UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ÁLVARO GUILHERME PIRES

ECOCOLETA: UMA APLICAÇÃO WEB PARA FACILITAR A COLETA SELETIVA

GUARAPUAVA

2025

ÁLVARO GUILHERME PIRES

ECOCOLETA: UMA APLICAÇÃO WEB PARA FACILITAR A COLETA SELETIVA

EcoColeta: A web application to facilitate selective collection

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Roni Fabio Banaszewski

Coorientador: Prof. Dr. Diego Marczal

GUARAPUAVA

2025

4.0 Internacional

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

ÁLVARO GUILHERME PIRES

ECOCOLETA: UMA APLICAÇÃO WEB PARA FACILITAR A COLETA SELETIVA

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 19/fevereiro/2025

Prof. Roni Fabio Banaszewski
Doutor
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

---

Prof. Luciano Ogiboski
Doutor
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

---

Prof. Andres Jessé Porfirio
Doutor
Universidade Tecnológica Federal do Paraná - Campus Guarapuava

GUARAPUAVA

2025

AGRADECIMENTOS

Agradeço, em primeiro lugar, à minha família, pelo apoio incondicional, pelo incentivo e por sempre acreditarem no meu potencial ao longo da minha jornada acadêmica.

Aos meus professores da Universidade Tecnológica Federal do Paraná, em especial ao meu orientador, Prof. Dr. Roni Fabio Banaszewski, e ao meu coorientador, Prof. Dr. Diego Marczal, pela dedicação, orientação e incentivo, fundamentais para a realização deste trabalho.

À minha namorada, Evelyn Carraro, por estar sempre ao meu lado, oferecendo apoio, motivação e compreensão nos momentos mais desafiadores.

Por fim, expresso minha sincera gratidão a todos que, de alguma forma, contribuíram para a concretização deste trabalho.

"A alegria está na luta, na tentativa, no sofrimento envolvido e não na vitória propriamente dita."

— Mahatma Gandhi

RESUMO

O descarte inadequado de resíduos sólidos provenientes de diversas fontes gera graves impactos ambientais e sociais. No Brasil, a quantidade de resíduos recicláveis descartada corretamente ainda é insuficiente, refletindo um desafio persistente. Diante deste cenário, diversas cidades brasileiras vêm implementando iniciativas para incentivar a reciclagem e o descarte correto do lixo. Em Guarapuava, Paraná, por exemplo, além da coleta semanal de lixo reciclável nas residências, foram instaladas lixeiras comunitárias em forma de contêineres plásticos em alguns bairros. No entanto, essa abordagem apresenta problemas, como o espaço ocupado nas ruas, altos custos de aquisição e manutenção, mau odor devido à falta de higienização frequente e, principalmente, a ineficiência no processo, uma vez que muitos residentes ainda descartam incorretamente seus resíduos. Isso leva à necessidade de intervenção direta dos catadores, que realizam a separação manual do material reciclável. A falta de informações e de acesso a soluções adequadas de descarte contribui significativamente para o agravamento desse problema no país. Assim, o incentivo à coleta seletiva e a melhoria das condições de trabalho dos catadores de materiais recicláveis tornam-se fundamentais para intensificar o reaproveitamento de resíduos e direcioná-los corretamente aos centros de reciclagem. Como resposta a essa necessidade, foi desenvolvido um Produto Mínimo Viável, do inglês *Minimum Viable Product* (MVP) na forma de uma aplicação web que facilita a comunicação direta entre catadores e separadores geradores de resíduos. A aplicação busca promover uma coleta mais eficiente e, ao mesmo tempo, valorizar o papel dos catadores no processo de reciclagem. A aplicação inclui funcionalidades como a conscientização sobre o descarte correto de resíduos, a avaliação da coleta pelo separador, a atribuição de pontuações por boas práticas e a orientação geográfica, permitindo que os catadores identifiquem com precisão os pontos de coleta de materiais recicláveis disponibilizados pelos residentes. Com isso, espera-se promover maior eficiência na coleta e maior valorização dos catadores. No entanto, para que a solução tenha um impacto significativo, é essencial que haja adesão em larga escala tanto por parte dos separadores quanto dos catadores.

Palavras-chave: catadores de materiais recicláveis; desenvolvimento web; coleta seletiva; reciclagem; descarte inadequado de resíduos.

