517" />
        </datasource-mapping>
        <naming-strategy-map />
      </configuration>
    </facet>
  </component>
</module>

```

Arquivo: lixo-no-ponto\mvnw ----

```

#!/bin/sh
# -----
# Licensed to the Apache Software Foundation (ASF) under one
# or more contributor license agreements. See the NOTICE file
# distributed with this work for additional information
# regarding copyright ownership. The ASF licenses this file
# to you under the Apache License, Version 2.0 (the
# "License"); you may not use this file except in compliance
# with the License. You may obtain a copy of the License at
#
# https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing,
# software distributed under the License is distributed on an
# "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied. See the License for the
# specific language governing permissions and limitations
# under the License.
# -----
# -----
# Apache Maven Wrapper startup batch script, version 3.3.2
#
# Optional ENV vars
# -----
# JAVA_HOME - location of a JDK home dir, required when download maven via java
source

```

```

# MVNW_REPOURL - repo url base for downloading maven distribution
# MVNW_USERNAME/MVNW_PASSWORD - user and password for downloading maven
# MVNW_VERBOSE - true: enable verbose log; debug: trace the mvnw script; others:
silence the output
# -----

set -euf
[ "${MVNW_VERBOSE-}" != debug ] || set -x

# OS specific support.
native_path() { printf %s\n "$1"; }
case "$(uname)" in
CYGWIN* | MINGW*)
  [ -z "${JAVA_HOME-}" ] || JAVA_HOME="$(cygpath --unix "$JAVA_HOME")"
  native_path() { cygpath --path --windows "$1"; }
  ;;
esac

# set JAVACMD and JAVACCMD
set_java_home() {
  # For Cygwin and MinGW, ensure paths are in Unix format before anything is touched
  if [ -n "${JAVA_HOME-}" ]; then
    if [ -x "$JAVA_HOME/jre/sh/java" ]; then
      # IBM's JDK on AIX uses strange locations for the executables
      JAVACMD="$JAVA_HOME/jre/sh/java"
      JAVACCMD="$JAVA_HOME/jre/sh/javac"
    else
      JAVACMD="$JAVA_HOME/bin/java"
      JAVACCMD="$JAVA_HOME/bin/javac"
    fi

    if [ ! -x "$JAVACMD" ] || [ ! -x "$JAVACCMD" ]; then
      echo "The JAVA_HOME environment variable is not defined correctly, so mvnw cannot
run." >&2
      echo "JAVA_HOME is set to \"\$JAVA_HOME\", but \"\$JAVA_HOME/bin/java\" or
\"\$JAVA_HOME/bin/javac\" does not exist." >&2
      return 1
    fi
  fi
else
  JAVACMD="$(
  'set' +e
  'unset' -f command 2>/dev/null
  'command' -v java
)" || :
  JAVACCMD="$(
  'set' +e
  'unset' -f command 2>/dev/null
  'command' -v javac

```

```

)" || :

if [ ! -x "${JAVACMD-}" ] || [ ! -x "${JAVACCMD-}" ]; then
    echo "The java/javac command does not exist in PATH nor is JAVA_HOME set, so mvnw
cannot run." >&2
    return 1
fi
fi
}

# hash string like Java String::hashCode
hash_string() {
    str="${1:-}" h=0
    while [ -n "$str" ]; do
        char="${str%${str#?}}"
        h=$((h * 31 + $(LC_CTYPE=C printf %d "$char"))) % 4294967296)
        str="${str#?}"
    done
    printf %x\n $h
}

verbose() { ;; }
[ "${MVNW_VERBOSE-}" != true ] || verbose() { printf %s\n "${1-}"; }

die() {
    printf %s\n "$1" >&2
    exit 1
}

trim() {
    # MWRAPPER-139:
    # Trims trailing and leading whitespace, carriage returns, tabs, and linefeeds.
    # Needed for removing poorly interpreted newline sequences when running in more
    # exotic environments such as mingw bash on Windows.
    printf "%s" "${1}" | tr -d '[:space:]'
}

# parse distributionUrl and optional distributionSha256Sum, requires
.mvn/wrapper/maven-wrapper.properties
while IFS="=" read -r key value; do
    case "${key-}" in
        distributionUrl) distributionUrl=$(trim "${value-}") ;;
        distributionSha256Sum) distributionSha256Sum=$(trim "${value-}") ;;
    esac
done <"${0%/*}/.mvn/wrapper/maven-wrapper.properties"
[ -n "${distributionUrl-}" ] || die "cannot read distributionUrl property in
${0%/*}/.mvn/wrapper/maven-wrapper.properties"

```

```

case "${distributionUrl##*/}" in
maven-mvnd-*bin.*)
  MVN_CMD=mvnd.sh _MVNW_REPO_PATTERN=/maven/mvnd/
  case "${PROCESSOR_ARCHITECTURE-}${PROCESSOR_ARCHITECTURE-}${uname
-a)" in
  *AMD64:CYGWIN* | *AMD64:MINGW*) distributionPlatform=windows-amd64 ;;
  :Darwin*x86_64) distributionPlatform=darwin-amd64 ;;
  :Darwin*arm64) distributionPlatform=darwin-aarch64 ;;
  :Linux*x86_64*) distributionPlatform=linux-amd64 ;;
  *)
    echo "Cannot detect native platform for mvnd on $(uname)-$(uname -m), use pure java
version" >&2
    distributionPlatform=linux-amd64
    ;;
  esac
  distributionUrl="${distributionUrl%-bin.*}-${distributionPlatform}.zip"
  ;;
maven-mvnd-*) MVN_CMD=mvnd.sh _MVNW_REPO_PATTERN=/maven/mvnd/ ;;
*) MVN_CMD="mvn${0##*/mavnw}" _MVNW_REPO_PATTERN=/org/apache/maven/ ;;
esac

# apply MVNW_REPOURL and calculate MAVEN_HOME
#           maven           home           pattern:
~/m2/wrapper/dists/{apache-maven-<version>,maven-mvnd-<version>-<platform>}/<hash>
[           -z           "${MVNW_REPOURL-}"           ]           ||
distributionUrl="$MVNW_REPOURL$_MVNW_REPO_PATTERN${distributionUrl#*"$MVN
W_REPO_PATTERN"}"
distributionUrlName="${distributionUrl##*/}"
distributionUrlNameMain="${distributionUrlName%.*}"
distributionUrlNameMain="${distributionUrlNameMain%-bin}"
MAVEN_USER_HOME="${MAVEN_USER_HOME:-${HOME}/.m2}"
MAVEN_HOME="${MAVEN_USER_HOME}/wrapper/dists/${distributionUrlNameMain-}/$(ha
sh_string "$distributionUrl")"

exec_maven() {
  unset MVNW_VERBOSE MVNW_USERNAME MVNW_PASSWORD MVNW_REPOURL ||
:
  exec "$MAVEN_HOME/bin/$MVN_CMD" "$@" || die "cannot exec
$MAVEN_HOME/bin/$MVN_CMD"
}

if [ -d "$MAVEN_HOME" ]; then
  verbose "found existing MAVEN_HOME at $MAVEN_HOME"
  exec_maven "$@"
fi

case "${distributionUrl-}" in
*?-bin.zip | *?maven-mvnd-*?*.zip) ;;

```

```

*) die "distributionUrl is not valid, must match *-bin.zip or maven-mvnd-*.zip, but found
'${distributionUrl}'" ;;
esac

```

```

# prepare tmp dir
if TMP_DOWNLOAD_DIR="$(mktemp -d)" && [ -d "$TMP_DOWNLOAD_DIR" ]; then
  clean() { rm -rf -- "$TMP_DOWNLOAD_DIR"; }
  trap clean HUP INT TERM EXIT
else
  die "cannot create temp dir"
fi

```

```

mkdir -p -- "${MAVEN_HOME%/*}"

```

```

# Download and Install Apache Maven
verbose "Couldn't find MAVEN_HOME, downloading and installing it ..."
verbose "Downloading from: $distributionUrl"
verbose "Downloading to: $TMP_DOWNLOAD_DIR/$distributionUrlName"

```

```

# select .zip or .tar.gz
if ! command -v unzip >/dev/null; then
  distributionUrl="${distributionUrl%.zip}.tar.gz"
  distributionUrlName="${distributionUrl###*}"
fi

```

```

# verbose opt
__MVNW_QUIET_WGET=--quiet          __MVNW_QUIET_CURL=--silent
__MVNW_QUIET_UNZIP=-q __MVNW_QUIET_TAR=
[ "${MVNW_VERBOSE-}" != true ] || __MVNW_QUIET_WGET=__MVNW_QUIET_CURL=
__MVNW_QUIET_UNZIP=__MVNW_QUIET_TAR=v

```

```

# normalize http auth
case "${MVNW_PASSWORD:+has-password}" in
  *) MVNW_USERNAME=" MVNW_PASSWORD=" ;;
  has-password) [ -n "${MVNW_USERNAME-}" ] || MVNW_USERNAME="
MVNW_PASSWORD=" ;;
esac

```

```

if [ -z "${MVNW_USERNAME-}" ] && command -v wget >/dev/null; then
  verbose "Found wget ... using wget"
  wget ${__MVNW_QUIET_WGET:+"${__MVNW_QUIET_WGET"}"} "$distributionUrl" -O
"$TMP_DOWNLOAD_DIR/$distributionUrlName" || die "wget: Failed to fetch $distributionUrl"
elif [ -z "${MVNW_USERNAME-}" ] && command -v curl >/dev/null; then
  verbose "Found curl ... using curl"
  curl ${__MVNW_QUIET_CURL:+"${__MVNW_QUIET_CURL"}"} -f -L -o
"$TMP_DOWNLOAD_DIR/$distributionUrlName" "$distributionUrl" || die "curl: Failed to fetch
$distributionUrl"
elif set_java_home; then

```



```

verbose "Falling back to use Java to download"
javaSource="$TMP_DOWNLOAD_DIR/Downloader.java"
targetZip="$TMP_DOWNLOAD_DIR/$distributionUrlName"
cat >"$javaSource" <<-END
    public class Downloader extends java.net.Authenticator
    {
        protected java.net.PasswordAuthentication getPasswordAuthentication()
        {
            return new java.net.PasswordAuthentication( System.getenv(
"MVNW_USERNAME" ), System.getenv( "MVNW_PASSWORD" ).toCharArray() );
        }
        public static void main( String[] args ) throws Exception
        {
            setDefault( new Downloader() );
            java.nio.file.Files.copy( java.net.URI.create( args[0] ).toURL().openStream(),
java.nio.file.Paths.get( args[1] ).toAbsolutePath().normalize() );
        }
    }
END
# For Cygwin/MinGW, switch paths to Windows format before running javac and java
verbose " - Compiling Downloader.java ..."
"${native_path "$JAVACMD"}" "${native_path "$javaSource"}" || die "Failed to compile
Downloader.java"
verbose " - Running Downloader.java ..."
"${native_path "$JAVACMD"}" -cp "${native_path "$TMP_DOWNLOAD_DIR"}" Downloader
"$distributionUrl" "${native_path "$targetZip"}"
fi

# If specified, validate the SHA-256 sum of the Maven distribution zip file
if [ -n "${distributionSha256Sum-}" ]; then
    distributionSha256Result=false
    if [ "$MVN_CMD" = mvnd.sh ]; then
        echo "Checksum validation is not supported for maven-mvnd." >&2
        echo "Please disable validation by removing 'distributionSha256Sum' from your
maven-wrapper.properties." >&2
        exit 1
    elif command -v sha256sum >/dev/null; then
        if echo "$distributionSha256Sum $TMP_DOWNLOAD_DIR/$distributionUrlName" |
sha256sum -c >/dev/null 2>&1; then
            distributionSha256Result=true
        fi
    elif command -v shasum >/dev/null; then
        if echo "$distributionSha256Sum $TMP_DOWNLOAD_DIR/$distributionUrlName" |
shasum -a 256 -c >/dev/null 2>&1; then
            distributionSha256Result=true
        fi
    else

```

```

    echo "Checksum validation was requested but neither 'sha256sum' or 'shasum' are
available." >&2
    echo "Please install either command, or disable validation by removing
'distributionSha256Sum' from your maven-wrapper.properties." >&2
    exit 1
fi
if [ $distributionSha256Result = false ]; then
    echo "Error: Failed to validate Maven distribution SHA-256, your Maven distribution might
be compromised." >&2
    echo "If you updated your Maven version, you need to update the specified
distributionSha256Sum property." >&2
    exit 1
fi
fi

# unzip and move
if command -v unzip >/dev/null; then
    unzip          ${__MVNW_QUIET_UNZIP:+"${__MVNW_QUIET_UNZIP}"}
"$TMP_DOWNLOAD_DIR/$distributionUrlName" -d "$TMP_DOWNLOAD_DIR" || die "failed
to unzip"
else
    tar            xzf${__MVNW_QUIET_TAR:+"${__MVNW_QUIET_TAR}"}
"$TMP_DOWNLOAD_DIR/$distributionUrlName" -C "$TMP_DOWNLOAD_DIR" || die "failed
to untar"
fi
printf            %s\n                                "$distributionUrl"
>"$TMP_DOWNLOAD_DIR/$distributionUrlNameMain/mvnw.url"
mv -- "$TMP_DOWNLOAD_DIR/$distributionUrlNameMain" "$MAVEN_HOME" || [ -d
"$MAVEN_HOME" ] || die "fail to move MAVEN_HOME"

clean || :
exec_maven "$@"

```

Arquivo: lixo-no-ponto\mvnw.cmd ----

<# : batch portion

```

@REM -----
@REM Licensed to the Apache Software Foundation (ASF) under one
@REM or more contributor license agreements. See the NOTICE file
@REM distributed with this work for additional information
@REM regarding copyright ownership. The ASF licenses this file
@REM to you under the Apache License, Version 2.0 (the
@REM "License"); you may not use this file except in compliance
@REM with the License. You may obtain a copy of the License at
@REM
@REM https://www.apache.org/licenses/LICENSE-2.0

```

```

@REM
@REM Unless required by applicable law or agreed to in writing,
@REM software distributed under the License is distributed on an
@REM "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
@REM KIND, either express or implied. See the License for the
@REM specific language governing permissions and limitations
@REM under the License.
@REM -----

@REM -----
@REM Apache Maven Wrapper startup batch script, version 3.3.2
@REM
@REM Optional ENV vars
@REM MVNW_REPOURL - repo url base for downloading maven distribution
@REM MVNW_USERNAME/MVNW_PASSWORD - user and password for downloading
maven
@REM MVNW_VERBOSE - true: enable verbose log; others: silence the output
@REM -----

@if "%__MVNW_ARG0_NAME__%"==" " (set __MVNW_ARG0_NAME__=%~nx0)
@set __MVNW_CMD__=
@set __MVNW_ERROR__=
@set __MVNW_PSMODULEP_SAVE=%PSModulePath%
@set PSModulePath=
@for /f "usebackq tokens=1* delims==" %%A IN (`powershell -noprofile "&
{$scriptDir='%~dp0'; $script='%__MVNW_ARG0_NAME__'; icm -ScriptBlock
([Scriptblock]::Create((Get-Content -Raw '%~f0')) -NoNewScope}") DO @(
  if "%%A"=="MVN_CMD" (set __MVNW_CMD__=%%B) ELSE IF "%%B"==" " (echo %%A)
  ELSE (echo %%A=%%B)
)
@set PSModulePath=%__MVNW_PSMODULEP_SAVE%
@set __MVNW_PSMODULEP_SAVE=
@set __MVNW_ARG0_NAME__=
@set MVNW_USERNAME=
@set MVNW_PASSWORD=
@if not "%__MVNW_CMD__%"==" " (%__MVNW_CMD__% %*)
@echo Cannot start maven from wrapper >&2 && exit /b 1
@goto :EOF
: end batch / begin powershell #>

$errorActionPreference = "Stop"
if ($env:MVNW_VERBOSE -eq "true") {
  $VerbosePreference = "Continue"
}

# calculate distributionUrl, requires .mvn/wrapper/maven-wrapper.properties
$distributionUrl = (Get-Content -Raw "$scriptDir/.mvn/wrapper/maven-wrapper.properties" |
ConvertFrom-StringData).distributionUrl

```

```

if (!$distributionUrl) {
    Write-Error "cannot read distributionUrl property in
$scriptDir/.mvn/wrapper/maven-wrapper.properties"
}

switch -wildcard -casesensitive ( $($distributionUrl -replace '^.*/',) ) {
    "maven-mvnd-*" {
        $USE_MVND = $true
        $distributionUrl = $distributionUrl -replace '-bin\[^\.*\]*$',"-windows-amd64.zip"
        $MVN_CMD = "mvnd.cmd"
        break
    }
    default {
        $USE_MVND = $false
        $MVN_CMD = $script -replace '^mvnw','mvn'
        break
    }
}

# apply MVNW_REPOURL and calculate MAVEN_HOME
#           maven           home           pattern:
~/m2/wrapper/dists/{apache-maven-<version>,maven-mvnd-<version>-<platform>}/<hash>
if ($env:MVNW_REPOURL) {
    $MVNW_REPO_PATTERN = if ($USE_MVND) { "/org/apache/maven/" } else {
"/maven/mvnd/" }
    $distributionUrl = "$env:MVNW_REPOURL$MVNW_REPO_PATTERN$( $distributionUrl
-replace '^.*'+$MVNW_REPO_PATTERN,)"
}
$distributionUrlName = $distributionUrl -replace '^.*/',"
$distributionUrlNameMain = $distributionUrlName -replace '\.[^\.]*$'," -replace '-bin$',"
$MAVEN_HOME_PARENT = "$HOME/m2/wrapper/dists/$distributionUrlNameMain"
if ($env:MAVEN_USER_HOME) {
    $MAVEN_HOME_PARENT =
"$env:MAVEN_USER_HOME/wrapper/dists/$distributionUrlNameMain"
}
$MAVEN_HOME_NAME =
([System.Security.Cryptography.MD5]::Create().ComputeHash([byte[]][char[]]$distributionUrl
| ForEach-Object {$_.ToString("x2")}) -join "
$MAVEN_HOME = "$MAVEN_HOME_PARENT/$MAVEN_HOME_NAME"

if (Test-Path -Path "$MAVEN_HOME" -PathType Container) {
    Write-Verbose "found existing MAVEN_HOME at $MAVEN_HOME"
    Write-Output "MVN_CMD=$MAVEN_HOME/bin/$MVN_CMD"
    exit $?
}

if (! $distributionUrlNameMain -or ($distributionUrlName -eq $distributionUrlNameMain)) {
    Write-Error "distributionUrl is not valid, must end with *-bin.zip, but found $distributionUrl"
}

```