ABSTRACT

The improper disposal of solid waste from various sources generates severe environmental and social impacts. In Brazil, the amount of correctly disposed recyclable waste is still insufficient, reflecting a persistent challenge. In response to this scenario, several Brazilian cities have been implementing initiatives to encourage recycling and proper waste disposal. In Guarapuava, Paraná, for example, in addition to the weekly collection of recyclable waste from residences, community bins in the form of plastic containers have been installed in some neighborhoods. However, this approach presents problems such as street space occupation, high acquisition and maintenance costs, unpleasant odors due to a lack of frequent cleaning, and, primarily, inefficiency in the process, as many residents still dispose of their waste incorrectly. This leads to the need for the direct intervention of waste pickers, who manually separate recyclable materials. The lack of information and access to proper disposal solutions significantly exacerbates this issue in the country. Thus, encouraging selective collection and improving the working conditions of recyclable material collectors is essential to enhance waste reuse and properly direct it to recycling centers. In response to this need, an MVP was developed in the form of a web application that facilitates direct communication between waste pickers and waste-generating residents. The application aims to promote more efficient waste collection while enhancing the role of waste pickers in the recycling process. The application includes features such as raising awareness about proper waste disposal, evaluating the collection by the resident, assigning scores for good practices, and providing geographic guidance, allowing waste pickers to accurately identify collection points for recyclable materials made available by residents. With this, the goal is to promote greater efficiency in collection and increased appreciation of waste pickers. However, for the solution to have a significant impact, large-scale adoption by both residents and waste pickers is essential.

Keywords: recyclable material pickers; web development; selective collect; recycling; improper waste disposal.

LISTA DE FIGURAS

Figura 1 – Exemplo das lixeiras coletivas na forma de contêiner 12
Figura 2 – Dinâmica do Scrum 21
Figura 3 – Dinâmica do Kanban 22
Figura 4 – Exemplo etapas TDD 23
Figura 5 – Apresenta a tela da organização de um quadro 24
Figura 6 – Fluxo de trabalho do Gitflow 25
Figura 7 – Imagem do fluxo de trabalho de um projeto utilizando Integração Contínua, do inglês *Continuous Integration* (CI)/Entrega Contínua, do inglês *Continuous Delivery* (CD). 25
Figura 8 – Como docker funciona 30
Figura 9 – Tela inicial Meu Copo Eco 33
Figura 10 – Telas app Cataki 34
Figura 11 – Telas app iRecycle 35
Figura 12 – Imagem inicial da modelagem do banco de dados da aplicação EcoColeta. 41
Figura 13 – Modelagem final do banco de dados da aplicação EcoColeta. 43
Figura 14 – Tela de login do sistema. 45
Figura 15 – Tela de cadastro de catador e separador. 46
Figura 16 – Tela de solicitar coleta. 47
Figura 17 – Telas inicial e perfil (catador). 48
Figura 18 – Telas coletas em andamento. 49
Figura 19 – Telas de login e cadastro de catador 51
Figura 20 – Telas de cadastro: separador, usuário com erro e endereços. 52
Figura 21 – Telas de solicitação de coleta e listagem das coletas atuais. 53
Figura 22 – Telas listagem das coletas e visualização de catadores ativos no mapa. 55
Figura 23 – Telas mapa com pontos e rota para coletas 58
Figura 24 – Telas interface de avaliação e pontuação da coleta. 60
Figura 25 – Telas relatórios e listagem de catadores. 61
Figura 26 – Telas de admin: listagem, edição e criação de prefeituras 62
Figura 27 – Telas de dicas e introdução da aplicação. 63

LISTA DE TABELAS

Tabela 1 – Estudo comparativo **35**
Tabela 2 – Histórias de usuário **39**
Tabela 3 – Histórias adicionadas **40**
Tabela 4 – Histórias de Implementação futura **69**

LISTA DE ABREVIATURAS E SIGLAS

Siglas

ACID	Atomicidade, Consistência, Isolamento e Durabilidade
API	Interface de Programação de Aplicativos, do inglês *Application Programming Interface*
APP	Aplicativo
CD	Entrega Contínua, do inglês *Continuous Delivery*
CI	Integração Contínua, do inglês *Continuous Integration*
GIST	Árvore de Busca Generalizada, do inglês *Generalized Search Tree*
HTTP	Protocolo de Transferência de Hipertexto, do inglês *Hypertext Transfer Protocol*
HTTPS	Protocolo Seguro de Transferência de Hipertexto, do inglês *Hypertext Transfer Protocol Secure*
IaaS	Infraestrutura como Serviço, do inglês *Infrastructure as a Service*
IDs	Identificação
JPA	API de Persistência Java, do inglês *Java Persistence API*
JSON	Notação de Objetos JavaScript, do inglês *JavaScript Object Notation*
JWT	Tokens Web no Formato JSON, do inglês *JSON Web Tokens*
MVC	Modelo-Visão-Controlador, do inglês *Model-View-Controller*
MVP	Produto Mínimo Viável, do inglês *Minimum Viable Product*
ONG	Organização não-governamental
PaaS	Plataforma como Serviço, do inglês *Platform as a Service*
PWA	Aplicativo Web Progressivo, do inglês *Progressive Web App*
REST	Transferência de Estado Representacional, do inglês *Representational State Transfer*
RxJS	Extensões Reativas para JavaScript, do inglês *Reactive Extensions for JavaScript*

SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	Linguagem de Consulta Estruturada, do inglês *Structured Query Language*
SSL	Camada de Soquetes Seguros, do inglês *Secure Sockets Layer*
TDD	Desenvolvimento Orientado por Testes, do inglês *Test Driven Development*

SUMÁRIO

1 INTRODUÇÃO 12

1.1 Objetivos 16

1.1.1 Objetivo geral 16

1.1.2 Objetivos específicos 16

1.2 Justificativa 17

1.3 Estrutura do trabalho 17

2 REFERENCIAL TEÓRICO 19

2.1 Processo de desenvolvimento 20

2.1.1 Metodologia ágil *Scrum* 20

2.1.2 Kanban 21

2.1.3 Desenvolvimento Orientado por Testes 22

2.2 Ferramentas de desenvolvimento 23

2.2.1 Trello 23

2.2.2 Git, Gitflow e GitHub 24

2.2.3 Render 26

2.2.4 Flyway 26

2.3 Tecnologias do lado cliente 27

2.3.1 Angular e Tecnologias Relacionadas 27

2.3.2 Progressive Web App (PWA) 28

2.3.3 Google Maps e Serviços de Geolocalização 29

2.4 Tecnologias do lado servidor 29

2.4.1 Docker 30

2.4.2 Spring Boot e Tecnologias Relacionadas 30

2.4.3 PostgreSQL 32

2.4.4 PostGIS 32

3 TRABALHOS RELACIONADOS 33

3.1 ReciclaBR 33

3.2 Meu Copo Eco 33

3.3 Cataki 34

3.4 iRecycle 34

3.5 Estudo Comparativo . . . 35
4 MATERIAIS E MÉTODOS . . . 36
5 ANALISE E PROJETO DO SISTEMA . . . 38
5.1 Análise do Sistema . . . 38
5.1.1 Papéis dos Usuários no Sistema . . . 40
5.1.2 Modelagem do Banco de Dados . . . 41
5.1.3 Protótipos das Telas . . . 45
5.1.3.1 Tela de Login . . . 45
5.1.3.2 Formulários de Cadastro de Catador e Separador . . . 46
5.1.3.3 Tela Inicial do Perfil de Separador e Formulário de Solicitação de Coleta . . . 46
5.1.3.4 Tela do Perfil de Catador e Mapa Interativo . . . 47
5.1.3.5 Tela de Funcionalidade de Coletas e Rotas Otimizadas . . . 49
6 DESENVOLVIMENTO DO SISTEMA . . . 50
6.1 Sprint 1 . . . 50
6.2 Sprint 2 . . . 51
6.3 Sprint 3 . . . 53
6.4 Sprint 4 . . . 55
6.5 Sprint 5 . . . 58
6.6 Sprint 6 . . . 59
6.7 Sprint 7 . . . 61
6.8 Sprint 8 . . . 63
6.9 Sprint 9 . . . 64
6.10 Considerações Finais do Desenvolvimento . . . 64
6.10.1 Desafios e Adaptações . . . 64
6.10.2 Aprendizado Técnico e Integrações . . . 65
6.10.3 Integração e Deploy no Ambiente de Produção . . . 65
6.10.4 Mudanças no Banco de Dados e Refinamento do Modelo . . . 65
6.10.5 Impacto do PWA no Projeto . . . 66
7 CONCLUSÃO . . . 67
7.1 Trabalhos Futuros . . . 68
REFERÊNCIAS . . . 70

1 INTRODUÇÃO

O descarte incorreto de resíduos provenientes de indústrias, residências, comércios, agrícolas e de serviços, pode acarretar diversos desequilíbrios ambientais e sociais (ANDRADE; FERREIRA, 2011). Segundo dados de 2022 da *International Solid Waste Association* (ISWA)[1], o Brasil produz cerca de 27,7 milhões de resíduos recicláveis ao ano e somente 4% obtêm o descarte correto.

No município de Guarapuava - PR, no ano de 2011, foi inaugurado o aterro sanitário da cidade, sendo destino diário de aproximadamente 78 toneladas de lixos. Em 2017, a captação foi de cerca de 120 toneladas diárias, dando um total ao ano de aproximadamente 212 mil toneladas. Este aumento do lixo se deve ao aumento populacional e também a falta de conscientização por parte da população em relação a separação adequada de resíduos. Muito lixo que é reciclável é levado ao aterro. Para a prefeitura descartar o lixo no aterro, tem um custo de R$ 147,00 por tonelada (GUARAPUAVA, 2023). Desta forma, é evidente a necessidade de soluções que contribuam com a redução do acúmulo de lixo no aterro sanitário e consequentemente, reduzam os custos de descarte e impactos ambientais na região (BRONOSKI; MENDES, 2017).