```

}

# prepare tmp dir
$tmp_DOWNLOAD_DIR HOLDER = New-TemporaryFile
$tmp_DOWNLOAD_DIR = New-Item -ItemType Directory -Path
"$tmp_DOWNLOAD_DIR HOLDER.dir"
$tmp_DOWNLOAD_DIR HOLDER.Delete() | Out-Null
trap {
    if ($tmp_DOWNLOAD_DIR.Exists) {
        try { Remove-Item $tmp_DOWNLOAD_DIR -Recurse -Force | Out-Null }
        catch { Write-Warning "Cannot remove $tmp_DOWNLOAD_DIR" }
    }
}

New-Item -ItemType Directory -Path "$MAVEN_HOME_PARENT" -Force | Out-Null

# Download and Install Apache Maven
Write-Verbose "Couldn't find MAVEN_HOME, downloading and installing it ..."
Write-Verbose "Downloading from: $distributionUrl"
Write-Verbose "Downloading to: $tmp_DOWNLOAD_DIR/$distributionUrlName"

$webclient = New-Object System.Net.WebClient
if ($env:MVNW_USERNAME -and $env:MVNW_PASSWORD) {
    $webclient.Credentials = New-Object
    System.Net.NetworkCredential($env:MVNW_USERNAME, $env:MVNW_PASSWORD)
}
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
$webclient.DownloadFile($distributionUrl, "$tmp_DOWNLOAD_DIR/$distributionUrlName")
| Out-Null

# If specified, validate the SHA-256 sum of the Maven distribution zip file
$distributionSha256Sum = (Get-Content -Raw
"$scriptDir/.mvn/wrapper/maven-wrapper.properties" |
ConvertFrom-StringData).distributionSha256Sum
if ($distributionSha256Sum) {
    if ($USE_MVND) {
        Write-Error "Checksum validation is not supported for maven-mvnd. `nPlease disable
validation by removing 'distributionSha256Sum' from your maven-wrapper.properties."
    }
    Import-Module $PSHOME\Modules\Microsoft.PowerShell.Utility -Function Get-FileHash
    if ((Get-FileHash "$tmp_DOWNLOAD_DIR/$distributionUrlName" -Algorithm
SHA256).Hash.ToLower() -ne $distributionSha256Sum) {
        Write-Error "Error: Failed to validate Maven distribution SHA-256, your Maven distribution
might be compromised. If you updated your Maven version, you need to update the specified
distributionSha256Sum property."
    }
}
}

```

```

# unzip and move
Expand-Archive "$TMP_DOWNLOAD_DIR/$distributionUrlName" -DestinationPath
"$TMP_DOWNLOAD_DIR" | Out-Null
Rename-Item -Path "$TMP_DOWNLOAD_DIR/$distributionUrlNameMain" -NewName
$MAVEN_HOME_NAME | Out-Null
try {
    Move-Item -Path "$TMP_DOWNLOAD_DIR/$MAVEN_HOME_NAME" -Destination
    $MAVEN_HOME_PARENT | Out-Null
} catch {
    if (!(Test-Path -Path "$MAVEN_HOME" -PathType Container)) {
        Write-Error "fail to move MAVEN_HOME"
    }
} finally {
    try { Remove-Item $TMP_DOWNLOAD_DIR -Recurse -Force | Out-Null }
    catch { Write-Warning "Cannot remove $TMP_DOWNLOAD_DIR" }
}

Write-Output "MVN_CMD=$MAVEN_HOME/bin/$MVN_CMD"

```

Arquivo: lixo-no-ponto\pom.xml ----

```

<?xml version="1.0" encoding="UTF-8"?>
<project
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://maven.apache.org/POM/4.0.0"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.3.1</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.example</groupId>
    <artifactId>lixo-no-ponto</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>lixo-no-ponto</name>
    <description>lixo-no-ponto</description>
    <url/>
    <licenses>
        <license/>
    </licenses>
    <developers>
        <developer/>
    </developers>
    <scm>

```

```
<connection/>
<developerConnection/>
<tag/>
<url/>
</scm>
<properties>
  <java.version>17</java.version>
</properties>
<dependencies>
  <dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-api</artifactId>
    <version>0.12.6</version>
  </dependency>
  <dependency>
    <groupId>org.springframework.plugin</groupId>
    <artifactId>spring-plugin-core</artifactId>
    <version>3.0.0</version>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.security</groupId>
    <artifactId>spring-security-crypto</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-data-jpa</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-validation</artifactId>
  </dependency>
  <dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-impl</artifactId>
    <version>0.12.5</version>
  </dependency>
  <dependency>
    <groupId>io.jsonwebtoken</groupId>
    <artifactId>jjwt-jackson</artifactId>
    <version>0.12.5</version>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
```

```

    <scope>runtime</scope>
    <optional>true</optional>
</dependency>
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-test</artifactId>
  <scope>test</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
  <groupId>org.flywaydb</groupId>
  <artifactId>flyway-core</artifactId>
</dependency>
<dependency>
  <groupId>org.flywaydb</groupId>
  <artifactId>flyway-database-postgresql</artifactId>
</dependency>
<dependency>
  <groupId>org.projectlombok</groupId>
  <artifactId>lombok</artifactId>
  <scope>annotationProcessor</scope>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
</dependencies>

<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
    </plugin>
  </plugins>
</build>

```



```
</project>
```

Arquivo:

```
lixo-no-ponto\src\main\java\com\example\lixonoponto\config\AuthenticationTokenFilter.java
```

```
----
```

```
package com.example.lixonoponto.config;

import com.example.lixonoponto.model.Usuario;
import com.example.lixonoponto.repository.UsuarioRepository;
import com.example.lixonoponto.service.TokenService;
import jakarta.servlet.FilterChain;
import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;
import org.springframework.security.core.context.SecurityContextHolder;
import org.springframework.web.filter.OncePerRequestFilter;

import java.io.IOException;
import java.util.List;

public class AuthenticationTokenFilter extends OncePerRequestFilter {
    private final TokenService tokenService;
    private final UsuarioRepository usuarioRepository;

    public AuthenticationTokenFilter(TokenService tokenService, UsuarioRepository
usuarioRepository) {
        this.tokenService = tokenService;
        this.usuarioRepository = usuarioRepository;
    }

    @Override
    protected void doFilterInternal(HttpServletRequest request, HttpServletResponse
response, FilterChain filterChain) throws ServletException, IOException {
        String token = getTokenHeader(request);
        boolean valid = token != null && tokenService.isTokenValid(token);
        if (valid) {
            authenticateClient(token);
        }
        filterChain.doFilter(request, response);
    }

    private void authenticateClient(String token) {
        if (tokenService.isTokenValid(token)) {
```

```

        Long id = tokenService.getUserId(token);
        Usuario usuario = usuarioRepository.findById(id).get();
        UsernamePasswordAuthenticationToken authenticationToken = new
UsernamePasswordAuthenticationToken(usuario, null, List.of());
        SecurityContextHolder.getContext().setAuthentication(authenticationToken);
    }
}

private String getTokenHeader(HttpServletRequest request) {
    String token = request.getHeader("Authorization");
    if (token == null || !token.startsWith("Bearer ")) {
        return null;
    }
    return token.substring(7);
}
}
}

```

Arquivo: lixo-no-ponto\src\main\java\com\example\lixonoponto\config\SecurityConfig.java ----

```

package com.example.lixonoponto.config;

import com.example.lixonoponto.repository.UsuarioRepository;
import com.example.lixonoponto.service.TokenService;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.method.configuration.EnableMethodSecurity;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import
org.springframework.security.config.annotation.web.configurers.AbstractHttpConfigurer;
import org.springframework.security.config.http.SessionCreationPolicy;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.security.web.SecurityFilterChain;
import
org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter;
import org.springframework.web.cors.CorsConfiguration;
import org.springframework.web.cors.UrlBasedCorsConfigurationSource;
import org.springframework.web.filter.CorsFilter;

import java.util.Collections;
import java.util.List;

@EnableWebSecurity

```

```

@EnableMethodSecurity
@Configuration
public class SecurityConfig {

    private static final String[] PUBLIC_AUTH = {"/api/user/auth/register",
"/api/user/auth/login"};
    private final TokenService tokenService;
    private final UsuarioRepository usuarioRepository;

    public SecurityConfig(TokenService tokenService, UsuarioRepository usuarioRepository) {
        this.tokenService = tokenService;
        this.usuarioRepository = usuarioRepository;
    }

    @Bean
    public PasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http.csrf(AbstractHttpConfigurer::disable)
                .sessionManagement(c ->
c.sessionCreationPolicy(SessionCreationPolicy.STATELESS))
                .authorizeHttpRequests(c ->
c.requestMatchers(PUBLIC_AUTH).permitAll().anyRequest().authenticated())
                .addFilterBefore(new AuthenticationTokenFilter(tokenService, usuarioRepository),
UsernamePasswordAuthenticationFilter.class);
        return http.build();
    }

    @Bean
    public CorsFilter corsFilter() {
        UrlBasedCorsConfigurationSource source = new UrlBasedCorsConfigurationSource();
        CorsConfiguration config = new CorsConfiguration();

        config.setAllowCredentials(true);
        config.setAllowedOrigins(List.of("http://localhost:4200", "https://lixonoponto.live"));
        config.setAllowedMethods(Collections.singletonList("*"));
        config.setAllowedHeaders(Collections.singletonList("*"));
        source.registerCorsConfiguration("/**", config);
        return new CorsFilter(source);
    }
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\controller\EmpresaReciclagemController.java ----

```
package com.example.lixonoponto.controller;

import com.example.lixonoponto.model.EmpresaReciclagem;
import com.example.lixonoponto.model.dto.EmpresaReciclagemRequest;
import com.example.lixonoponto.service.EmpresaReciclagemService;
import jakarta.servlet.http.HttpServletRequest;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.support.ServletUriComponentsBuilder;

import java.net.URI;
import java.util.List;

@RestController
@RequestMapping("/api/recycling-company")
public class EmpresaReciclagemController {

    private final EmpresaReciclagemService empresaReciclagemService;

    public EmpresaReciclagemController(EmpresaReciclagemService
    empresaReciclagemService) {
        this.empresaReciclagemService = empresaReciclagemService;
    }

    @GetMapping
    public ResponseEntity<List<EmpresaReciclagem>> getAllEmpresasReciclagem() {
        List<EmpresaReciclagem> empresasReciclagem =
    empresaReciclagemService.findAll();
        return ResponseEntity.ok(empresasReciclagem);
    }

    // @PreAuthorize("hasAuthority('ADMIN')")
    @PostMapping("/create")
    public ResponseEntity<EmpresaReciclagem> createEmpresaReciclagem(@RequestBody
    EmpresaReciclagemRequest empresaReciclagem, HttpServletRequest request) {
        EmpresaReciclagem createdEmpresaReciclagem =
    empresaReciclagemService.save(empresaReciclagem);
        URI location = ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}")
        .buildAndExpand(createdEmpresaReciclagem.getId()).toUri();
        return ResponseEntity.created(location).body(createdEmpresaReciclagem);
    }
}
```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\controller\PontoDescarteController.jav

a ----

```
package com.example.lixonoponto.controller;

import com.example.lixonoponto.model.PontoDescarte;
import com.example.lixonoponto.model.dto.PontoDescarteDTO;
import com.example.lixonoponto.model.dto.PontoDescarteRequest;
import com.example.lixonoponto.model.dto.PontoDescarteUpdateRequest;
import com.example.lixonoponto.service.PontoDescarteService;
import com.example.lixonoponto.service.TokenService;
import jakarta.servlet.http.HttpServletRequest;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.support.ServletUriComponentsBuilder;

import java.net.URI;
import java.util.List;
import java.util.Optional;

@RestController
@RequestMapping("/api/disposal-point")
public class PontoDescarteController {

    private final PontoDescarteService pontoDescarteService;
    private final TokenService tokenService;

    public PontoDescarteController(PontoDescarteService pontoDescarteService,
TokenService tokenService) {
        this.pontoDescarteService = pontoDescarteService;
        this.tokenService = tokenService;
    }

    @GetMapping
    public List<PontoDescarteDTO> getPontosDescarte(@RequestParam Double south,
@RequestParam Double north, @RequestParam Double west, @RequestParam Double
east, @RequestParam(required = false) List<Long> wastesId) {
        return pontoDescarteService.getAllPontoDescarte(south, north, west, east, wastesId);
    }

    @GetMapping("/{id}")
    public Optional<PontoDescarte> getPontoDescarte(@PathVariable Long id) {
        return pontoDescarteService.getPontoDescarteById(id);
    }

    @PutMapping("/{id}")
```

```

    public ResponseEntity<PontoDescarte> updatePontoDescarte(@PathVariable Long id,
@RequestBody PontoDescarteUpdateRequest request) {
        PontoDescarte      updatedPontoDescarte      =
pontoDescarteService.updatePontoDescarte(id, request);
        return ResponseEntity.ok(updatedPontoDescarte);
    }

// @PreAuthorize("hasAuthority('ADMIN')")
@DeleteMapping("/{id}")
public ResponseEntity<Void> deletePontoDescarte(@PathVariable Long id) {
    pontoDescarteService.deletePontoDescarte(id);
    return ResponseEntity.noContent().build();
}

@PostMapping("/create")
    public ResponseEntity<PontoDescarte> createPontoDescarte(@RequestBody
PontoDescarteRequest pontoDescarteRequest, @RequestHeader("Authorization") String
authHeader, HttpServletRequest request) {
        Long userId = tokenService.getUserId(tokenService.getTokenByRequest(authHeader));
        PontoDescarte      novoPontoDescarte      =
pontoDescarteService.createPontoDescarte(pontoDescarteRequest, userId);
        URI location = ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}")
        .buildAndExpand(novoPontoDescarte.getId()).toUri();
        return ResponseEntity.created(location).body(novoPontoDescarte);
    }

@PostMapping("/verify/{pontoDescarteId}")
    public ResponseEntity<PontoDescarte> verificarPontoDescarte(@PathVariable Long
pontoDescarteId, @RequestHeader("Authorization") String authHeader) {
        Long userId = tokenService.getUserId(tokenService.getTokenByRequest(authHeader));
        PontoDescarte      pontoDescarte      =
pontoDescarteService.addUsuarioInPontoDescarte(pontoDescarteId, userId);
        return ResponseEntity.ok(pontoDescarte);
    }
}
}

```

Arquivo:

```

lixo-no-ponto\src\main\java\com\example\lixonoponto\controller\RotaCaminhaoController.java
a ----

```

```

package com.example.lixonoponto.controller;

```

```

import com.example.lixonoponto.model.RotaCaminhao;
import com.example.lixonoponto.model.dto.RotaCaminhaoRequest;
import com.example.lixonoponto.service.RotaCaminhaoService;
import jakarta.servlet.http.HttpServletRequest;
import org.springframework.http.ResponseEntity;

```

```

import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.support.ServletUriComponentsBuilder;

import java.net.URI;
import java.util.List;

@RestController
@RequestMapping("/api/truck-routing")
public class RotaCaminhaoController {

    private final RotaCaminhaoService rotaCaminhaoService;

    public RotaCaminhaoController(RotaCaminhaoService rotaCaminhaoService) {
        this.rotaCaminhaoService = rotaCaminhaoService;
    }

    // @PreAuthorize("hasAuthority('ADMIN')")
    @PostMapping("/create")
    public ResponseEntity<RotaCaminhao> createRotaCaminhao(@RequestBody
RotaCaminhaoRequest rotaCaminhao, HttpServletRequest request) {
        RotaCaminhao createdRotaCaminhao =
rotaCaminhaoService.createRotaCaminhao(rotaCaminhao);
        URI location = ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}")
        .buildAndExpand(createdRotaCaminhao.getId()).toUri();
        return ResponseEntity.created(location).body(createdRotaCaminhao);
    }

    @GetMapping
    public ResponseEntity<List<RotaCaminhao>>
getRotasCaminhaoByCidade(@RequestParam String cidade, @RequestParam(required =
false) Integer idTipoCaminhao) {
        List<RotaCaminhao> rotasCaminhao =
rotaCaminhaoService.getRotasCaminhaoByCidadeAndTipoCaminhao(cidade,
idTipoCaminhao);
        return ResponseEntity.ok(rotasCaminhao);
    }
}

```

Arquivo:

```

lixo-no-ponto\src\main\java\com\example\lixonoponto\controller\TipoCaminhaoController.java
a ----

```

```

package com.example.lixonoponto.controller;

```

```

import com.example.lixonoponto.model.TipoCaminhao;
import com.example.lixonoponto.service.TipoCaminhaoService;

```

```

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController
@RequestMapping("/api/truck-type")
public class TipoCaminhaoController {

    private final TipoCaminhaoService tipoCaminhaoService;

    @Autowired
    public TipoCaminhaoController(TipoCaminhaoService tipoCaminhaoService) {
        this.tipoCaminhaoService = tipoCaminhaoService;
    }

    @GetMapping
    public List<TipoCaminhao> getAllTiposCaminhao() {
        return tipoCaminhaoService.getAllTipoCaminhao();
    }
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\controller\TipoResiduoController.java

```

package com.example.lixonoponto.controller;

import com.example.lixonoponto.model.TipoResiduo;
import com.example.lixonoponto.service.TipoResiduoService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;

@RestController
@RequestMapping("/api/waste-list")
public class TipoResiduoController {

    private final TipoResiduoService tipoResiduoService;

    @Autowired
    public TipoResiduoController(TipoResiduoService tipoResiduoService) {

```