Com a finalidade de simplificar e incentivar a coleta seletiva foi iniciado em Guarapuava - PR, no ano de 2021 a implementação de lixeiras coletivas na forma de contêiner para o descarte separado de resíduos recicláveis e orgânicos, ilustrado na Figura 1. Com isso, a intenção é concentrar uma certa quantidade de lixo reciclável e orgânico para facilitar a coleta semanal realizada pela prefeitura, tanto a coleta seletiva como a coleta de lixo em geral. E também, esta política de concentração de lixo auxilia os catadores autônomos que tem a coleta seletiva como trabalho e fonte de renda, levando os recicláveis para a destinação correta, ou seja, para os centros de reciclagens (SOUZA; PAULA; PINTO, 2011).



Figura 1 – Exemplo das lixeiras coletivas na forma de contêiner
Fonte: Autoria própria (2025).

[1] Disponível em https://www.iswa.org/

A solução de contêiner resolve o problema de identificação do lixo reciclável, que outrora era colocado nas lixeiras particulares das residências sem nenhuma identificação, muitas vezes confundido com lixo comum e indo parar em aterros. Também resolve o problema de residências que não possuem lixeiras, sendo o lixo descartado inapropriadamente nas calçadas, podendo ser revirado por animais de rua. Com o descarte individual do lixo na frente da residência, pode ocorrer do separador não se atentar ao dia em que passa o caminhão da coleta seletiva e disponibilizar o lixo orgânico ou reciclável em um dia qualquer. Esta falta de informação dificulta o trabalho da coleta seletiva municipal e também dos catadores autônomos. Muitas vezes, os catadores precisam analisar o conteúdo dos sacos de lixo a fim de verificar se contém lixo reciclável. Uma alternativa seria identificar o lixo orgânico e reciclável por meio de sacos com cores diferentes, de preferência bio-degradáveis, ou mesmo a lixeira com alguma placa descritiva do tipo de lixo, podendo tal material ser doado pela prefeitura ou obtido por meio de empresas parceiras. Desta forma, reduziria a chance de um saco com conteúdo reciclável não ser devidamente identificado para tal e ser levado como lixo comum.

O descarte consciente do lixo reciclável em contêiner de forma coletiva tem algumas vantagens em relação ao descarte individual em residência de forma identificada. Uma vantagem é a otimização do tempo de coleta, uma vez que o lixo estará concentrado em alguns pontos. Também, há uma outra vantagem, por se tratar de um repositório, o lixo pode ficar armazenado por alguns dias, isso independe do dia da semana que o caminhão passa coletar e também, não polui visualmente a fachada das casas com sacos de lixos disponibilizados nas lixeiras ou calçadas.

No entanto, nem sempre uma solução agrada todos os envolvidos. No caso dos contêiner, eles apresentam um custo elevado para compra e também exige manutenção, o que inviabiliza a curto prazo atender todos os bairros de uma cidade de forma confortável. Ademais, tais estruturas ocupam espaço às margens das calçadas que atrapalham o tráfego ou o estacionamento de veículos, principalmente se as ruas forem estreitas ou então, de alto fluxo de veículos. E ainda, se disponibilizado na frente de alguma residência, atrapalha na circulação dos separadores e também polui visualmente. Além disso, se a coleta não for feita com frequência, os contêiner de lixo orgânico passam a emitir mau cheiro, que já é uma das principais reclamações de separadores onde o sistema foi implantado. Por conta disso, as lixeiras geralmente são colocadas na frente de terrenos baldios ou então, em regiões com menos circulação de pessoas e mais distante de algumas casas, fazendo com que alguns indivíduos necessitem andar algumas quadras ou utilizar de automóvel para poder depositar o material na lixeira mais próxima. Isto gera um grande problema de acessibilidade e prejudica pessoas idosas ou com problemas de locomoção. E a dificuldade para se descartar o lixo ainda é aumentada quando está chovendo.

Em ambas as situações, tanto no descarte coletivo como no individual, o fator pessoa está presente e consequentemente, a questão da conscientização e informação para o descarte correto do lixo. Mesmo que sejam disponibilizados meios para separação organizada do

lixo, tanto em contêiner ou por meio da doação de sacos bio-degradáveis com cores identificáveis do tipo de lixo, ainda pode ocorrer de separadores agirem de má-fé ou desinformada, levando a continuação do descarte errado do lixo. Neste caso, no sentido de não separar adequadamente ou misturar o lixo reciclável com orgânico ou ainda, disponibilizar em contêiner errado ou em saco com a cor inapropriada. Mesmo que tais recursos sejam disponibilizados de forma gratuita para a população, não se pode considerar que todos irão aderir fielmente a esta política. Portanto, é preciso um meio de identificar realmente os separadores interessados em participar deste processo de reciclagem e fazer com que o lixo gerado por estes separadores tenha o destino da reciclagem. Com a adesão de mais pessoas a esta política, cada vez mais lixo será reciclado e cada vez menos lixo irá para os aterros sanitários.