```

        this.tipoResiduoService = tipoResiduoService;
    }

    @GetMapping
    public List<TipoResiduo> getAllTiposResiduo() {
        return tipoResiduoService.getAllTipoResiduo();
    }
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\controller\UsuarioController.java ----

```

package com.example.lixonoponto.controller;

import com.example.lixonoponto.exception.InvalidCredentialsException;
import com.example.lixonoponto.exception.ResourceNotFoundException;
import com.example.lixonoponto.model.ClasseDeUsuario;
import com.example.lixonoponto.model.Usuario;
import com.example.lixonoponto.model.dto.LoginRequestDTO;
import com.example.lixonoponto.repository.ClasseDeUsuarioRepository;
import com.example.lixonoponto.repository.UsuarioClasseDeUsuarioRepository;
import com.example.lixonoponto.service.TokenService;
import com.example.lixonoponto.service.UsuarioService;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.validation.Valid;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.support.ServletUriComponentsBuilder;

import java.net.URI;
import java.util.HashSet;
import java.util.List;
import java.util.Optional;
import java.util.Set;

@RestController
@RequestMapping("/api/user")
public class UsuarioController {
    private final UsuarioService usuarioService;
    private final TokenService tokenService;
    private final PasswordEncoder passwordEncoder;
    private final ClasseDeUsuarioRepository classeDeUsuarioRepository;
    private final UsuarioClasseDeUsuarioRepository usuarioClasseDeUsuarioRepository;

    @Autowired

```

```

    public UsuarioController(UsuarioService usuarioService, TokenService tokenService,
    PasswordEncoder passwordEncoder, ClasseDeUsuarioRepository
    classeDeUsuarioRepository, UsuarioClasseDeUsuarioRepository
    usuarioClasseDeUsuarioRepository) {
        this.usuarioService = usuarioService;
        this.tokenService = tokenService;
        this.passwordEncoder = passwordEncoder;
        this.classeDeUsuarioRepository = classeDeUsuarioRepository;
        this.usuarioClasseDeUsuarioRepository = usuarioClasseDeUsuarioRepository;
    }

    @GetMapping
    public ResponseEntity<List<Usuario>> getAllUsuarios() {
        List<Usuario> usuarios = usuarioService.getUsuarios();
        return ResponseEntity.ok(usuarios);
    }

    @GetMapping("info")
    public ResponseEntity<Optional<Usuario>>
    getUsuarioById(@RequestHeader("Authorization") String authHeader) {
        Long userId = tokenService.getUserId(tokenService.getTokenByRequest(authHeader));
        Optional<Usuario> usuario = usuarioService.getUsuarioById(userId);
        return ResponseEntity.ok(usuario);
    }

    @PostMapping("auth/register")
    public ResponseEntity<Usuario> createUsuario(@Valid @RequestBody Usuario usuario,
    HttpServletRequest request) {
        usuario.setSenha(passwordEncoder.encode(usuario.getSenha()));

        usuario = usuarioService.addUsuario(usuario);

        ClasseDeUsuario classeDeUsuario =
        classeDeUsuarioRepository.findById(2L).orElseThrow(()
        -> new
        ResourceNotFoundException("Classe de usuário", "id", "USER"));

        if (usuario.getClassesDeUsuario() == null) {
            usuario.setClassesDeUsuario(new HashSet<>());
        }

        usuario.getClassesDeUsuario().add(classeDeUsuario);

        usuario = usuarioService.updateUsuario(usuario);

        URI location = ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}")
        .buildAndExpand(usuario.getId()).toUri();
        return ResponseEntity.created(location).body(usuario);
    }
}

```

```

@PostMapping("auth/login")
public ResponseEntity<String> authenticateUser(@RequestBody LoginRequestDTO
loginRequest, HttpServletRequest request) {
    Optional<Usuario> usuario =
usuarioService.getUsuarioByEmail(loginRequest.getEmail());
    if (usuario.isPresent() && passwordEncoder.matches(loginRequest.getPassword(),
usuario.get().getSenha())) {
        Set<String> authorities = new HashSet<>();
        for (ClasseDeUsuario classeDeUsuario : usuario.get().getClassesDeUsuario()) {
            authorities.add(classeDeUsuario.getName());
        }
        String token = tokenService.generateToken(usuario.get().getId(),
usuario.get().getEmail(), authorities);
        URI location = ServletUriComponentsBuilder.fromCurrentRequest().path("/{id}")
            .buildAndExpand(usuario.get().getId()).toUri();
        return ResponseEntity.created(location).body(token);
    } else {
        throw new InvalidCredentialsException();
    }
}
}

```

```

@PutMapping()
public ResponseEntity<Usuario> updateUsuario(@RequestBody Usuario usuario) {
    Usuario updatedUsuario = usuarioService.updateUsuario(usuario);
    return ResponseEntity.ok(updatedUsuario);
}

```

```

@DeleteMapping("/{id}")
public ResponseEntity<Void> deleteUsuario(@PathVariable Long id) {
    usuarioService.deleteUsuario(id);
    return ResponseEntity.ok().build();
}
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\exception\GlobalExceptionHandler.java
a ----

```
package com.example.lixonoponto.exception;
```

```
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.MethodArgumentNotValidException;
import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
```

```

@ControllerAdvice
public class GlobalExceptionHandler {

    @ExceptionHandler(Exception.class)
    public ResponseEntity<?> handleException(Exception e) {
        return new ResponseEntity<>(e.getMessage(),
        HttpStatus.INTERNAL_SERVER_ERROR);
    }

    @ExceptionHandler(MethodArgumentNotValidException.class)
    public ResponseEntity<?>
    customValidationErrorHandling(MethodArgumentNotValidException exception) {
        var fieldError = exception.getBindingResult().getFieldError();

        if (fieldError != null) {
            String errorMessage = fieldError.getDefaultMessage();

            String detailedErrorMessage = String.format("%s", errorMessage);

            return new ResponseEntity<>(detailedErrorMessage, HttpStatus.BAD_REQUEST);
        }

        return new ResponseEntity<>("Ocorreu um erro.", HttpStatus.BAD_REQUEST);
    }

    @ExceptionHandler(UserAlreadyVerifiedException.class)
    public ResponseEntity<?>
    handleUserAlreadyVerifiedException(UserAlreadyVerifiedException e) {
        return new ResponseEntity<>(e.getMessage(), HttpStatus.BAD_REQUEST);
    }

    @ExceptionHandler(InvalidCredentialsException.class)
    public ResponseEntity<?> handleInvalidCredentialsException(InvalidCredentialsException
    e) {
        return new ResponseEntity<>(e.getMessage(), HttpStatus.UNAUTHORIZED);
    }

    // @Getter
    // public static class ErrorMessage {
    //     private String error;
    //
    //     public ErrorMessage(String error) {
    //         this.error = error;
    //     }
    //
    //     public void setErro(String error) {
    //         this.error = error;
    //     }
    // }

```

```
//    }
// }
}
```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\exception\InvalidCredentialsException.
java ----

```
package com.example.lixonoponto.exception;

import lombok.Getter;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.ResponseStatus;

@Getter
@ResponseStatus(value = HttpStatus.UNAUTHORIZED)
public class InvalidCredentialsException extends RuntimeException {
    public InvalidCredentialsException() {
        super("Credenciais incorretas.");
    }
}
```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\exception\ResourceNotFoundException.
n.java ----

```
package com.example.lixonoponto.exception;

import lombok.Getter;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.ResponseStatus;

@Getter
@ResponseStatus(value = HttpStatus.NOT_FOUND)
public class ResourceNotFoundException extends RuntimeException {
    private final String resourceName;
    private final String fieldName;
    private final Object fieldValue;

    public ResourceNotFoundException(String resourceName, String fieldName, Object
    fieldValue) {
        super(String.format("%s not found with %s : '%s'", resourceName, fieldName,
    fieldValue));
        this.resourceName = resourceName;
        this.fieldName = fieldName;
        this.fieldValue = fieldValue;
    }
}
```

```

    }
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\exception\UserAlreadyVerifiedException.java ----

```

package com.example.lixonoponto.exception;

public class UserAlreadyVerifiedException extends RuntimeException {
    public UserAlreadyVerifiedException() {
        super("O usuário já verificou este ponto de descarte");
    }

    public UserAlreadyVerifiedException(String message) {
        super(message);
    }
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\job\PontoDescarteJobService.java ----

```

package com.example.lixonoponto.job;

import com.example.lixonoponto.model.PontoDescarte;
import com.example.lixonoponto.service.PontoDescarteService;
import org.springframework.scheduling.annotation.Scheduled;
import org.springframework.stereotype.Service;

import java.time.ZonedDateTime;
import java.util.List;

@Service
public class PontoDescarteJobService {

    private final PontoDescarteService pontoDescarteService;

    public PontoDescarteJobService(PontoDescarteService pontoDescarteService) {
        this.pontoDescarteService = pontoDescarteService;
    }

    /**
     * Verifica se o ponto de descarte foi cadastrado a mais de 2 meses e se o número de
     * verificações é menor que 3, todo dia à meia noite.
     */
}

```

```

@Scheduled(cron = "0 0 0 * * ?")
public void verificarPontosDescarte() {
    List<PontoDescarte> pontosDescarte = pontoDescarteService.findAll();
    ZonedDateTime twoMonthsAgo = ZonedDateTime.now().minusMonths(2);

    for (PontoDescarte pontoDescarte : pontosDescarte) {
        if (pontoDescarte.getDataCadastro().isBefore(twoMonthsAgo) &&
            pontoDescarte.getCheckNumber() < 3) {
            pontoDescarteService.deletePontoDescarte(pontoDescarte.getId());
        }
    }
}
}
}
}

```

Arquivo: lixo-no-ponto\src\main\java\com\example\lixonoponto\LixoNoPontoApplication.java

```

package com.example.lixonoponto;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.scheduling.annotation.EnableScheduling;

@SpringBootApplication
@EnableScheduling
public class LixoNoPontoApplication {

    public static void main(String[] args) {
        SpringApplication.run(LixoNoPontoApplication.class, args);
    }

}
}

```

Arquivo: lixo-no-ponto\src\main\java\com\example\lixonoponto\model\ClasseDeUsuario.java

```

package com.example.lixonoponto.model;

import com.fasterxml.jackson.annotation.JsonBackReference;
import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;

import java.util.List;

```

```

@Getter
@Setter
@Entity
@Table(name = "classe_de_usuario")
public class ClasseDeUsuario {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "role_id_gen")
    @SequenceGenerator(name = "role_id_gen", sequenceName = "role_id_seq",
allocationSize = 1)
    @Column(name = "id", nullable = false)
    private Long id;

    @Column(name = "name", nullable = false)
    private String name;

    @JsonBackReference
    @OneToMany(mappedBy = "classeDeUsuario", fetch = FetchType.LAZY)
    private List<UsuarioClasseDeUsuario> usuarioClasseDeUsuarios;
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\model\dto\EmpresaReciclagemRequest.java ----

```
package com.example.lixonoponto.model.dto;
```

```
import lombok.Getter;
import lombok.Setter;
```

```
import java.util.List;
```

```

@Getter
@Setter
public class EmpresaReciclagemRequest {
    private String nome;
    private String contato;
    private String site;
    private String horarioFuncionamento;
    private Boolean patrocinador;
    private String foto;
    private List<Long> residuosIds;
}

```


Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\model\dto>LoginRequestDTO.java ----

```
package com.example.lixonoponto.model.dto;
```

```
import lombok.Getter;
```

```
import lombok.Setter;
```

```
@Getter
```

```
@Setter
```

```
public class LoginRequestDTO {
```

```
    private String email;
```

```
    private String password;
```

```
}
```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\model\dto\PontoDescarteDTO.java ----

```
package com.example.lixonoponto.model.dto;
```

```
import com.example.lixonoponto.model.TipoResiduo;
```

```
import lombok.Getter;
```

```
import lombok.Setter;
```

```
import java.util.List;
```

```
@Getter
```

```
@Setter
```

```
public class PontoDescarteDTO {
```

```
    private Long id;
```

```
    private String nome;
```

```
    private String foto;
```

```
    private String descricao;
```

```
    private String telefone;
```

```
    private Boolean verificado;
```

```
    private Double latitude;
```

```
    private Double longitude;
```

```
    private List<TipoResiduo> residuos;
```

```
}
```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\model\dto\PontoDescarteRequest.java

```

package com.example.lixonoponto.model.dto;

import lombok.Getter;
import lombok.Setter;

import java.util.List;

@Getter
@Setter
public class PontoDescarteRequest {
    private String nome;
    private String descricao;
    private String telefone;
    private String foto;
    private List<Long> residuosIds;
    private Double longitude;
    private Double latitude;
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\model\dto\PontoDescarteUpdateRequest.java ----

```

package com.example.lixonoponto.model.dto;

import lombok.Getter;
import lombok.Setter;

import java.util.List;

@Getter
@Setter
public class PontoDescarteUpdateRequest {
    private String nome;
    private String telefone;
    private String foto;
    private List<Long> residuosIds;
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\model\dto\RotaCaminhaoDTO.java ----

```

package com.example.lixonoponto.model.dto;

```

```
public class RotaCaminhaoDTO {
    private Integer id;
}
```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\model\dto\RotaCaminhaoRequest.java

```
package com.example.lixonoponto.model.dto;
```

```
import lombok.Getter;
import lombok.Setter;
```

```
import java.time.Instant;
```

```
@Getter
```

```
@Setter
```

```
public class RotaCaminhaoRequest {
    private Long idTipoCaminhao;
    private Instant horarioInicio;
    private Instant horarioFim;
    private String cidade;
    private Long idEmpresaReciclagem;
    private String ruasPassagem;
}
```

Arquivo: lixo-no-ponto\src\main\java\com\example\lixonoponto\model\dto\UsuarioDTO.java

```
package com.example.lixonoponto.model.dto;
```

```
import com.example.lixonoponto.model.ClasseDeUsuario;
import lombok.Getter;
import lombok.Setter;
```

```
import java.time.ZonedDateTime;
```

```
@Getter
```

```
@Setter
```

```
public class UsuarioDTO {
    private ZonedDateTime dataCadastro;
    private String nome;
    private String email;
    private Integer pontosVerificados;
    private ClasseDeUsuario classesDeUsuario;
}
```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\model\EmpresaReciclagem.java ----

```
package com.example.lixonoponto.model;

import jakarta.persistence.*;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Size;
import lombok.Getter;
import lombok.Setter;
import org.hibernate.annotations.ColumnDefault;

import java.util.*;

@Getter
@Setter
@Entity
@Table(name = "empresa_reciclagem")
public class EmpresaReciclagem {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"empresa_reciclagem_id_gen")
    @SequenceGenerator(name = "empresa_reciclagem_id_gen", sequenceName =
"empresa_reciclagem_id_seq", allocationSize = 1)
    @Column(name = "id", nullable = false)
    private Integer id;

    @Size(max = 255)
    @NotNull
    @Column(name = "nome", nullable = false)
    private String nome;

    @Size(max = 255)
    @NotNull
    @Column(name = "contato", nullable = false)
    private String contato;

    @Size(max = 255)
    @Column(name = "site")
    private String site;

    @ManyToMany(fetch = FetchType.LAZY)
    @JoinTable(name = "empresa_reciclagem_tipo_residuo",
        joinColumns = @JoinColumn(name = "empresa_reciclagem_id"),
        inverseJoinColumns = @JoinColumn(name = "tipo_residuo_id"))
```

```

private List<TipoResiduo> tipoResiduos = new ArrayList<>();

@Size(max = 255)
@Column(name = "horario_funcionamento")
private String horarioFuncionamento;

@Column(name = "foto")
private String foto;

@NotNull
@ColumnDefault("false")
@Column(name = "patrocinador", nullable = false)
private Boolean patrocinador = false;

@OneToMany(mappedBy = "idEmpresaReciclagem", fetch = FetchType.LAZY)
private Set<PontoDescarte> pontoDescartes = new LinkedHashSet<>();
//
// @OneToMany(mappedBy = "idEmpresaReciclagem", fetch = FetchType.LAZY)
// private Set<PontoDescarteEmpresaReciclagem> pontoDescarteEmpresaReciclagens =
new LinkedHashSet<>();

@OneToMany(mappedBy = "idEmpresaReciclagem", fetch = FetchType.LAZY)
private Set<RotaCaminhao> rotaCaminhoes = new LinkedHashSet<>();

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (!(o instanceof EmpresaReciclagem)) return false;
    EmpresaReciclagem that = (EmpresaReciclagem) o;
    return Objects.equals(id, that.id);
}

@Override
public int hashCode() {
    return Objects.hash(id);
}
}

```

Arquivo: lixo-no-ponto\src\main\java\com\example\lixonoponto\model\PontoDescarte.java

```

package com.example.lixonoponto.model;

import com.fasterxml.jackson.annotation.JsonBackReference;
import com.fasterxml.jackson.annotation.JsonIgnore;
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import com.fasterxml.jackson.annotation.JsonManagedReference;

```

```

import jakarta.persistence.*;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Size;
import lombok.Getter;
import lombok.Setter;
import org.hibernate.annotations.ColumnDefault;
import org.hibernate.annotations.CreationTimestamp;

import java.time.Instant;
import java.time.ZoneId;
import java.time.ZonedDateTime;
import java.util.LinkedHashSet;
import java.util.List;
import java.util.Objects;
import java.util.Set;

@Getter
@Setter
@Entity
@Table(name = "ponto_descarte")
@JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})
public class PontoDescarte {

    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"ponto_descarte_id_gen")
    @SequenceGenerator(name = "ponto_descarte_id_gen", sequenceName =
"ponto_descarte_id_seq", allocationSize = 1)
    @Column(name = "id", nullable = false)
    private Long id;

    @Size(max = 255)
    @NotNull
    @Column(name = "nome", nullable = false)
    private String nome;

    @Column(name = "telefone", nullable = false)
    private String telefone;

    @NotNull
    @Column(name = "latitude", nullable = false)
    private Double latitude;

    @NotNull
    @Column(name = "longitude", nullable = false)
    private Double longitude;

    @JsonManagedReference

```

```

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "id_status_verificacao")
private StatusVerificacao idStatusVerificacao;

@ColumnDefault("CURRENT_TIMESTAMP")
@Column(name = "data_cadastro", nullable = false, updatable = false)
@CreationTimestamp
private Instant dataCadastro;

@JsonBackReference
@NotNull
@ManyToOne(fetch = FetchType.LAZY, optional = false)
@JoinColumn(name = "id_usuario_cadastro", nullable = false)
private Usuario idUsuarioCadastro;

@Column(name = "foto")
private String foto;

@Column(name = "descricao", length = Integer.MAX_VALUE)
private String descricao;

@Column(name = "data_ultima_verificacao")
private Instant dataUltimaVerificacao;

@ManyToOne(fetch = FetchType.LAZY)
@JoinColumn(name = "id_empresa_reciclagem")
private EmpresaReciclagem idEmpresaReciclagem;

@JsonManagedReference
@OneToMany(mappedBy = "pontoDescarte", fetch = FetchType.LAZY)
private List<PontoDescarteTipoResiduo> pontoDescarteTipoResiduos;

@OneToMany(mappedBy = "idPontoDescarte", fetch = FetchType.LAZY)
private Set<PontoDescarteEmpresaReciclagem> pontoDescarteEmpresaReciclagems =
new LinkedHashSet<>();

@JsonIgnore
@OneToMany(mappedBy = "idPontoDescarte", fetch = FetchType.LAZY)
private Set<PontoDescarteUsuario> pontoDescarteUsuarios = new LinkedHashSet<>();

public PontoDescarte() {

}

public PontoDescarte(String nome, String descricao, Double latitude, Double longitude,
Instant dataUltimaVerificacao, Usuario idUsuarioCadastro, StatusVerificacao
idStatusVerificacao, String telefone, String foto) {
    this.nome = nome;