Uma solução para tornar mais eficiente a coleta seletiva do lixo seria unir as vantagens do descarte coletivo com o individual, resolvendo os problemas supra-citados de cada categoria de descarte. Para tanto, esta solução precisaria da participação em massa dos catadores autônomos, a fim de estreitar, por meio da comunicação, a relação com os indivíduos, que geram e separam o lixo reciclável. A comunicação entre estes lados precisa existir para que o descarte e coleta ocorra de forma mais organizada, eficiente e econômica. E esta comunicação seria possível de ocorrer por meio do desenvolvimento de um aplicativo conectado à Internet para ser usado pelos interessados.

Com isso, um separador poderia separar e armazenar o lixo reciclável em sua residência e quando tivesse uma quantidade que achasse suficiente, poderia comunicar aos catadores para virem retirar o lixo. Certamente, caso seja um dia em que o caminhão de coleta seletiva passará em sua rua, caso ele saiba o horário, poderá a contraponto, fazer uso deste serviço (GUARAPUAVA, 2023). Mas como alternativa, ele poderia entregar o lixo pessoalmente a um catador após um agendamento a um intervalo de horário ou então disponibilizar externamente à residência, mas informando a localização aos catadores. Assim, o separador teria certeza de que o seu esforço em separar o lixo não foi em vão, ou seja, que o seu lixo teve realmente o destino esperado, não sendo levado como lixo comum. No entanto, esta solução tecnológica também contribui para o melhor uso dos processos existentes de coleta de lixo, apresentando avisos claros sobre o dia e preferencialmente horário e rota do caminhão municipal de coleta seletiva e dicas para melhor identificar visualmente o lixo depositado externamente à residência para facilitar a coleta.

Por outro lado, a solução tecnológica também auxiliaria no trabalho dos catadores autônomos. O Movimento Nacional dos Catadores de Materiais Recicláveis (MNCR)[2] afirma que em 2021 haviam cerca de 800 mil catadores em atividade no Brasil, atuando em coletas, separações e comercialização de resíduos recicláveis, sendo eles responsáveis por cerca de 90% de toda a coleta realizada.

Apesar de grande importância para a sociedade brasileira, os catadores enfrentam grandes dificuldades para exercer seus serviços de forma adequada. Por exemplo, em inúmeros

[2] Disponível em https://www.mncr.org.br/

casos o deslocamento do material coletado é feito a pé, movimentando o veículo de transporte da carga totalmente com sua própria força, em subidas e descidas, muitas vezes com cargas pesadas em vias de tráfego difícil após períodos excessivos de trabalho, sem descanso ou remuneração suficiente, expostos a condições climáticas, perigos no trânsito, e muitas vezes, com problemas de saúde. Nestas condições e sem uma logística orientada de onde encontrar o lixo a ser coletado, eles andam distâncias longas e muitas vezes, de forma desnecessária, se comparada a uma situação em que eles tivessem a piori a informação da localização do lixo a ser coletado (JUNIOR *et al.*, 2013). Além disso, mesmo que um catador geralmente siga uma rota, ele pode passar em horários que não coincidam com a presença do lixo, ou seja, ele pode passar em uma rua em um certo horário, mas por falta de comunicação com os separadores, os mesmos, disponibilizam o lixo após a sua passagem. Esta situação leva a um desperdício de esforços do catador, de tempo e financeiro. Em termos financeiros, os catadores são os menos favorecidos, visto que a maior parte do lucro fica com as grandes recicladoras (PISANO; DEMAJOROVIC; BESEN, 2022). Neste sentido, é importante dar uma maior atenção a este setor tão importante para a sociedade. Em uma visão global, se este setor tiver o seu processo de trabalho aperfeiçoado, leva o aumento da eficiência e consequentemente ao aumento do volume de lixo destinado corretamente à reciclagem, favorecendo toda a sociedade e o meio-ambiente.

Deste modo, a solução computacional desenvolvida auxilia os catadores na realização de seus serviços. Como exemplo, a aplicação oferece orientações geográficas para melhorar a escolha da rota a ser seguida, facilitando a localização do lixo disponibilizado por separadores e otimizando o volume coletado de recicláveis em menor tempo.

Vale salientar que, o lixo pode ser disponibilizado externamente à residência de um separador, ou seja, na lixeira ou calçada, ou então, internamente ao terreno da residência ou condomínio, quando é preciso autorização do separador para a coleta. Em ambos os casos, o catador terá a localização do resíduo a ser coletado, podendo inclusive ter a informação sobre o tipo do lixo (ex. papel, vidro, pilhas), caso informado pelo separador. Porém, a coleta do lixo disponibilizado na modalidade interna somente ocorrerá quando solicitado pelo separador ou responsável pelo lixo. Neste processo, o separador pode informar a quantidade, tipo e um horário para a realização da coleta. Com estas informações, os catadores mais próximos poderão se candidatar para a coleta, o separador escolhe um entre os candidatos e então, a entrega é agendada. Ainda nesta modalidade, o separador pode oferecer doações de produtos que ainda podem ser úteis para uso (ex. um eletrodoméstico ou louça usada) ou então, propor a retirada de lixo que não é de interesse natural ao catador por não ser reciclável, volumoso ou pesado (ex. restos de árvores, material de construção ou eletrodomésticos estragados), via pagamento de um valor a ser negociado diretamente com o catador, agregando uma renda extra ao catador

A aplicação também torna o trabalho mais divertido e compensador, utilizando práticas de gamificação para pontuação pela coleta de quantidades, podendo pontuar diferente para cada categoria de reciclável coletado, buscando incentivar a coleta de tipos de lixo não atraentes aos catadores, tal como o vidro e pilhas. Esta pontuação adquirida pode levar a apresentação

de uma tabela geral de classificação por pontuação, separada por diferentes períodos, a fim de gerar uma competição saudável entre os catadores e por fim, a troca da pontuação adquirida por produtos oferecidos pela prefeitura em parceria com produtores rurais ou empresas.

Contudo, a solução computacional desenvolvida já traz benefícios para todos os envolvidos no processo de geração de lixo e coleta de recicláveis. Os separadores separadores de lixo reciclável se beneficiarão por terem uma percepção maior de que o lixo devidamente separado foi destinado corretamente para os centros de reciclagem. Os catadores terão os benefícios de otimização de rotas, premiação pelo desempenho através dos programas de parceria com a prefeitura e renda extra por serviços de transporte de outros resíduos via negociação direta com os separadores. Evidentemente, os catadores passam a exercer um papel ainda maior de destaque neste processo de coleta de lixo de um município, por estarem em comunicação direta com os separadores. E por fim, a prefeitura que terá mais uma ferramenta para auxiliar no melhoramento da coleta de lixo reciclável, trabalhando de forma integrada ao processo existente, podendo levar mais informações e atualização aos separadores, tal como o horário dos caminhões de coleta seletiva do município, e incentivando uma modalidade que une as vantagens do descarte coletivo em contêiner (i.e. armazenamento e separação em casa de uma quantidade razoável) e individual (i.e. descarte na lixeira particular por meio de um aviso ao catador ou entrega direta ao catador), mas principalmente, melhorada, inclusiva e econômica.

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver uma aplicação web no formato de um MVP, com foco em promover a coleta seletiva de resíduos recicláveis. O sistema tem como objetivo principal melhorar a comunicação direta e o engajamento contínuo entre os separadores (incluindo residências, condomínios e empresas) e os catadores de materiais recicláveis. Por meio dessa aplicação, busca-se aumentar a eficiência no processo de reciclagem e valorizar o trabalho dos catadores.

1.1.2 Objetivos específicos

- Permitir que o separador informe a localização do lixo reciclável em ambiente interno ou externo.
- Permitir que o separador escolha um catador e agende a coleta de lixo reciclável em ambiente interno.
- Permitir ao separador avaliar os serviços dos catadores.

- Apresentar aos interessados a localização do endereço dos resíduos recicláveis em mapa ou em texto.
- Promover práticas de engajamento aos usuários catadores com o uso de práticas de gamificação, pontuação e classificação por meio da pontuação por períodos.
- Permitir o débito da pontuação ao trocar por produtos oferecidos pela prefeitura ou empresas parceiras.
- Emitir relatórios para a prefeitura da cidade.
- Disponibilizar avisos e informações educacionais sobre a reciclagem.

1.2 Justificativa

A introdução aborda a problemática do descarte inadequado de resíduos, destacando os desequilíbrios ambientais e sociais causados por essa prática. No contexto específico do município de Guarapuava - PR, uma solução proposta é a criação de um aplicativo, visando facilitar a comunicação entre os separadores, os catadores e a prefeitura. Esse aplicativo permite que os separadores realizem a separação e o armazenamento dos materiais recicláveis em suas residências, notificando os catadores quando houver uma quantidade suficiente para coleta.

Além disso, o aplicativo fornece informações importantes sobre o processo de coleta seletiva, como os dias e horários de coleta, orientações visuais para identificação correta dos resíduos e até mesmo orientações geográficas para que os catadores possam otimizar suas rotas. Essa solução também oferece uma oportunidade para aplicar os conhecimentos adquiridos durante a graduação, utilizando as tecnologias aprendidas de maneira prática e efetiva.

1.3 Estrutura do trabalho

Este trabalho está organizado em sete capítulos, conforme descrito a seguir:

No Capítulo 1, apresenta-se a introdução do projeto, destacando seus objetivos, motivações e a importância do tema abordado.