```

```

    this.descricao = descricao;
    this.latitude = latitude;
    this.longitude = longitude;
    this.dataUltimaVerificacao = dataUltimaVerificacao;
    this.idUsuarioCadastro = idUsuarioCadastro;
    this.idStatusVerificacao = idStatusVerificacao;
    this.telefone = telefone;
    this.foto = foto;
}

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (!(o instanceof PontoDescarte)) return false;
    PontoDescarte that = (PontoDescarte) o;
    return Objects.equals(id, that.id);
}

@Override
public int hashCode() {
    return Objects.hash(id);
}

public void updateStatusVerificacao(StatusVerificacao newStatus) {
    if (getCheckNumber() >= 3) {
        this.idStatusVerificacao = newStatus;
    }
}

public int getCheckNumber() {
    return pontoDescarteUsuarios.size();
}

public ZonedDateTime getDataCadastro() {
    return
dataCadastro.atZone(ZoneId.of("UTC")).withZoneSameInstant(ZoneId.of("America/Sao_Pau
lo"));
}

public ZonedDateTime getDataUltimaVerificacao() {
    return
dataUltimaVerificacao.atZone(ZoneId.of("UTC")).withZoneSameInstant(ZoneId.of("America/
Sao_Paulo"));
}
}

```


Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\model\PontoDescarteEmpresaReciclagem.java ----

```
package com.example.lixonoponto.model;
```

```
import jakarta.persistence.*;
```

```
import lombok.Getter;
```

```
import lombok.Setter;
```

```
import java.util.Objects;
```

```
@Getter
```

```
@Setter
```

```
@Entity
```

```
@Table(name = "ponto_descarte_empresa_reciclagem")
```

```
public class PontoDescarteEmpresaReciclagem {
```

```
    @Id
```

```
        @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"ponto_descarte_empresa_reciclagem_id_gen")
```

```
        @SequenceGenerator(name = "ponto_descarte_empresa_reciclagem_id_gen",
sequenceName = "ponto_descarte_empresa_reciclagem_id_seq", allocationSize = 1)
```

```
        @Column(name = "id", nullable = false)
```

```
        private Integer id;
```

```
        @ManyToOne(fetch = FetchType.LAZY)
```

```
        @JoinColumn(name = "id_ponto_descarte")
```

```
        private PontoDescarte idPontoDescarte;
```

```
        @ManyToOne(fetch = FetchType.LAZY)
```

```
        @JoinColumn(name = "id_empresa_reciclagem")
```

```
        private EmpresaReciclagem idEmpresaReciclagem;
```

```
@Override
```

```
public boolean equals(Object o) {
```

```
    if (this == o) return true;
```

```
    if (!(o instanceof PontoDescarteEmpresaReciclagem)) return false;
```

```
    PontoDescarteEmpresaReciclagem that = (PontoDescarteEmpresaReciclagem) o;
```

```
    return Objects.equals(id, that.id);
```

```
}
```

```
@Override
```

```
public int hashCode() {
```

```
    return Objects.hash(id);
```

```
}
```

```
}
```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\model\PontoDescarteTipoResiduo.java ----

```
package com.example.lixonoponto.model;

import com.fasterxml.jackson.annotation.JsonBackReference;
import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;

@Entity
@Getter
@Setter
@Table(name = "ponto_descarte_tipo_residuo")
public class PontoDescarteTipoResiduo {
    @JsonBackReference
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @JsonBackReference
    @ManyToOne
    @JoinColumn(name = "ponto_descarte_id")
    private PontoDescarte pontoDescarte;
    @ManyToOne
    @JoinColumn(name = "tipo_residuo_id")
    private TipoResiduo tipoResiduo;

    public PontoDescarteTipoResiduo(PontoDescarte pontoDescarte, TipoResiduo
tipoResiduo) {
        this.pontoDescarte = pontoDescarte;
        this.tipoResiduo = tipoResiduo;
    }

    public PontoDescarteTipoResiduo() {

    }
}
```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\model\PontoDescarteUsuario.java ----

```
package com.example.lixonoponto.model;

import jakarta.persistence.*;
import jakarta.validation.constraints.NotNull;
```

```

import lombok.Getter;
import lombok.Setter;

import java.util.Objects;

@Getter
@Setter
@Entity
@Table(name = "ponto_descarte_usuario")
public class PontoDescarteUsuario {
    @Id
        @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"ponto_descarte_usuario_id_gen")
        @SequenceGenerator(name = "ponto_descarte_usuario_id_gen", sequenceName =
"ponto_descarte_usuario_id_seq", allocationSize = 1)
        @Column(name = "id", nullable = false)
    private Integer id;

    @NotNull
    @ManyToOne(fetch = FetchType.LAZY, optional = false)
    @JoinColumn(name = "id_ponto_descarte", nullable = false)
    private PontoDescarte idPontoDescarte;

    @NotNull
    @ManyToOne(fetch = FetchType.LAZY, optional = false)
    @JoinColumn(name = "id_usuario", nullable = false)
    private Usuario idUsuario;

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof PontoDescarteUsuario)) return false;
        PontoDescarteUsuario that = (PontoDescarteUsuario) o;
        return Objects.equals(id, that.id);
    }

    @Override
    public int hashCode() {
        return Objects.hash(id);
    }
}

```

Arquivo: lixo-no-ponto\src\main\java\com\example\lixonoponto\model\RotaCaminhao.java

```
package com.example.lixonoponto.model;
```

```

import com.fasterxml.jackson.annotation.JsonBackReference;
import jakarta.persistence.*;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Size;
import lombok.Getter;
import lombok.Setter;

import java.time.Instant;
import java.util.Objects;

@Getter
@Setter
@Entity
@Table(name = "rota_caminhao")
public class RotaCaminhao {
    @Id
        @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"rota_caminhao_id_gen")
        @SequenceGenerator(name = "rota_caminhao_id_gen", sequenceName =
"rota_caminhao_id_seq", allocationSize = 1)
        @Column(name = "id", nullable = false)
    private Integer id;

    // @JsonBackReference
    @ManyToOne(fetch = FetchType.LAZY)
    @JoinColumn(name = "id_tipo_caminhao")
    private TipoCaminhao idTipoCaminhao;

    @NotNull
    @Column(name = "horario_inicio", nullable = false)
    private Instant horarioInicio;

    @NotNull
    @Column(name = "horario_fim", nullable = false)
    private Instant horarioFim;

    @Size(max = 255)
    @NotNull
    @Column(name = "cidade", nullable = false)
    private String cidade;

    @JsonBackReference
    @NotNull
    @ManyToOne(fetch = FetchType.LAZY, optional = false)
    @JoinColumn(name = "id_empresa_reciclagem", nullable = false)
    private EmpresaReciclagem idEmpresaReciclagem;

    @Column(name = "ruas_passagem", length = Integer.MAX_VALUE)

```

```

private String ruasPassagem;

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (!(o instanceof RotaCaminhao)) return false;
    RotaCaminhao that = (RotaCaminhao) o;
    return Objects.equals(id, that.id);
}

@Override
public int hashCode() {
    return Objects.hash(id);
}
}

```

Arquivo: lixo-no-ponto\src\main\java\com\example\lixonoponto\model>StatusVerificacao.java

```

package com.example.lixonoponto.model;

import com.fasterxml.jackson.annotation.JsonBackReference;
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import jakarta.persistence.*;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Size;
import lombok.Getter;
import lombok.Setter;

import java.util.LinkedHashSet;
import java.util.Objects;
import java.util.Set;

@Getter
@Setter
@Entity
@Table(name = "status_verificacao")
@JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})
public class StatusVerificacao {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"status_verificacao_id_gen")
    @SequenceGenerator(name = "status_verificacao_id_gen", sequenceName =
"status_verificacao_id_seq", allocationSize = 1)
    @Column(name = "id", nullable = false)
    private Integer id;
}

```

```

@Size(max = 255)
@NotNull
@Column(name = "nome", nullable = false)
private String nome;

@JsonBackReference
@OneToMany(mappedBy = "idStatusVerificacao", fetch = FetchType.LAZY)
private Set<PontoDescarte> pontoDescartes = new LinkedHashSet<>();

```

```

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (!(o instanceof StatusVerificacao)) return false;
    StatusVerificacao that = (StatusVerificacao) o;
    return Objects.equals(id, that.id);
}

```

```

@Override
public int hashCode() {
    return Objects.hash(id);
}
}

```

Arquivo: lixo-no-ponto\src\main\java\com\example\lixonoponto\model\TipoCaminhao.java ----

```

package com.example.lixonoponto.model;

import com.fasterxml.jackson.annotation.JsonIgnore;
import com.fasterxml.jackson.annotation.JsonIgnoreProperties;
import jakarta.persistence.*;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Size;
import lombok.Getter;
import lombok.Setter;

import java.util.LinkedHashSet;
import java.util.Objects;
import java.util.Set;

@Getter
@Setter
@Entity
@Table(name = "tipo_caminhao")
@JsonIgnoreProperties({"hibernateLazyInitializer", "handler"})
public class TipoCaminhao {

```

```

    @Id
        @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"tipo_caminhao_id_gen")
        @SequenceGenerator(name = "tipo_caminhao_id_gen", sequenceName =
"tipo_caminhao_id_seq", allocationSize = 1)
        @Column(name = "id", nullable = false)
        private Integer id;

    @Size(max = 255)
    @NotNull
    @Column(name = "nome", nullable = false)
    private String nome;

    @JsonIgnore
    @OneToMany(mappedBy = "idTipoCaminhao", fetch = FetchType.LAZY)
    private Set<RotaCaminhao> rotaCaminhoes = new LinkedHashSet<>();

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof TipoCaminhao)) return false;
        TipoCaminhao that = (TipoCaminhao) o;
        return Objects.equals(id, that.id);
    }

    @Override
    public int hashCode() {
        return Objects.hash(id);
    }
}

```

Arquivo: lixo-no-ponto\src\main\java\com\example\lixonoponto\model\TipoResiduo.java ----

```

package com.example.lixonoponto.model;

import com.fasterxml.jackson.annotation.JsonIgnore;
import jakarta.persistence.*;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Size;
import lombok.Getter;
import lombok.Setter;

import java.util.LinkedHashSet;
import java.util.Objects;
import java.util.Set;

```

```

@Getter
@Setter
@Entity
@Table(name = "tipo_residuo")
public class TipoResiduo {
    @Id
        @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"tipo_residuo_id_gen")
        @SequenceGenerator(name = "tipo_residuo_id_gen", sequenceName =
"tipo_residuo_id_seq", allocationSize = 1)
        @Column(name = "id", nullable = false)
        private Integer id;

        @Size(max = 255)
        @NotNull
        @Column(name = "nome", nullable = false)
        private String nome;

        @JsonIgnore
        @ManyToMany(mappedBy = "tipoResiduos", fetch = FetchType.LAZY)
        private Set<EmpresaReciclagem> empresaReciclagens = new LinkedHashSet<>();

// @OneToMany(mappedBy = "idTipoResiduo", fetch = FetchType.LAZY)
// private Set<PontoDescarte> pontoDescartes = new LinkedHashSet<>();

        @Override
        public boolean equals(Object o) {
            if (this == o) return true;
            if (!(o instanceof TipoResiduo)) return false;
            TipoResiduo that = (TipoResiduo) o;
            return Objects.equals(id, that.id);
        }

        @Override
        public int hashCode() {
            return Objects.hash(id);
        }
}

```

Arquivo: lixo-no-ponto\src\main\java\com\example\lixonoponto\model\Usuario.java ----

```

package com.example.lixonoponto.model;

import com.fasterxml.jackson.annotation.JsonBackReference;
import com.fasterxml.jackson.annotation.JsonIgnore;
import jakarta.persistence.*;
import jakarta.validation.constraints.NotNull;

```



```

import jakarta.validation.constraints.Size;
import lombok.Getter;
import lombok.Setter;
import org.hibernate.annotations.ColumnDefault;
import org.hibernate.annotations.CreationTimestamp;

import java.time.Instant;
import java.time.ZoneId;
import java.time.ZonedDateTime;
import java.util.LinkedHashSet;
import java.util.Objects;
import java.util.Set;

@Getter
@Setter
@Entity
@Table(name = "usuario")
public class Usuario {
    @JsonIgnore
    @Id
        @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"usuario_id_gen")
        @SequenceGenerator(name = "usuario_id_gen", sequenceName = "usuario_id_seq",
allocationSize = 1)
        @Column(name = "id", nullable = false)
    private Long id;

    @ColumnDefault("CURRENT_TIMESTAMP")
    @Column(name = "data_cadastro", nullable = false, updatable = false)
    @CreationTimestamp
    private Instant dataCadastro;

    @Size(max = 255)
    @NotNull
    @Column(name = "nome", nullable = false)
    private String nome;

    @Size(max = 255)
    @NotNull
    @Column(name = "email", nullable = false)
    private String email;

    @JsonBackReference
    @NotNull
    @Column(name = "senha", nullable = false)
    private String senha;

    @NotNull

```

```

@ColumnDefault("0")
@Column(name = "pontos_verificados", nullable = false)
private Integer pontosVerificados = 0;

@JsonIgnore
@OneToMany(mappedBy = "idUsuarioCadastro", fetch = FetchType.LAZY)
private Set<PontoDescarte> pontoDescartes = new LinkedHashSet<>();

@JsonIgnore
@OneToMany(mappedBy = "idUsuario", fetch = FetchType.LAZY)
private Set<PontoDescarteUsuario> pontoDescarteUsuarios = new LinkedHashSet<>();

// @JsonIgnore
@ManyToMany(fetch = FetchType.EAGER)
@JoinTable(
    name = "usuario_classe_de_usuario",
    joinColumns = @JoinColumn(name = "usuario_id"),
    inverseJoinColumns = @JoinColumn(name = "classe_de_usuario_id")
)
private Set<ClasseDeUsuario> classesDeUsuario;

@Override
public boolean equals(Object o) {
    if (this == o) return true;
    if (!(o instanceof Usuario)) return false;
    Usuario usuario = (Usuario) o;
    return Objects.equals(id, usuario.id);
}

@Override
public int hashCode() {
    return Objects.hash(id);
}

public ZonedDateTime getDataCadastro() {
    if (dataCadastro == null) {
        return
dataCadastro.atZone(ZoneId.of("UTC")).withZoneSameInstant(ZoneId.of("America/Sao_Pau
lo"));
    }
}
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\model\UsuarioClasseDeUsuario.java

```

package com.example.lixonoponto.model;

import jakarta.persistence.*;
import lombok.Getter;
import lombok.Setter;

@Getter
@Setter
@Entity
@Table(name = "usuario_classe_de_usuario")
public class UsuarioClasseDeUsuario {
    @Id
        @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"usuario_classe_de_usuario_id_gen")
        @SequenceGenerator(name = "usuario_classe_de_usuario_id_gen", sequenceName =
"usuario_classe_de_usuario_id_seq", allocationSize = 1)
        @Column(name = "id", nullable = false)
    private Long id;

    @ManyToOne
    @JoinColumn(name = "usuario_id", nullable = false)
    private Usuario usuario;

    @ManyToOne
    @JoinColumn(name = "classe_de_usuario_id", nullable = false)
    private ClasseDeUsuario classeDeUsuario;
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\repository\ClasseDeUsuarioRepository.java ----

```

package com.example.lixonoponto.repository;

import com.example.lixonoponto.model.ClasseDeUsuario;
import org.springframework.data.jpa.repository.JpaRepository;

public interface ClasseDeUsuarioRepository extends JpaRepository<ClasseDeUsuario,
Long> {
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\repository\EmpresaReciclagemRepository.java ----

```

package com.example.lixonoponto.repository;

import com.example.lixonoponto.model.EmpresaReciclagem;
import org.springframework.data.jpa.repository.JpaRepository;

public interface EmpresaReciclagemRepository extends
JpaRepository<EmpresaReciclagem, Long> {
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\repository\PontoDescarteEmpresaReciclagemRepository.java ----

```

package com.example.lixonoponto.repository;

import com.example.lixonoponto.model.PontoDescarte;
import com.example.lixonoponto.model.PontoDescarteEmpresaReciclagem;
import org.springframework.data.jpa.repository.JpaRepository;

public interface PontoDescarteEmpresaReciclagemRepository extends
JpaRepository<PontoDescarteEmpresaReciclagem, Long> {
    void deleteAllByIdPontoDescarte(PontoDescarte pontoDescarte);
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\repository\PontoDescarteRepository.java ----

```

package com.example.lixonoponto.repository;

import com.example.lixonoponto.model.PontoDescarte;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;

import java.util.List;

public interface PontoDescarteRepository extends JpaRepository<PontoDescarte, Long> {

    @Query("SELECT p FROM PontoDescarte p WHERE p.latitude BETWEEN :south AND :north AND p.longitude BETWEEN :west AND :east AND (:wastesId IS NULL OR p.id IN (SELECT pdtr.pontoDescarte.id FROM PontoDescarteTipoResiduo pdtr WHERE pdtr.tipoResiduo.id IN :wastesId))")

```

```

        List<PontoDescarte> findAllWithinBounds(@Param("south") Double south,
        @Param("north") Double north, @Param("west") Double west, @Param("east") Double east,
        @Param("wastesId") List<Long> wastesId);
    }

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\repository\PontoDescarteTipoResiduo
Repository.java ----

```
package com.example.lixonoponto.repository;
```

```
import com.example.lixonoponto.model.PontoDescarte;
import com.example.lixonoponto.model.PontoDescarteTipoResiduo;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
```

```
@Repository
public interface PontoDescarteTipoResiduoRepository extends
JpaRepository<PontoDescarteTipoResiduo, Long> {
    void deleteAllByPontoDescarte(PontoDescarte pontoDescarte);
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\repository\PontoDescarteUsuarioRepo
sitory.java ----