No Capítulo 2, é apresentada a fundamentação teórica que embasa o trabalho, incluindo conceitos relevantes e uma descrição das tecnologias utilizadas no desenvolvimento e na gestão do projeto.

No Capítulo 3, são discutidos os sistemas similares existentes, destacando suas funcionalidades e características relevantes que possam contribuir para o desenvolvimento deste projeto.

O Capítulo 4 detalha os materiais e métodos utilizados, descrevendo o processo de desenvolvimento do sistema com base na metodologia escolhida.

No Capítulo 5, são apresentadas a análise e o projeto do sistema, incluindo a modelagem dos requisitos, os diagramas arquiteturais e as principais decisões de design adotadas.

No Capítulo 6, são descritos os detalhes do desenvolvimento do sistema, abordando a implementação, as funcionalidades principais e as integrações realizadas.

Por fim, no Capítulo 7, são apresentadas as considerações finais do projeto, uma avaliação dos resultados obtidos e as perspectivas para trabalhos futuros.

2 REFERENCIAL TEÓRICO

A reciclagem desempenha um papel crucial na preservação do meio ambiente, na conservação dos recursos naturais e na redução da poluição causada pelo descarte incorreto de resíduos (ANDRADE; FERREIRA, 2011). A coleta seletiva é um componente fundamental da gestão adequada de resíduos, permitindo a separação dos materiais recicláveis dos resíduos comuns e facilitando o processo de reciclagem.

No contexto do município de Guarapuava, como mencionado no Capítulo 1, a falta de conscientização da população em relação à separação adequada dos resíduos e o aumento populacional resultaram em um acúmulo significativo de lixo no aterro sanitário, gerando custos elevados de descarte e impactos ambientais negativos. Além disso, a infraestrutura atual de coleta seletiva apresenta limitações, como a distância entre as lixeiras coletivas e algumas residências, a falta de informação sobre os horários de coleta e a possibilidade de descarte incorreto.

Diante dos desafios enfrentados pela coleta seletiva em Guarapuava, a implementação de um aplicativo móvel específico para aprimorar esse processo se mostra uma solução promissora. Por meio do fornecimento de informações, recursos de localização e mecanismos de incentivo, espera-se que o aplicativo possa engajar a população, facilitar a separação adequada dos resíduos e melhorar a eficiência da coleta seletiva, contribuindo assim para a preservação ambiental e a sustentabilidade do município.

Os aplicativos móveis têm se mostrado eficientes na otimização de processos e na melhoria da comunicação entre diferentes atores envolvidos em determinadas atividades. No contexto da coleta seletiva, eles podem desempenhar um papel fundamental na conscientização da população, na facilitação da separação adequada dos resíduos e na comunicação entre os separadores, catadores autônomos e a prefeitura.

Ao desenvolver uma aplicação, é importante considerar diversas ferramentas disponíveis no mercado, como linguagens de programação, *frameworks* e bibliotecas que auxiliam os programadores no desenvolvimento. A escolha dessas tecnologias impacta diretamente no processo de desenvolvimento, na qualidade e no tempo de implementação da aplicação. Portanto, é fundamental definir quais tecnologias serão utilizadas, levando em consideração o sucesso da aplicação. Além disso, é importante adotar boas práticas de programação e utilizar técnicas que auxiliem na manutenção e integridade da aplicação.

Nesse capítulo, serão apresentadas tecnologias que contribuirão para o desenvolvimento do projeto, incluindo ferramentas e metodologias que serão utilizadas para alcançar resultados promissores.

2.1 Processo de desenvolvimento

O desenvolvimento de uma aplicação abrange todo o período em que a mesma está sendo construída. Durante esse processo, é essencial utilizar práticas e metodologias que ajudem a gerenciar e organizar todas as etapas, especialmente em projetos desenvolvidos em equipe. Trabalhar em equipe requer uma maior organização, garantindo que todos os membros compreendam exatamente o que precisa ser feito. O objetivo final é entregar produtos de qualidade dentro dos prazos estabelecidos. Nesse contexto, esta seção apresenta algumas abordagens e técnicas que auxiliam no gerenciamento, organização e desenvolvimento de uma aplicação.

2.1.1 Metodologia ágil *Scrum*

O método *Scrum* trata- se de uma metodologia frequentemente utilizada por equipes de profissionais desenvolvedores, sendo usado desde o início de 1990. O trabalho é dividido em ciclos chamados de *Sprints*, que geralmente têm uma duração fixa de duas a quatro semanas em que uma versão potencialmente utilizável do produto irá ser criada, assim uma nova *Sprint* começa imediatamente após a conclusão da anterior. O *Scrum* define também um *Team Scrum*, que consiste em um *Scrum Master*, um *Product owner* e *Developers*. O *Product Owner* é o responsável por fornecer novos requisitos juntamente com a ordem que devem ser executados, além de avaliar a qualidade final das entregas. O *Scrum Master* é responsável por garantir que o *Scrum* seja entendido e aplicado. A gestão das atividades é realizada por meio de reuniões de acompanhamento, conhecidas como *Daily Scrum*. Nesses encontros, a equipe compartilha o que foi feito desde a última reunião, o que será feito até a próxima. Além disso, ao final de cada *Sprint* ocorre a revisão, na qual a equipe apresenta os resultados alcançados, sendo importante para validar as entregas e ajustar o plano para as próximas *Sprints*.