```
package com.example.lixonoponto.repository;
```

```
import com.example.lixonoponto.model.PontoDescarte;
import com.example.lixonoponto.model.PontoDescarteUsuario;
import org.springframework.data.jpa.repository.JpaRepository;
```

```
public interface PontoDescarteUsuarioRepository extends
JpaRepository<PontoDescarteUsuario, Long> {
    void deleteAllByIdPontoDescarte(PontoDescarte idPontoDescarte);
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\repository\RotaCaminhaoRepository.ja
va ----

```
package com.example.lixonoponto.repository;
```

```

import com.example.lixonoponto.model.RotaCaminhao;
import jakarta.validation.constraints.NotNull;
import jakarta.validation.constraints.Size;
import org.springframework.data.jpa.repository.JpaRepository;

import java.util.List;

public interface RotaCaminhaoRepository extends JpaRepository<RotaCaminhao, Long> {
    List<RotaCaminhao> findByCidadeContainingAndIdTipoCaminhao_Id(@Size(max = 255)
@NotNull String cidade, Integer idTipoCaminhao_id);

    List<RotaCaminhao> findByCidadeContaining(String cidade);
}

```

Arquivo:

```

lixo-no-ponto\src\main\java\com\example\lixonoponto\repository>StatusVerificacaoRepository
.java ----

```

```

package com.example.lixonoponto.repository;

import com.example.lixonoponto.model.StatusVerificacao;
import org.springframework.data.jpa.repository.JpaRepository;

public interface StatusVerificacaoRepository extends JpaRepository<StatusVerificacao,
Long> {
}

```

Arquivo:

```

lixo-no-ponto\src\main\java\com\example\lixonoponto\repository\TipoCaminhaoRepository.ja
va ----

```

```

package com.example.lixonoponto.repository;

import com.example.lixonoponto.model.TipoCaminhao;
import org.springframework.data.jpa.repository.JpaRepository;

public interface TipoCaminhaoRepository extends JpaRepository<TipoCaminhao, Long> {
}

```

Arquivo:

```

lixo-no-ponto\src\main\java\com\example\lixonoponto\repository\TipoResiduoRepository.java
----

```

```

package com.example.lixonoponto.repository;

import com.example.lixonoponto.model.TipoResiduo;
import org.springframework.data.jpa.repository.JpaRepository;

public interface TipoResiduoRepository extends JpaRepository<TipoResiduo, Long> {
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\repository\UsuarioClasseDeUsuarioRepository.java ----

```

package com.example.lixonoponto.repository;

import com.example.lixonoponto.model.UsuarioClasseDeUsuario;
import org.springframework.data.jpa.repository.JpaRepository;

public interface UsuarioClasseDeUsuarioRepository extends
JpaRepository<UsuarioClasseDeUsuario, Long> {
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\repository\UsuarioRepository.java ----

```

package com.example.lixonoponto.repository;

import com.example.lixonoponto.model.Usuario;
import org.springframework.data.jpa.repository.JpaRepository;

import java.util.Optional;

public interface UsuarioRepository extends JpaRepository<Usuario, Long> {
    Optional<Usuario> findByEmail(String email);
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\service\EmpresaReciclagemService.java ----

```

package com.example.lixonoponto.service;

```

```

import com.example.lixonoponto.model.EmpresaReciclagem;
import com.example.lixonoponto.model.TipoResiduo;
import com.example.lixonoponto.model.dto.EmpresaReciclagemRequest;
import com.example.lixonoponto.repository.EmpresaReciclagemRepository;
import com.example.lixonoponto.repository.TipoResiduoRepository;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.Optional;

@Service
public class EmpresaReciclagemService {

    private final EmpresaReciclagemRepository empresaReciclagemRepository;
    private final TipoResiduoRepository tipoResiduoRepository;

    public EmpresaReciclagemService(EmpresaReciclagemRepository
empresaReciclagemRepository,
    TipoResiduoRepository tipoResiduoRepository) {
        this.empresaReciclagemRepository = empresaReciclagemRepository;
        this.tipoResiduoRepository = tipoResiduoRepository;
    }

    public List<EmpresaReciclagem> findAll() {
        return empresaReciclagemRepository.findAll();
    }

    public Optional<EmpresaReciclagem> findById(Long id) {
        return empresaReciclagemRepository.findById(id);
    }

    public EmpresaReciclagem save(EmpresaReciclagemRequest
empresaReciclagemRequest) {

        EmpresaReciclagem empresaReciclagem = new EmpresaReciclagem();
        empresaReciclagem.setNome(empresaReciclagemRequest.getNome());
        empresaReciclagem.setContato(empresaReciclagemRequest.getContato());
        empresaReciclagem.setSite(empresaReciclagemRequest.getSite());

        empresaReciclagem.setHorarioFuncionamento(empresaReciclagemRequest.getHorarioFun
cionamiento());
        empresaReciclagem.setPatrocinador(empresaReciclagemRequest.getPatrocinador());
        empresaReciclagem.setFoto(empresaReciclagemRequest.getFoto());
        List<TipoResiduo> residuos =
tipoResiduoRepository.findAllById(empresaReciclagemRequest.getResiduosIds());
        empresaReciclagem.setTipoResiduos(residuos);
        return empresaReciclagemRepository.save(empresaReciclagem);
    }
}

```



```

    public void deleteById(Long id) {
        empresaReciclagemRepository.deleteById(id);
    }
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\service\PontoDescarteEmpresaReciclagemService.java ----

```

package com.example.lixonoponto.service;

```

```

import com.example.lixonoponto.model.PontoDescarteEmpresaReciclagem;
import com.example.lixonoponto.repository.PontoDescarteEmpresaReciclagemRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

```

```

import java.util.List;
import java.util.Optional;

```

```

@Service

```

```

public class PontoDescarteEmpresaReciclagemService {

```

```

    private final PontoDescarteEmpresaReciclagemRepository repository;

```

```

    @Autowired

```

```

    public PontoDescarteEmpresaReciclagemService(PontoDescarteEmpresaReciclagemRepository repository) {
        this.repository = repository;
    }

```

```

    public List<PontoDescarteEmpresaReciclagem> findAll() {
        return repository.findAll();
    }

```

```

    public Optional<PontoDescarteEmpresaReciclagem> findById(Long id) {
        return repository.findById(id);
    }

```

```

    public PontoDescarteEmpresaReciclagem save(PontoDescarteEmpresaReciclagem pontoDescarteEmpresaReciclagem) {
        return repository.save(pontoDescarteEmpresaReciclagem);
    }

```

```

    public void deleteById(Long id) {

```

```

        repository.deleteById(id);
    }
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\service\PontoDescarteService.java ----

```
package com.example.lixonoponto.service;
```

```

import com.example.lixonoponto.exception.ResourceNotFoundException;
import com.example.lixonoponto.exception.UserAlreadyVerifiedException;
import com.example.lixonoponto.model.*;
import com.example.lixonoponto.model.dto.PontoDescarteDTO;
import com.example.lixonoponto.model.dto.PontoDescarteRequest;
import com.example.lixonoponto.model.dto.PontoDescarteUpdateRequest;
import com.example.lixonoponto.repository.*;
import jakarta.persistence.EntityManager;
import jakarta.transaction.Transactional;
import org.springframework.stereotype.Service;

```

```

import java.time.Instant;
import java.time.LocalDateTime;
import java.time.ZoneOffset;
import java.util.List;
import java.util.Optional;
import java.util.stream.Collectors;

```

```
@Service
```

```
public class PontoDescarteService {
```

```

    private final PontoDescarteRepository pontoDescarteRepository;
    private final PontoDescarteTipoResiduoRepository pontoDescarteTipoResiduoRepository;
    private final PontoDescarteUsuarioRepository pontoDescarteUsuarioRepository;
    private final PontoDescarteEmpresaReciclagemRepository
pontoDescarteEmpresaRepository;
    private final TipoResiduoRepository tipoResiduoRepository;
    private final StatusVerificacaoService statusVerificacaoService;
    private final UsuarioService usuarioService;
    private final EntityManager entityManager;

```

```

    public PontoDescarteService(PontoDescarteRepository pontoDescarteRepository,
        TipoResiduoRepository tipoResiduoRepository,
        UsuarioService usuarioService,
        PontoDescarteUsuarioRepository pontoDescarteUsuarioRepository,
        StatusVerificacaoService statusVerificacaoService,
        EntityManager entityManager,

```

```

        PontoDescarteEmpresaReciclagemRepository
pontoDescarteEmpresaRepository,
        PontoDescarteTipoResiduoRepository
pontoDescarteTipoResiduoRepository) {
    this.pontoDescarteRepository = pontoDescarteRepository;
    this.tipoResiduoRepository = tipoResiduoRepository;
    this.usuarioService = usuarioService;
    this.statusVerificacaoService = statusVerificacaoService;
    this.pontoDescarteTipoResiduoRepository = pontoDescarteTipoResiduoRepository;
    this.pontoDescarteUsuarioRepository = pontoDescarteUsuarioRepository;
    this.pontoDescarteEmpresaRepository = pontoDescarteEmpresaRepository;
    this.entityManager = entityManager;
}

@Transactional
    public PontoDescarte createPontoDescarte(PontoDescarteRequest
pontoDescarteRequest, Long userId) {
    Usuario usuario = usuarioService.getUsuarioById(userId)
        .orElseThrow(() -> new RuntimeException("Usuário não encontrado"));

        StatusVerificacao statusVerificacao =
statusVerificacaoService.getStatusVerificacaoById(2L)
        .orElseThrow(() -> new RuntimeException("Status de verificação não
encontrado"));

    PontoDescarte pontoDescarte = new PontoDescarte(
        pontoDescarteRequest.getNome(),
        pontoDescarteRequest.getDescricao(),
        pontoDescarteRequest.getLatitude(),
        pontoDescarteRequest.getLongitude(),
        LocalDateTime.now().toInstant(ZoneOffset.of("-03:00")),
        usuario,
        statusVerificacao,
        pontoDescarteRequest.getTelefone(),
        pontoDescarteRequest.getFoto()
    );

    pontoDescarte = pontoDescarteRepository.save(pontoDescarte);

        List<TipoResiduo> residuos =
tipoResiduoRepository.findAllById(pontoDescarteRequest.getResiduosIds());
    PontoDescarte finalPontoDescarte = pontoDescarte;
    List<PontoDescarteTipoResiduo> pdtrs = residuos.stream()
        .map(residuo -> new PontoDescarteTipoResiduo(finalPontoDescarte, residuo))
        .collect(Collectors.toList());

    pontoDescarteTipoResiduoRepository.saveAll(pdtrs);

```

```

PontoDescarteUsuario pontoDescarteUsuario = new PontoDescarteUsuario();
pontoDescarteUsuario.setIdPontoDescarte(pontoDescarte);
pontoDescarteUsuario.setIdUsuario(usuario);
pontoDescarteUsuarioRepository.save(pontoDescarteUsuario);

entityManager.flush();
entityManager.refresh(pontoDescarte);

return pontoDescarte;
}

public Optional<PontoDescarte> getPontoDescarteById(Long id) {
    return pontoDescarteRepository.findById(id);
}

public List<PontoDescarteDTO> getAllPontoDescarte(Double south, Double north,
Double west, Double east, List<Long> wastesId) {
    return pontoDescarteRepository.findAllWithinBounds(south, north, west, east,
wastesId).stream()
        .map(pontoDescarte -> {
            PontoDescarteDTO dto = new PontoDescarteDTO();
            dto.setNome(pontoDescarte.getNome());
            dto.setId(pontoDescarte.getId());
            dto.setFoto(pontoDescarte.getFoto());
            dto.setVerificado(pontoDescarte.getIdStatusVerificacao().getId() == 1);
            dto.setDescricao(pontoDescarte.getDescricao());
            dto.setTelefone(pontoDescarte.getTelefone());
            dto.setLatitude(pontoDescarte.getLatitude());
            dto.setLongitude(pontoDescarte.getLongitude());
            dto.setResiduos(pontoDescarte.getPontoDescarteTipoResiduos().stream()
                .map(PontoDescarteTipoResiduo::getTipoResiduo)
                .collect(Collectors.toList()));
            return dto;
        })
        .collect(Collectors.toList());
}

@Transactional
public PontoDescarte updatePontoDescarte(Long id, PontoDescarteUpdateRequest
request) {
    PontoDescarte pontoDescarte = pontoDescarteRepository.findById(id)
        .orElseThrow(() -> new ResourceNotFoundException("PontoDescarte", "id", id));

    if (request.getNome() != null) {
        pontoDescarte.setNome(request.getNome());
    }

    if (request.getTelefone() != null) {

```

```

        pontoDescarte.setTelefone(request.getTelefone());
    }

    if (request.getFoto() != null) {
        pontoDescarte.setFoto(request.getFoto());
    }

    if (request.getResiduosIds() != null) {
        List<TipoResiduo> novosResiduos =
tipoResiduoRepository.findAllById(request.getResiduosIds());

        List<PontoDescarteTipoResiduo> pdtrsToRemove =
pontoDescarte.getPontoDescarteTipoResiduos().stream()
            .filter(pdtr -> !novosResiduos.contains(pdtr.getTipoResiduo()))
            .collect(Collectors.toList());

        pontoDescarte.getPontoDescarteTipoResiduos().removeAll(pdtrsToRemove);
        pontoDescarteTipoResiduoRepository.deleteAll(pdtrsToRemove);

        for (TipoResiduo novoResiduo : novosResiduos) {
            if (pontoDescarte.getPontoDescarteTipoResiduos().stream().noneMatch(pdtr ->
pdtr.getTipoResiduo().equals(novoResiduo))) {
                PontoDescarteTipoResiduo newPdtr = new
PontoDescarteTipoResiduo(pontoDescarte, novoResiduo);
                pontoDescarte.getPontoDescarteTipoResiduos().add(newPdtr);
                pontoDescarteTipoResiduoRepository.save(newPdtr);
            }
        }
    }

    return pontoDescarteRepository.save(pontoDescarte);
}

public PontoDescarte savePontoDescarte(PontoDescarte pontoDescarte) {
    return pontoDescarteRepository.save(pontoDescarte);
}

@Transactional
public void deletePontoDescarte(Long id) {
    PontoDescarte pontoDescarte = pontoDescarteRepository.findById(id)
        .orElseThrow(() -> new ResourceNotFoundException("PontoDescarte", "id", id));

    pontoDescarteUsuarioRepository.deleteAllByIdPontoDescarte(pontoDescarte);

    pontoDescarteTipoResiduoRepository.deleteAllByPontoDescarte(pontoDescarte);

    pontoDescarteEmpresaRepository.deleteAllByIdPontoDescarte(pontoDescarte);
}

```

```

        pontoDescarteRepository.deleteById(id);
    }

    public PontoDescarte updateStatusVerificacao(Long pontoDescarteId, StatusVerificacao
novoStatus) {
        PontoDescarte pontoDescarte = pontoDescarteRepository.findById(pontoDescarteId)
            .orElseThrow(() -> new ResourceNotFoundException("PontoDescarte", "id",
pontoDescarteId));

        pontoDescarte.updateStatusVerificacao(novoStatus);

        return pontoDescarteRepository.save(pontoDescarte);
    }

    @Transactional
    public PontoDescarte addUsuarioInPontoDescarte(Long pontoDescarteId, Long
usuarioid) {
        PontoDescarte pontoDescarte = getPontoDescarteById(pontoDescarteId)
            .orElseThrow(() -> new ResourceNotFoundException("PontoDescarte", "id",
pontoDescarteId));

        Usuario usuario = usuarioService.getUsuarioById(usuarioid)
            .orElseThrow(() -> new ResourceNotFoundException("Usuário", "id", usuarioid));

        boolean usuarioAlreadyVerify = pontoDescarte.getPontoDescarteUsuarios().stream()
            .anyMatch(pdu -> pdu.getIdUsuario().getId().equals(usuario.getId()));

        if (usuarioAlreadyVerify) {
            throw new UserAlreadyVerifiedException();
        }

        PontoDescarteUsuario pontoDescarteUsuario = new PontoDescarteUsuario();
        pontoDescarteUsuario.setIdPontoDescarte(pontoDescarte);
        pontoDescarteUsuario.setIdUsuario(usuario);

        pontoDescarteUsuarioRepository.save(pontoDescarteUsuario);

        usuario.setPontosVerificados(usuario.getPontosVerificados() + 1);
        usuarioService.updateUsuario(usuario);

        entityManager.flush();
        entityManager.refresh(pontoDescarte);

        pontoDescarte.setDataUltimaVerificacao(Instant.now());

        if (pontoDescarte.getCheckNumber() >= 3) {
            StatusVerificacao statusVerificacao =
statusVerificacaoService.getStatusVerificacaoById(1L)

```

```

        .orElseThrow(() -> new ResourceNotFoundException("StatusVerificacao", "id",
1L));
        pontoDescarte.setIdStatusVerificacao(statusVerificacao);
        pontoDescarte = savePontoDescarte(pontoDescarte);
    }

    return pontoDescarte;
}

public List<PontoDescarte> findAll() {
    return this.pontoDescarteRepository.findAll();
}
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\service\PontoDescarteUsuarioService.
java ----

```
package com.example.lixonoponto.service;
```

```
import com.example.lixonoponto.model.PontoDescarte;
import com.example.lixonoponto.repository.PontoDescarteRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
@Service
```

```
public class PontoDescarteUsuarioService {
```

```
    private final PontoDescarteRepository pontoDescarteRepository;
```

```
    @Autowired
```

```
    public PontoDescarteUsuarioService(PontoDescarteRepository pontoDescarteRepository)
    {
        this.pontoDescarteRepository = pontoDescarteRepository;
    }

```

```
    public PontoDescarte adicionarPontoDescarte(PontoDescarte pontoDescarte) {
```

```
        // Implementar lógica para adicionar um ponto de descarte
        return pontoDescarteRepository.save(pontoDescarte);
    }

```

```
    public PontoDescarte verificarPontoDescarte(Long id) {
```

```
        // Implementar lógica para verificar um ponto de descarte
        PontoDescarte pontoDescarte = pontoDescarteRepository.findById(id).orElseThrow(()
-> new RuntimeException("Ponto de descarte não encontrado"));
        // pontoDescarte.setVerificado(true);
        return pontoDescarteRepository.save(pontoDescarte);
    }

```

```

    }

    public void removerPontoDescarte(Long id) {
        // Implementar lógica para remover um ponto de descarte
        pontoDescarteRepository.deleteById(id);
    }
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\service\RotaCaminhaoService.java ----

```
package com.example.lixonoponto.service;
```

```
import com.example.lixonoponto.exception.ResourceNotFoundException;
import com.example.lixonoponto.model.RotaCaminhao;
import com.example.lixonoponto.model.dto.RotaCaminhaoRequest;
import com.example.lixonoponto.repository.EmpresaReciclagemRepository;
import com.example.lixonoponto.repository.RotaCaminhaoRepository;
import com.example.lixonoponto.repository.TipoCaminhaoRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
```

```
import java.util.List;
```

```
@Service
```

```
public class RotaCaminhaoService {
```

```
    private final RotaCaminhaoRepository rotaCaminhaoRepository;
    private final TipoCaminhaoRepository tipoCaminhaoRepository;
    private final EmpresaReciclagemRepository empresaReciclagemRepository;
```

```
@Autowired
```

```
    public RotaCaminhaoService(RotaCaminhaoRepository rotaCaminhaoRepository,
        TipoCaminhaoRepository tipoCaminhaoRepository, EmpresaReciclagemRepository
        empresaReciclagemRepository) {
        this.rotaCaminhaoRepository = rotaCaminhaoRepository;
        this.tipoCaminhaoRepository = tipoCaminhaoRepository;
        this.empresaReciclagemRepository = empresaReciclagemRepository;
    }
```

```
        public RotaCaminhao createRotaCaminhao(RotaCaminhaoRequest
        rotaCaminhaoRequest) {
            RotaCaminhao rotaCaminhao = new RotaCaminhao();
```

```
            rotaCaminhao.setTipoCaminhao(tipoCaminhaoRepository.findById(rotaCaminhaoRequest.
            getTipoCaminhaoId()))
```



```

        .orElseThrow(() -> new ResourceNotFoundException("TipoCaminhao", "id",
rotaCaminhaoRequest.getIdTipoCaminhao()));
        rotaCaminhao.setHorarioInicio(rotaCaminhaoRequest.getHorarioInicio());
        rotaCaminhao.setHorarioFim(rotaCaminhaoRequest.getHorarioFim());
        rotaCaminhao.setCidade(rotaCaminhaoRequest.getCidade());

rotaCaminhao.setIdEmpresaReciclagem(empresaReciclagemRepository.findById(rotaCaminhaoRequest.getIdEmpresaReciclagem())
        .orElseThrow(() -> new ResourceNotFoundException("EmpresaReciclagem", "id ",
rotaCaminhaoRequest.getIdEmpresaReciclagem())));
        rotaCaminhao.setRuasPassagem(rotaCaminhaoRequest.getRuasPassagem());
        return rotaCaminhaoRepository.save(rotaCaminhao);
    }

    public RotaCaminhao updateRotaCaminhao(RotaCaminhao rotaCaminhao) {
        return rotaCaminhaoRepository.save(rotaCaminhao);
    }

    public void deleteRotaCaminhao(Long id) {
        rotaCaminhaoRepository.deleteById(id);
    }

    public List<RotaCaminhao> getAllRotasCaminhao() {
        return rotaCaminhaoRepository.findAll();
    }

    public RotaCaminhao getRotaCaminhaoById(Long id) {
        return rotaCaminhaoRepository.findById(id).orElse(null);
    }

    public List<RotaCaminhao> getRotasCaminhaoByCidadeAndTipoCaminhao(String cidade, Integer idTipoCaminhao) {
        if (idTipoCaminhao != null) {
            return
rotaCaminhaoRepository.findByCidadeContainingAndIdTipoCaminhao_Id(cidade,
idTipoCaminhao);
        } else {
            return rotaCaminhaoRepository.findByCidadeContaining(cidade);
        }
    }
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\service>StatusVerificacaoService.java

```
package com.example.lixonoponto.service;
```

```

import com.example.lixonoponto.model.StatusVerificacao;
import com.example.lixonoponto.repository.StatusVerificacaoRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.Optional;

@Service
public class StatusVerificacaoService {

    private final StatusVerificacaoRepository statusVerificacaoRepository;

    @Autowired
    public StatusVerificacaoService(StatusVerificacaoRepository statusVerificacaoRepository)
    {
        this.statusVerificacaoRepository = statusVerificacaoRepository;
    }

    public Optional<StatusVerificacao> getStatusVerificacaoById(Long id) {
        return statusVerificacaoRepository.findById(id);
    }
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\service\TipoCaminhaoService.java ---

```

package com.example.lixonoponto.service;

import com.example.lixonoponto.model.TipoCaminhao;
import com.example.lixonoponto.repository.TipoCaminhaoRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class TipoCaminhaoService {

    private final TipoCaminhaoRepository tipoCaminhaoRepository;

    @Autowired
    public TipoCaminhaoService(TipoCaminhaoRepository tipoCaminhaoRepository) {
        this.tipoCaminhaoRepository = tipoCaminhaoRepository;
    }

    public TipoCaminhao createTipoCaminhao(TipoCaminhao tipoResiduo) {

```

```

        return tipoCaminhaoRepository.save(tipoResiduo);
    }

    public TipoCaminhao updateTipoCaminhao(TipoCaminhao tipoResiduo) {
        return tipoCaminhaoRepository.save(tipoResiduo);
    }

    public void deleteTipoCaminhao(Long id) {
        tipoCaminhaoRepository.deleteById(id);
    }

    public List<TipoCaminhao> getAllTipoCaminhao() {
        return tipoCaminhaoRepository.findAll();
    }

    public TipoCaminhao getTipoCaminhaoById(Long id) {
        return tipoCaminhaoRepository.findById(id).orElse(null);
    }
}

```

Arquivo:

lixo-no-ponto\src\main\java\com\example\lixonoponto\service\TipoResiduoService.java ----

```

package com.example.lixonoponto.service;

import com.example.lixonoponto.model.TipoResiduo;
import com.example.lixonoponto.repository.TipoResiduoRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class TipoResiduoService {

    private final TipoResiduoRepository tipoResiduoRepository;

    @Autowired
    public TipoResiduoService(TipoResiduoRepository tipoResiduoRepository) {
        this.tipoResiduoRepository = tipoResiduoRepository;
    }

    public TipoResiduo createTipoResiduo(TipoResiduo tipoResiduo) {
        return tipoResiduoRepository.save(tipoResiduo);
    }

    public TipoResiduo updateTipoResiduo(TipoResiduo tipoResiduo) {

```

```

        return tipoResiduoRepository.save(tipoResiduo);
    }

    public void deleteTipoResiduo(Long id) {
        tipoResiduoRepository.deleteById(id);
    }

    public List<TipoResiduo> getAllTipoResiduo() {
        return tipoResiduoRepository.findAll();
    }

    public TipoResiduo getTipoResiduoById(Long id) {
        return tipoResiduoRepository.findById(id).orElse(null);
    }
}

```

Arquivo: lixo-no-ponto\src\main\java\com\example\lixonoponto\service\TokenService.java ----

```

package com.example.lixonoponto.service;

import io.jsonwebtoken.Claims;
import io.jsonwebtoken.Jwts;
import io.jsonwebtoken.SignatureAlgorithm;
import io.jsonwebtoken.io.Decoders;
import io.jsonwebtoken.security.Keys;
import org.apache.tomcat.util.codec.binary.Base64;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.data.util.Pair;
import org.springframework.stereotype.Service;

import java.util.Date;
import java.util.HashMap;
import java.util.Map;
import java.util.Set;

@Service
public class TokenService {

    @Value("${jwt.secretKey}")
    private String secret;

    @Value("${jwt.expiration}")
    private String expiration;

    public boolean isTokenValid(String token) {
        try {

```

```

Jwts.parser().verifyWith(Keys.hmacShaKeyFor(Decoders.BASE64.decode(secret))).build().p
arseClaimsJws(token);
    return true;
} catch (Exception e) {
    return false;
}
}

public String getTokenByRequest(String authorizationHeader) {
    return authorizationHeader.substring(7);
}

public Long getUserId(String token) {
    Claims body =
Jwts.parser().setSigningKey(secret).build().parseClaimsJws(token).getBody();
    return Long.parseLong(body.getSubject());
}

public Pair<String, String> getUsuarioSenha(String token) {
    String[] decodedToken = (new String(Base64.decodeBase64(token))).split(":");
    return Pair.of(decodedToken[0], decodedToken[1]);
}

public String generateToken(Long id, String userEmail, Set<String> authorities) {
    Map<String, Object> claims = new HashMap<>();
    claims.put("authorities", authorities);

    Date today = new Date();
    Date dateExpiration = new Date(today.getTime() + Long.parseLong(expiration));

    return Jwts.builder()
        .setIssuer(userEmail)
        .setSubject(id.toString())
        .setIssuedAt(today)
        .setExpiration(dateExpiration)
        .signWith(SignatureAlgorithm.HS256, secret)
        .addClaims(claims)
        .compact();
}
}

```

Arquivo: lixo-no-ponto\src\main\java\com\example\lixonoponto\service\UsuarioService.java

```
package com.example.lixonoponto.service;
```

```
import com.example.lixonoponto.exception.ResourceNotFoundException;
import com.example.lixonoponto.model.Usuario;
import com.example.lixonoponto.repository.UsuarioRepository;
import org.springframework.stereotype.Service;
```

```
import java.util.List;
import java.util.Optional;
```

```
@Service
```

```
public class UsuarioService {
```

```
    private final UsuarioRepository usuarioRepository;
```

```
    public UsuarioService(UsuarioRepository usuarioRepository) {
        this.usuarioRepository = usuarioRepository;
    }
```

```
    public List<Usuario> getUsuarios() {
        return usuarioRepository.findAll();
    }
```

```
    public Optional<Usuario> getUsuarioById(Long id) {
        return usuarioRepository.findById(id);
    }
```

```
    public Optional<Usuario> getUsuarioByEmail(String email) {
        return usuarioRepository.findByEmail(email);
    }
```

```
    public Usuario addUsuario(Usuario usuario) {
        if (usuario.getNome() == null || usuario.getNome().trim().isEmpty()) {
            throw new RuntimeException("O nome do usuário não pode estar vazio.");
        }
```

```
        if (!usuario.getEmail().matches("^[A-Za-z0-9+_.-]+@(.+)$")) {
            throw new RuntimeException("O e-mail informado não é válido.");
        }
```

```
        Optional<Usuario> usuarioExistente =
usuarioRepository.findByEmail(usuario.getEmail());
        if (usuarioExistente.isPresent()) {
            throw new RuntimeException("Dados já cadastrados no sistema.");
        }

        return usuarioRepository.save(usuario);
    }
```

```

public Usuario updateUsuario(Usuario usuario) {
    Optional<Usuario> existingUsuario = usuarioRepository.findById(usuario.getId());
    if (existingUsuario.isPresent()) {
        return usuarioRepository.save(usuario);
    } else {
        throw new ResourceNotFoundException("User", "id", usuario.getId());
    }
}

public void deleteUsuario(Long id) {
    usuarioRepository.deleteById(id);
}
}

```

Arquivo: lixo-no-ponto\src\main\resources\application.properties ----

```

spring.application.name=lixo-no-ponto
spring.datasource.url=jdbc:${DATABASE_URL}
spring.datasource.username=${USER}
spring.datasource.password=${PASSWORD}
jwt.secretKey=${JWT_SECRET}
jwt.expiration=${JWT_EXPIRATION}
logging.level.org.springframework.security=INFO
spring.jpa.show-sql=true
logging.level.org.springframework=DEBUG
#logging.level.root=TRACE

```

Arquivo: lixo-no-ponto\src\main\resources\db\migration\V1.0.0__create_table_usuario.sql ----

```

CREATE TABLE IF NOT EXISTS usuario (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL UNIQUE,
    senha VARCHAR(255) NOT NULL,
    pontos_verificados INTEGER NOT NULL DEFAULT 0
);

```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.10__insert_into_status_verificacao.sql

```

INSERT INTO status_verificacao (nome) VALUES
    ('verificado'),
    ('nao_verificado');

```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.11__insert_into_tipo_caminhao.sql ----

```
INSERT INTO tipo_caminhao (nome) VALUES
    ('normal'),
    ('reciclável');
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.12__create_table_ponto_descarte_tipo_residuo.sql ----

```
CREATE TABLE ponto_descarte_tipo_residuo
(
    id          BIGSERIAL PRIMARY KEY,
    ponto_descarte_id BIGINT NOT NULL,
    tipo_residuo_id  BIGINT NOT NULL,
    FOREIGN KEY (ponto_descarte_id) REFERENCES ponto_descarte (id),
    FOREIGN KEY (tipo_residuo_id) REFERENCES tipo_residuo (id)
);
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.13__alter_table_ponto_descarte.sql ----

```
ALTER TABLE ponto_descarte
    DROP COLUMN IF EXISTS id_tipo_residuo;
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.14__update_ponto_descarte.sql ----

```
ALTER TABLE ponto_descarte
    ADD COLUMN telefone VARCHAR(255);
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.15__create_table_empresa_reciclagem_tipo_residuo.sql ----

```
CREATE TABLE empresa_reciclagem_tipo_residuo
(
    empresa_reciclagem_id BIGINT NOT NULL,
    tipo_residuo_id      BIGINT NOT NULL,
    PRIMARY KEY (empresa_reciclagem_id, tipo_residuo_id),
    FOREIGN KEY (empresa_reciclagem_id) REFERENCES empresa_reciclagem (id),
```



```
FOREIGN KEY (tipo_residuo_id) REFERENCES tipo_residuo (id)
);
```

Arquivo: lixo-no-ponto\src\main\resources\db\migration\V1.0.16__alter_table_usuario.sql ----

```
ALTER TABLE usuario
ADD COLUMN data_cadastro TIMESTAMP;
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.17__create_table_classe_de_usuario.sql ----

```
CREATE SEQUENCE role_id_seq START WITH 1 INCREMENT BY 1;
```

```
CREATE TABLE classe_de_usuario
(
  id BIGINT NOT NULL DEFAULT nextval('role_id_seq'),
  name VARCHAR(255) NOT NULL,
  PRIMARY KEY (id)
);
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.18__create_table_usuario_classe_de_usuario.sql ----

```
CREATE SEQUENCE usuario_classe_de_usuario_id_seq START WITH 1 INCREMENT BY 1;
```

```
CREATE TABLE usuario_classe_de_usuario
(
  id BIGINT NOT NULL DEFAULT nextval('usuario_classe_de_usuario_id_seq'),
  usuario_id BIGINT NOT NULL,
  classe_de_usuario_id BIGINT NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (usuario_id) REFERENCES usuario (id),
  FOREIGN KEY (classe_de_usuario_id) REFERENCES classe_de_usuario (id)
);
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.19__create_table_permissao.sql ----

```
CREATE SEQUENCE permissao_id_seq START WITH 1 INCREMENT BY 1;
```

```
CREATE TABLE permissao
```

```
(
  id BIGINT NOT NULL DEFAULT nextval('permissao_id_seq'),
  name VARCHAR(255) NOT NULL,
  PRIMARY KEY (id)
);
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.1__create_table_tipo_residuo.sql ----

```
CREATE TABLE IF NOT EXISTS tipo_residuo (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(255) NOT NULL
);
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.20__create_table_classe_de_usuario_permissao.sql ----

```
CREATE SEQUENCE classe_de_usuario_permissao_id_seq START WITH 1 INCREMENT
BY 1;
```

```
CREATE TABLE classe_de_usuario_permissao
(
    id BIGINT NOT NULL DEFAULT
nextval('classe_de_usuario_permissao_id_seq'),
    classe_de_usuario_id BIGINT NOT NULL,
    permissao_id BIGINT NOT NULL,
    PRIMARY KEY (id),
    FOREIGN KEY (classe_de_usuario_id) REFERENCES classe_de_usuario (id),
    FOREIGN KEY (permissao_id) REFERENCES permissao (id)
);
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.21__alter_table_empresa_reciclagem.sql ----

```
ALTER TABLE empresa_reciclagem
  ALTER COLUMN foto TYPE TEXT;
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.22__alter_table_ponto_descarte.sql ----

```
ALTER TABLE ponto_descarte
  ALTER COLUMN foto TYPE TEXT;
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.2__create_table_status_verificacao.sql

```
CREATE TABLE IF NOT EXISTS status_verificacao (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(255) NOT NULL
);
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.3__create_table_empresa_reciclagem.sql ----

```
CREATE TABLE IF NOT EXISTS empresa_reciclagem (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    contato VARCHAR(255) NOT NULL,
    site VARCHAR(255),
    id_tipo_residuo INTEGER REFERENCES tipo_residuo(id),
    horario_funcionamento VARCHAR(255),
    foto VARCHAR(255),
    patrocinador BOOLEAN NOT NULL DEFAULT FALSE
);
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.4__create_table_ponto_descarte.sql ----

```
CREATE TABLE IF NOT EXISTS ponto_descarte (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    id_tipo_residuo INTEGER REFERENCES tipo_residuo(id),
    latitude DECIMAL(10,6) NOT NULL,
    longitude DECIMAL(10,6) NOT NULL,
    id_status_verificacao INTEGER REFERENCES
status_verificacao(id),
    data_cadastro TIMESTAMP NOT NULL DEFAULT
CURRENT_TIMESTAMP,
    id_usuario_cadastro INTEGER NOT NULL REFERENCES
usuario(id),
    foto VARCHAR(255),
    descricao TEXT,
    data_ultima_verificacao TIMESTAMP,
    id_empresa_reciclagem INTEGER REFERENCES
empresa_reciclagem(id)
```

);

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.5__create_table_tipo_caminhao.sql ----

```
CREATE TABLE IF NOT EXISTS tipo_caminhao (
    id SERIAL PRIMARY KEY,
    nome VARCHAR(255) NOT NULL
);
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.6__create_table_rota_caminhao.sql ----

```
CREATE TABLE IF NOT EXISTS rota_caminhao (
    id SERIAL PRIMARY KEY,
    id_tipo_caminhao INTEGER REFERENCES tipo_caminhao(id),
    horario_inicio TIMESTAMP NOT NULL,
    horario_fim TIMESTAMP NOT NULL,
    cidade VARCHAR(255) NOT NULL,
    id_empresa_reciclagem INTEGER NOT NULL REFERENCES
empresa_reciclagem(id),
    ruas_passagem TEXT
);
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.7__create_table_ponto_descarte_empresa_reciclagem.sql ----

```
CREATE TABLE IF NOT EXISTS ponto_descarte_empresa_reciclagem (
    id SERIAL PRIMARY KEY,
    id_ponto_descarte INTEGER REFERENCES
ponto_descarte(id),
    id_empresa_reciclagem INTEGER REFERENCES
empresa_reciclagem(id)
);
```

Arquivo:

lixo-no-ponto\src\main\resources\db\migration\V1.0.8__create_table_ponto_descarte_usuario.sql ----

```
CREATE TABLE IF NOT EXISTS ponto_descarte_usuario (
    id SERIAL PRIMARY KEY,
    id_ponto_descarte INTEGER NOT NULL REFERENCES
ponto_descarte(id),
```

```

                                id_usuario INTEGER NOT NULL REFERENCES
usuario(id)
);

```

Arquivo: lixo-no-ponto\src\main\resources\db\migration\V1.0.9__insert_into_tipo_residuo.sql

```

INSERT INTO tipo_residuo (nome) VALUES
    ('Material Orgânico'),
    ('Papel e Papelão'),
    ('Plásticos'),
    ('Vidros'),
    ('Metais'),
    ('Eletrônicos'),
    ('Móveis Grandes'),
    ('Óleos'),
    ('Roupas');

```

Arquivo:
 lixo-no-ponto\src\test\java\com\example\lixonoponto\LixoNoPontoApplicationTests.java ----

```

package com.example.lixonoponto;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class LixoNoPontoApplicationTests {

    @Test
    void contextLoads() {
    }

}

```

Arquivo: lixo-no-ponto\system.properties ----

```

java.runtime.version=17

```

Apêndice F - Artigo SBC:

Lixo no ponto: um aplicativo para promover a colaboração comunitária na coleta de resíduos nas áreas urbanas.

André Joaquim Bez Birolo

Centro Tecnológico – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brazil

andre.j.bez.birolo@grad.ufsc.br

Abstract. *Urban waste management is a significant challenge, particularly in densely populated areas. This paper addresses this issue by presenting the development of a Progressive Web Application (PWA) aimed at improving urban waste management. The application, accessible on both mobile and desktop devices, was developed using Angular for the front-end and Java with Spring Boot for the back-end. The system includes features for user registration, waste disposal point verification, route visualization, and recycling company management. The evaluation, based on the Goal-Question-Metric approach, showed positive user feedback, confirming the system's effectiveness in urban waste management.*

Resumo. *A gestão de resíduos urbanos é um desafio significativo, especialmente em áreas densamente povoadas. Este artigo aborda essa questão apresentando o desenvolvimento de uma aplicação PWA para melhorar a gestão de resíduos urbanos. A aplicação, acessível em dispositivos móveis e desktops, foi desenvolvida utilizando Angular para o front-end e Java com Spring Boot para o back-end. O sistema inclui funcionalidades para cadastro de usuários, verificação de pontos de descarte, visualização de rotas e gerenciamento de empresas de reciclagem. A avaliação, baseada na abordagem Goal-Question-Metric, mostrou feedback positivo dos usuários, confirmando a eficácia do sistema na gestão de resíduos urbanos.*

1. Introdução

A gestão de resíduos sólidos é um grande desafio atual, especialmente devido à sua ligação com o clima. Em 2019, o Brasil produziu 79 milhões de toneladas de resíduos sólidos urbanos, dos quais apenas 59,5% foram coletados seletivamente. A Política Nacional de Resíduos Sólidos (PNRS) visa a gestão integrada de resíduos, no entanto, muitos municípios enfrentam dificuldades devido à falta de recursos, educação ambiental e infraestrutura (ABRELPE, 2020).

A má gestão de resíduos afeta não apenas o meio ambiente, mas também a saúde pública e a economia. Os impactos incluem poluição, doenças, perda de biodiversidade e prejuízos econômicos. Portanto, é crucial educar a população e fornecer soluções tecnológicas para melhorar a gestão de resíduos.

Nesse contexto, uma solução proposta é desenvolver um aplicativo web para promover a gestão integrada de resíduos em Florianópolis. O aplicativo informará sobre as rotas dos caminhões de lixo e os locais de descarte, além de permitir que os usuários informem pontos de coleta seletiva. Espera-se que essa iniciativa aumente a

conscientização e promova a reciclagem, beneficiando os catadores, a indústria e o meio ambiente.

A estrutura do artigo está organizada da seguinte forma: na seção 1, apresenta-se a introdução, contextualizando o problema da gestão de resíduos urbanos e a proposta da solução. Em seguida, a seção 2 discute o background, revisando a literatura sobre tecnologias de gestão de resíduos e soluções similares existentes. Posteriormente, a seção 3 detalha o desenvolvimento do aplicativo, incluindo a escolha das tecnologias, a modelagem de dados e a implementação das funcionalidades. Na seção 4, descreve-se o sistema Lixo no Ponto, destacando suas principais funcionalidades. Além disso, a seção 5 apresenta a avaliação do sistema, utilizando a abordagem GQM (Goal-Question-Metric) para medir a eficácia e a usabilidade do aplicativo. Finalmente, a seção 6 conclui o artigo, resumindo os resultados obtidos e sugerindo trabalhos futuros.

2. Background

Atualmente, diversas tecnologias têm sido desenvolvidas para auxiliar na gestão de resíduos sólidos e promover a reciclagem. No mercado, existem várias soluções que utilizam tecnologias da informação e comunicação (TICs) para sensibilizar, informar e engajar os cidadãos na separação e no descarte correto dos resíduos. Essas soluções tecnológicas não só melhoram a eficiência da gestão de resíduos, mas também visam engajar os cidadãos de forma mais efetiva na adoção de práticas sustentáveis, transformando o problema do descarte inadequado em uma oportunidade para a construção de comunidades mais conscientes e ecologicamente responsáveis (EXATI, 2023).

Um exemplo de tecnologia aliada à reciclagem são os aplicativos móveis que informam os cidadãos sobre os tipos e as formas de descarte e reciclagem dos resíduos. Esses aplicativos fornecem informações sobre os serviços públicos de limpeza urbana e coleta seletiva, bem como sobre os locais adequados para a entrega voluntária ou a reciclagem dos resíduos (EXATI, 2024). Além disso, esses aplicativos podem estimular a participação social e o controle social na gestão dos resíduos sólidos, por meio de canais de comunicação entre os usuários, os gestores públicos e os agentes envolvidos na cadeia produtiva dos resíduos (NOGUEIRA, 2024).

Outra solução tecnológica é o uso de sensores e dispositivos IoT (Internet of Things) para monitorar a quantidade de resíduos em lixeiras e contêineres. Esses dispositivos podem enviar dados em tempo real para os gestores de resíduos, permitindo uma coleta mais eficiente e reduzindo os custos operacionais (TEKON ELECTRONICS, 2018). Além disso, a análise de dados coletados por esses dispositivos pode ajudar a identificar padrões de geração de resíduos e a planejar melhor as rotas de coleta (DICATUDO, 2024; SHE SOLUÇÕES, 2024).

No contexto internacional, algumas cidades têm implementado sistemas de gestão de resíduos baseados em tecnologias avançadas. Por exemplo, em algumas cidades europeias, foram instalados contêineres subterrâneos equipados com sensores que monitoram o nível de resíduos e notificam os serviços de coleta quando estão

cheios. Esses sistemas não só melhoram a eficiência da coleta de resíduos, mas também contribuem para a redução de odores e a melhoria da estética urbana (LEMOBS, 2024; EUROPARL, 2024). Cidades como Estocolmo e Copenhague têm se destacado na implementação dessas tecnologias, utilizando soluções inovadoras para otimizar a gestão de resíduos e promover a sustentabilidade urbana (CIDAESSUSTENTAVEIS, 2024; NOVELO, 2024).

Além disso, plataformas online e redes sociais têm sido utilizadas para promover campanhas de educação ambiental e conscientização sobre a importância da reciclagem. Essas plataformas permitem a disseminação de informações e boas práticas, além de facilitar a organização de eventos e ações comunitárias voltadas para a gestão de resíduos (REDE SOCIAIS, 2024; ECOLOGIA ATIVA, 2024). Iniciativas como grupos de discussão e páginas dedicadas à sustentabilidade têm se mostrado eficazes em engajar a população e fomentar um comportamento mais responsável em relação ao descarte de resíduos (VERDE FUTURO, 2024).

Em resumo, a tecnologia tem se mostrado uma aliada poderosa na promoção da reciclagem e na melhoria da gestão de resíduos sólidos. As soluções tecnológicas disponíveis no mercado têm o potencial de transformar a forma como os resíduos são geridos, tornando o processo mais eficiente, sustentável e participativo.

3. Desenvolvimento

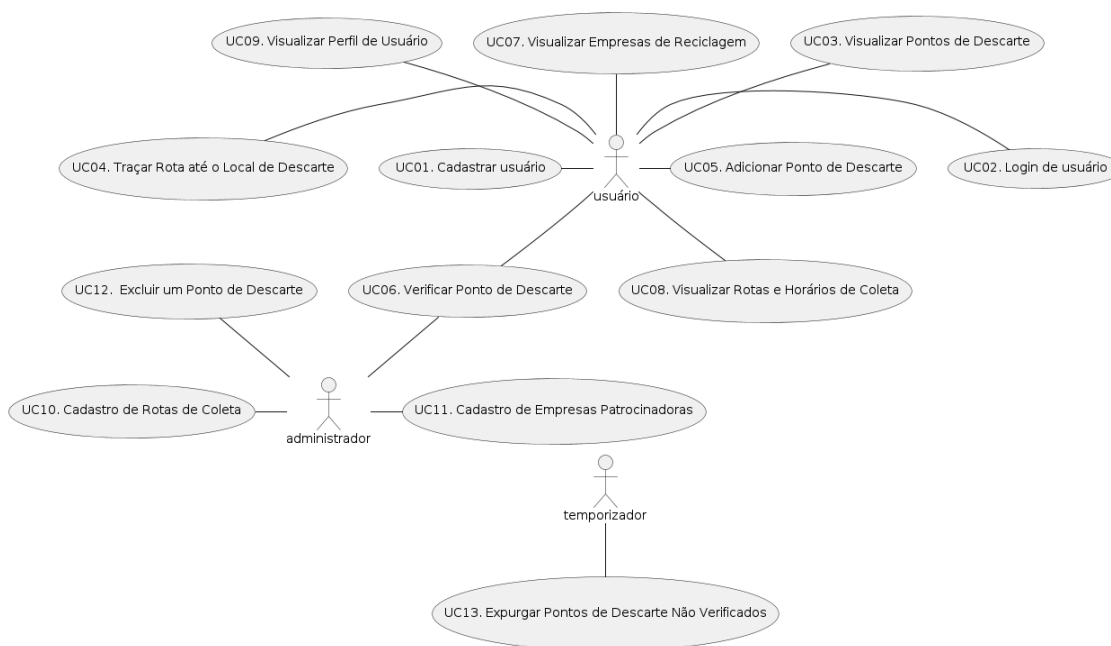
Na fase de análise e projeto, são levantados os requisitos e características técnicas que a aplicação terá. Além disso, é realizada a modelagem de classes utilizando a UML (Unified Modeling Language) e a prototipação de alta fidelidade através da ferramenta Figma. A implementação é a maior etapa do desenvolvimento, destinada ao desenvolvimento da aplicação. Para o Front-End, é utilizado o framework de JavaScript chamado Angular, enquanto para o Back-End, é utilizado o framework de Java chamado Spring Boot.

Por fim, a etapa de teste é destinada ao teste de integração das funcionalidades da aplicação, garantindo que todos os componentes funcionem corretamente em conjunto.

3.1. Especificação

Os casos de uso são uma técnica de modelagem que descreve a interação entre um ator (geralmente um usuário) e um sistema para atingir um objetivo específico. Eles fornecem uma visão clara e concisa do que o sistema fará e como será utilizado, garantindo que os requisitos do sistema estejam claramente definidos e que as expectativas do usuário sejam atendidas. A Figura 1 mostra o diagrama de caso de uso do sistema.

Figura 1 – Diagrama de caso de uso

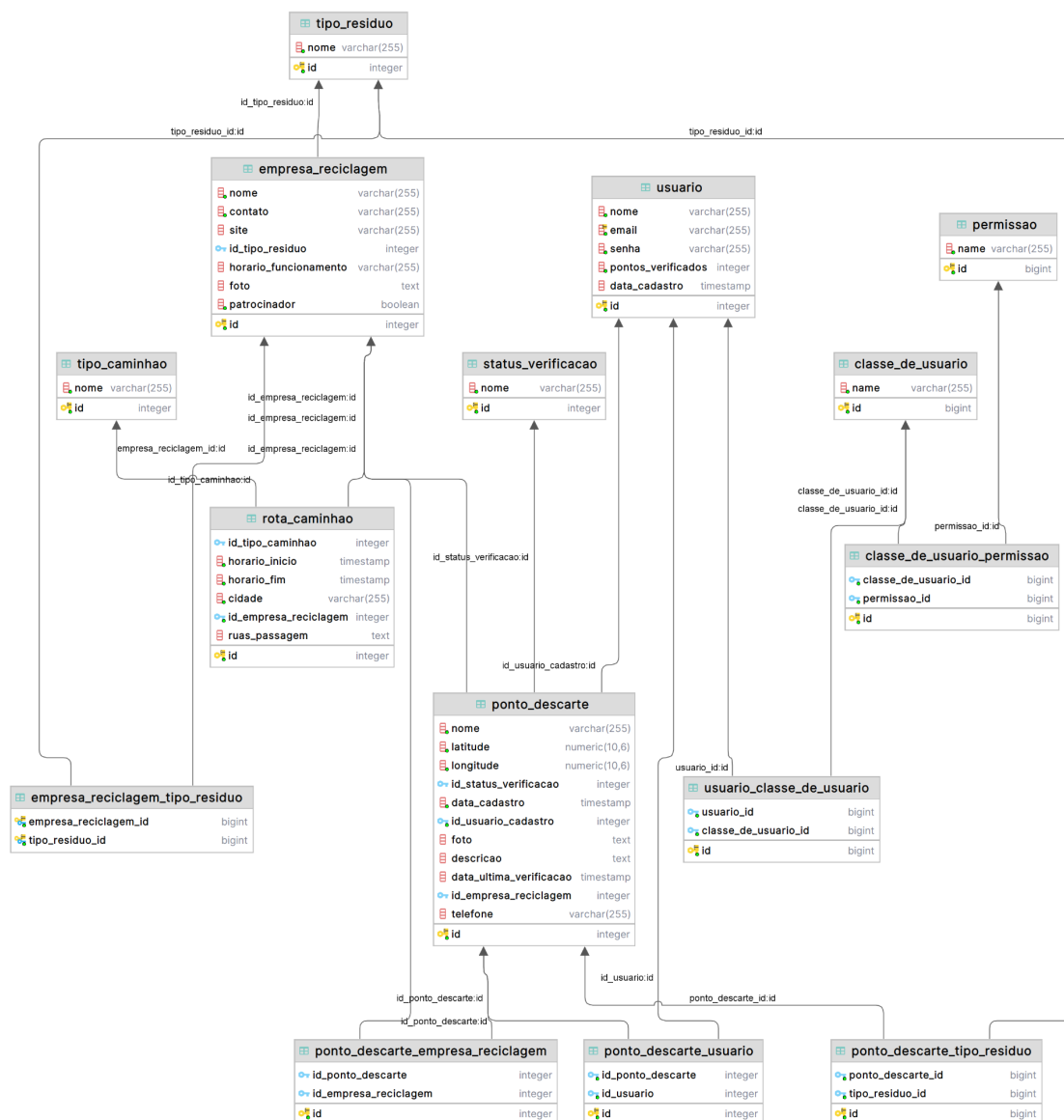


Com base em Amoasei (2023), as regras de negócio são essenciais para definir e limitar a estrutura e o comportamento de um negócio. No contexto do aplicativo em questão, essas regras estão intimamente ligadas à interação do usuário com o aplicativo e à maneira como as informações são apresentadas e gerenciadas. As regras de negócio do sistema são apresentadas a seguir:

1. O cadastro deve ser bem-sucedido apenas se todos os campos de cadastro forem preenchidos.
2. As categorias para os pontos de descarte são: Material Orgânico, Papel e Papelão, Plásticos, Vidros, Metais, Eletrônicos, Móveis Grandes, Óleos e Roupas.
3. Os meios de contato com as empresas de reciclagem podem ser: E-mail, Telefone e Whatsapp.
4. Um ponto de descarte se torna "verificado" quando:
 - a. Três usuários diferentes confirmam a localização do ponto de descarte, ou
 - b. Um administrador verifica o ponto.
5. O usuário precisa estar conectado em sua conta para verificar um ponto de descarte.
6. Os dados do usuário poderão ser visualizados apenas pelo próprio usuário, não sendo compartilhados em nenhum local público da aplicação.
7. As opções para o tipo de coleta são: Normal e Reciclável.

A técnica de modelagem de dados é empregada para organizar e documentar a representação dos dados em um sistema. O Modelo Entidade-Relacionamento é um dos diagramas utilizados para essa finalidade, onde os dados são representados por entidades e suas relações (LI; CHEN, 2009). A Figura 2 mostra o modelo escolhido para a solução proposta neste estudo.

Figura 2 – Modelo Entidade-Relacionamento



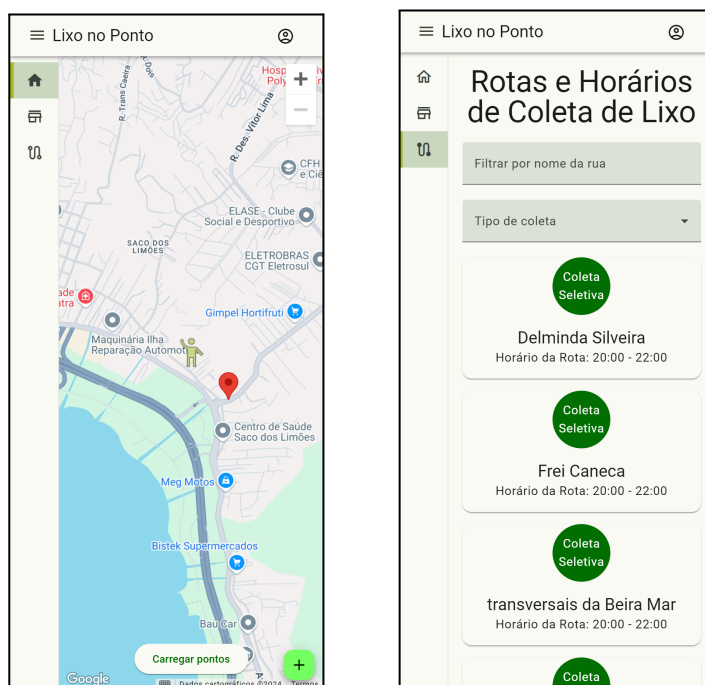
4. Lixo no Ponto

O PWA Lixo no Ponto possui diversas funcionalidades que facilitam a gestão de resíduos sólidos urbanos. Entre as principais funcionalidades, destacam-se: cadastro de usuário, que permite que novos usuários se registrem no sistema; login de usuário, que

permite que usuários registrados façam login no sistema; visualização de pontos de descarte, que exibe no mapa os pontos de descarte cadastrados; cadastro de pontos de descarte, que permite que os usuários adicionem novos pontos de descarte; verificação de pontos de descarte, onde usuários podem verificar a localização de pontos de descarte; visualização de empresas de reciclagem, que exibe informações sobre empresas de reciclagem; visualização de rotas e horários de coleta, que mostra as rotas e horários de coleta de lixo; e filtragem de rotas de coleta, que permite filtrar as rotas de coleta por rua ou tipo de coleta.

A seguir, são apresentadas duas imagens da versão mobile do PWA Lixo no Ponto, conforme a Figura 3a, mostrando a tela inicial do aplicativo, e a Figura 3b, mostrando tela de rotas do aplicativo:

Figura 3 - (a) Página Inicial e (b) Rotas em dispositivos móveis



Na versão desktop, a Figura 4 mostra as informações de um ponto de descarte e outra imagem mostra a página de empresas de reciclagem:

Figura 4 - Modal de Informações do Ponto de Descarte

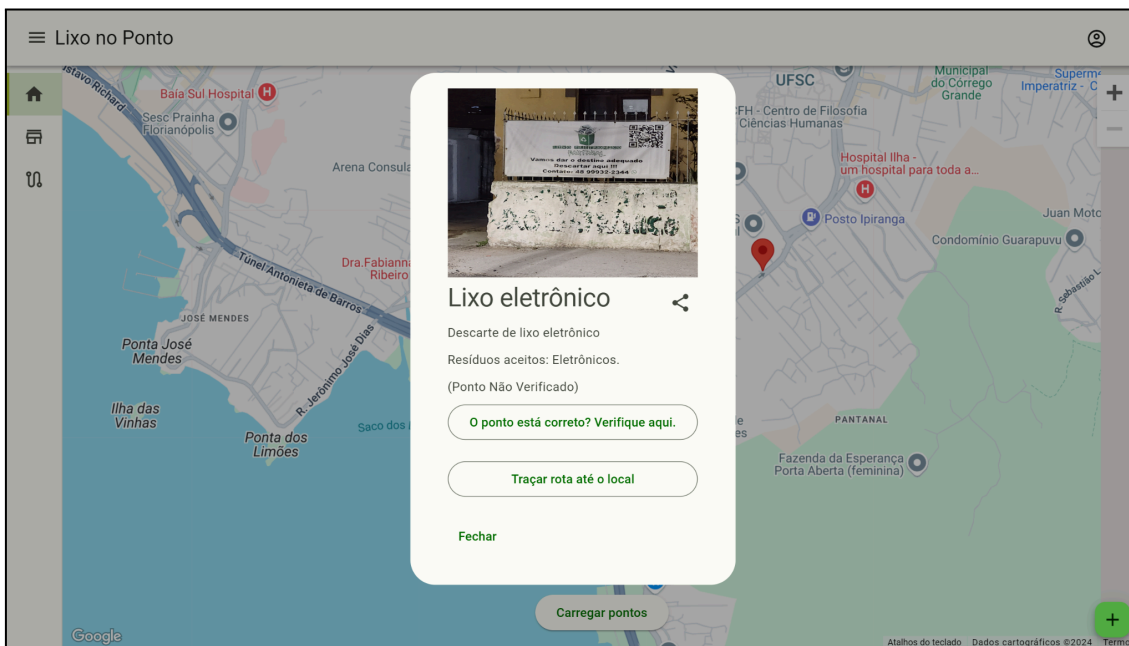


Figura 5 - Página de Empresas de Reciclagem



5. Avaliação

Para avaliar o sistema desenvolvido, foi utilizada a abordagem Goal-Question-Metric (GQM), que define metas específicas, formula perguntas operacionais e específica métricas necessárias para responder a essas perguntas. A estrutura do GQM é organizada em três níveis: Goal (Objetivo), Question (Pergunta) e Metric (Métrica).

A seção de planejamento envolve a definição dos objetivos, perguntas e métricas. A Escala Likert foi empregada para quantificar as respostas dos usuários, permitindo

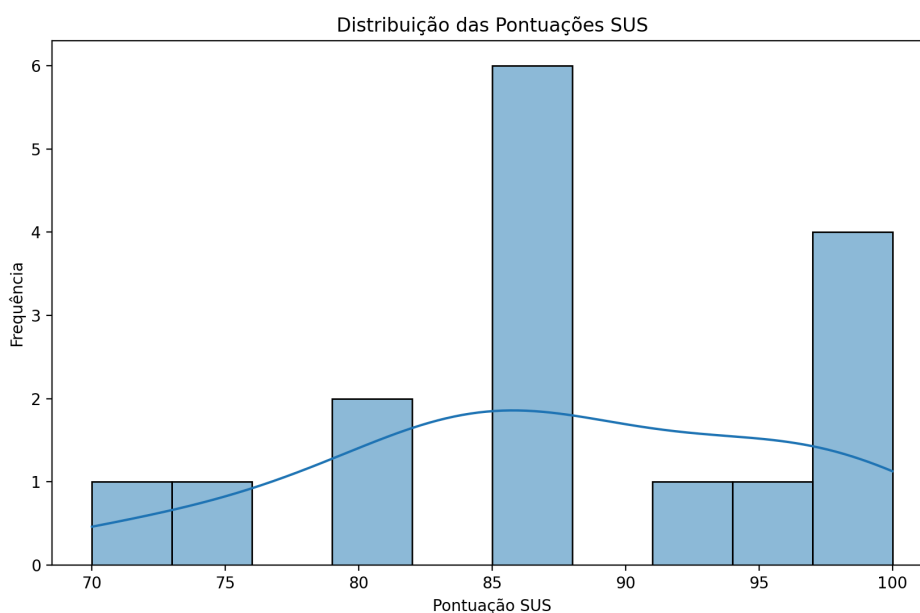
uma avaliação precisa e padronizada das percepções e opiniões em relação aos objetivos do projeto.

O primeiro objetivo (G1) é ampliar o conhecimento sobre a prática e políticas de gestão de resíduos urbanos da comunidade em um ambiente urbano. Para isso, a pergunta Q1.1 investiga quão efetivo é o sistema em informar sobre os pontos de descarte e rotas de coleta de lixo. As métricas associadas incluem quantos pontos de descarte e rotas de coleta de lixo os usuários conheciam antes e após usar o sistema.

O segundo objetivo (G2) é avaliar a usabilidade do projeto Lixo no Ponto em relação à experiência do usuário e à eficiência do sistema. A pergunta Q2.1 busca determinar a pontuação na escala SUS obtida pelo sistema, com métricas que avaliam a frequência de uso, complexidade, facilidade de uso e necessidade de ajuda técnica.

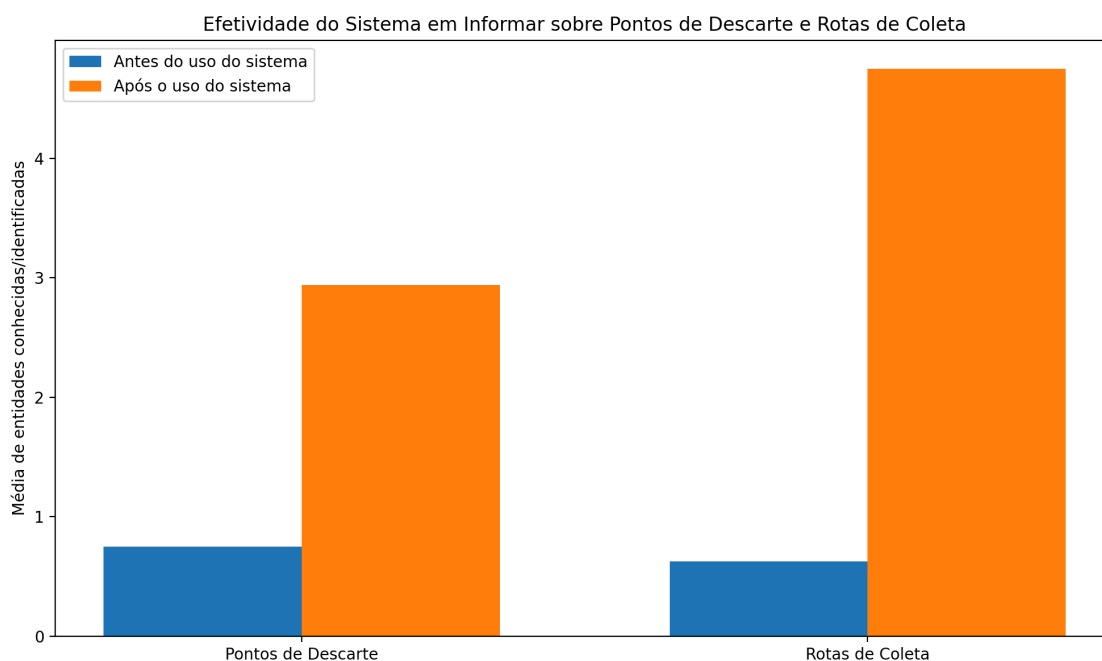
A execução da avaliação seguiu três etapas. Na primeira etapa, foi fornecido aos usuários um documento detalhado orientando-os a realizarem determinadas tarefas. Na segunda etapa, os usuários utilizaram o aplicativo para realizar as tarefas especificadas. Na terceira etapa, os usuários responderam ao questionário.

Figura 6 - Pontuação SUS da Avaliação



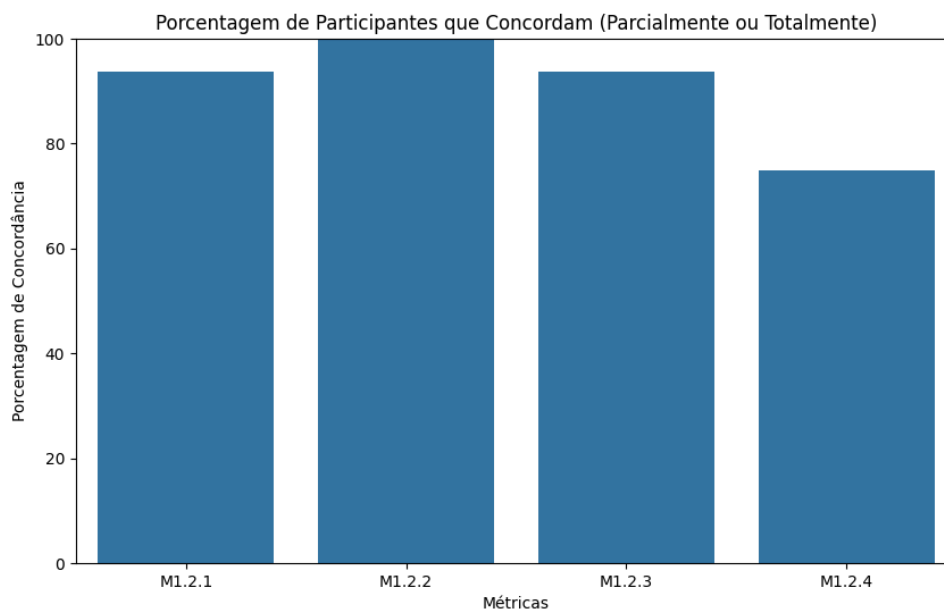
A análise dos resultados revelou que a média de pontos de descarte conhecidos antes do uso do sistema é de 0,75, enquanto a média de pontos de descarte identificados após o uso do sistema é de 2,94. Em relação às rotas de coleta de lixo, a média de rotas conhecidas antes do uso do sistema é de 0,62, enquanto a média de rotas identificadas após o uso do sistema é de 4,75. A pontuação média SUS obtida pelo sistema foi de 87,50, considerada excelente, conforme Figura 6.

Figura 7 - Relação entre Pontos de Descartes e Rotas de Coletas Conhecidos Antes e Depois do Uso da Aplicação



Fonte: Autor (2024)

Figura 8 - Grau de Compreensão das Informações Sobre os Pontos de Descarte e Rotas de Coleta de Lixo



Fonte: Autor (2024)

A partir das Figuras 7 e 8, os resultados indicam que o sistema cumpre satisfatoriamente seu objetivo de informar os usuários sobre pontos de descarte e rotas de coleta de lixo, com a maioria dos usuários descobrindo novas entidades após a utilização do sistema. A análise quantitativa dos dados corrobora a alta efetividade do sistema em ambas as dimensões avaliadas.

6. Conclusões

Conclui-se que o desenvolvimento da aplicação PWA para a gestão de resíduos urbanos atingiu plenamente os objetivos propostos. A aplicação, acessível em dispositivos móveis e desktops, foi projetada utilizando Angular para o front-end e Java com Spring Boot para o back-end, tendo sua usabilidade considerada adequada pelos usuários que a avaliaram. A definição dos requisitos funcionais e não funcionais, seguida pela elaboração de casos de uso e protótipos, permitiu a implementação de um sistema Produto Mínimo Viável (MVP), com funcionalidades mínimas para usuários comuns e administradores, incluindo cadastro e verificação de pontos de descarte, visualização de rotas de coleta e gerenciamento de empresas de reciclagem. A avaliação do sistema foi realizada por meio da abordagem Goal-Question-Metric (GQM), que possibilitou a definição de objetivos específicos, formulação de perguntas e coleta de métricas. A análise dos resultados obtidos através de questionários aplicados aos usuários revelou avaliações positivas, indicando que a maioria dos participantes considerou o sistema fácil de usar, com informações claras e adequadas, além de funcionalidades bem integradas. Tais resultados confirmam que o sistema desenvolvido atende às necessidades dos usuários e contribui para a gestão eficiente de resíduos urbanos, demonstrando o sucesso do projeto.

Referências

ABRELPE. Panorama dos Resíduos Sólidos no Brasil 2019. São Paulo: ABRELPE, 2020. Disponível em: <https://abrelpe.org.br/download-panorama-2018-2019/>. Acesso em: 25 ago. 2023.

AMOASEI, Juliana (org.). O que são regras de negócio? 2023. Disponível em: <https://www.alura.com.br/artigos/o-que-sao-regras-de-negocio>. Acesso em: 16 jun. 2024.

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. Goal Question Metric Paradigm. 2. ed. EUA: Wiley, 1994.

CIDADESSUSTENTAVEIS. Cidades sustentáveis: inovações na gestão de resíduos. Disponível em: <https://www.cidadessustentaveis.org/inovacoes-gestao-residuos>. Acesso em: 17 dez. 2024.

DICATUDO (org.). Uso de Dispositivos Inteligentes para Gerenciamento de Resíduos Domésticos. Disponível em: <https://dicatudo.com.br/uso-de-dispositivos-inteligentes-para-gerenciamento-de-residuos-domesticos/>. Acesso em: 11 dez. 2024.

ECOLOGIA ATIVA. Campanhas de conscientização: como as plataformas digitais ajudam na reciclagem. Disponível em: <https://www.ecologiaativa.com.br/campanhas-conscientizacao-reciclagem>. Acesso em: 17 dez. 2024.

EVELYN NOGUEIRA (org.). 6 aplicativos que auxiliam na reciclagem de resíduos sem sair de casa. 2024. Disponível em: <https://umsoplaneta.globo.com/sociedade/noticia/2021/05/24/6-aplicativos-que-auxiliam-na-reciclagem-de-residuos-sem-sair-de-casa.ghtml>. Acesso em: 10 dez. 2024.

EXATI. Tecnologia na gestão de resíduos: qual a importância para as cidades? 2023. Disponível em: <https://blog.exati.com.br/tecnologia-na-gestao-de-residuos/>. Acesso em: 10 dez. 2024.

EUROPARL. Gestão de resíduos na União Europeia: desafios e inovações. Disponível em: <https://www.europarl.europa.eu/portal/gestao-residuos>. Acesso em: 17 dez. 2024.

LEMOBS. Lixo: a gestão de resíduos sólidos nas cidades. Disponível em: <https://www.lemobs.com.br/lixo-gestao-residuos>. Acesso em: 17 dez. 2024.

LI, Qing; CHEN, Yu-Liu. Modeling and Analysis of Enterprise and Information Systems: from requirements to realization. Beijing: Springer, 2009. Disponível em: <https://link.springer.com/book/10.1007/978-3-540-89556-5>. Acesso em: 17 jun. 2024.

NOVELO. Tecnologias de gestão de resíduos em Estocolmo e Copenhague. Disponível em: <https://www.novelo.com.br/tecnologias-gestao-residuos-estocolmo-copenhague>. Acesso em: 17 dez. 2024.

REDE SOCIAIS. O papel das redes sociais na educação ambiental. Disponível em: <https://www.redessociais.com.br/educacao-ambiental>. Acesso em: 17 dez. 2024.

SECRETARIA DE MEIO AMBIENTE, INFRAESTRUTURA E LOGÍSTICA (SEMIL). Resíduos Sólidos. Disponível em: <https://semil.sp.gov.br/residuos-solidos/>. Acesso em: 26 ago. 2023.

SHE SOLUÇÕES (org.). Inovação em gestão de resíduos: soluções inteligentes para um futuro sustentável. Disponível em: <https://shesolucoes.com.br/gestao-de-residuos/>. Acesso em: 13 dez. 2024.

TEKONELETRONICS (org.). Revolucionar a Gestão de Resíduos com Sensores IoT. Disponível em: <https://www.tekonelectronics.com/pt/media/tekon-blog/waste-management-with-iot-sensores-7c-tekon-electronics/>. Acesso em: 12 dez. 2024.

VERDE FUTURO. Engajamento comunitário na gestão de resíduos: o uso das redes sociais. Disponível em:

<https://www.verdefuturo.com.br/engajamento-comunitario-redes-sociais>. Acesso em: 17 dez. 2024.