A fim de garantir total funcionamento, o método *Scrum* funciona a partir de dois blocos: o *Product Backlog* e o *Sprint Backlog*. O *Product Backlog* é uma lista ordenada de todas as funcionalidades, requisitos, melhorias e correções que precisam ser feitas em um produto. Ele representa todas as demandas do cliente, bem como os requisitos identificados pela equipe de desenvolvimento. O *Sprint Backlog* caracteriza-se por um conjunto de itens do *Backlog* selecionados para a *Sprint*, visando o plano da entrega do produto e assim atingir o propósito da *Sprint* (SCHWABER; SUTHERLAND, 2020).

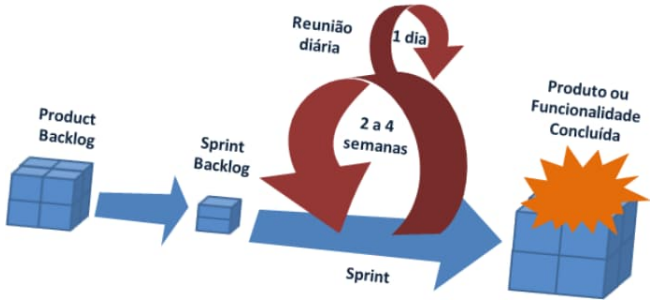


Figura 2 – Dinâmica do Scrum
Fonte: MindMaster (2023).

2.1.2 Kanban

Junto à metodologia *Scrum*, podemos utilizar no desenvolvimento deste projeto o quadro Kanban, deixando de forma clara a distribuição das tarefas. O Kanban é uma metodologia de gerenciamento visual que auxilia equipes e empresas a controlar e otimizar seu fluxo de trabalho. Originado no sistema de produção da Toyota, o Kanban se baseia em cartões visuais para representar tarefas ou itens que precisam ser realizados. Sua principal finalidade é criar um sistema de acompanhamento claro e transparente, permitindo que as equipes visualizem o status das tarefas em tempo real, o quadro Kanban é dividido em colunas que representam os estágios do processo, como "A fazer", "Em andamento" e "Concluído". Cada tarefa é representada por um cartão que é movido de uma coluna para outra conforme progride. Além de fornecer uma visão geral do trabalho em andamento, o Kanban facilita o monitoramento do fluxo de trabalho e a identificação de gargalos e atrasos. Ele promove o trabalho em equipe, a colaboração e a melhoria contínua, uma vez que as equipes podem identificar e resolver problemas de forma mais eficiente.

O Kanban pode ser aplicado em diversos contextos, desde o desenvolvimento de software até a gestão de projetos, produção industrial e até mesmo na organização pessoal. Sua abordagem flexível e adaptável permite que as equipes personalizem o método para atender às suas necessidades específicas (ARTIA, 2023).

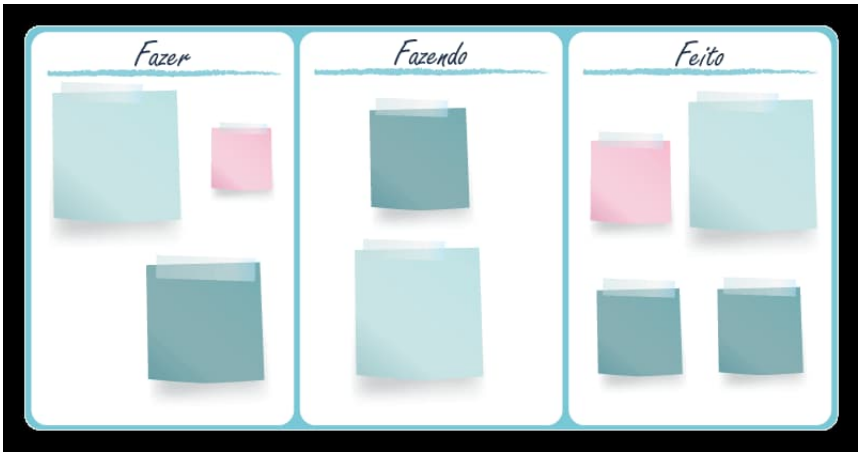


Figura 3 – Dinâmica do Kanban
Fonte: Artia (2023).

2.1.3 Desenvolvimento Orientado por Testes

Uma metodologia de teste amplamente utilizada no desenvolvimento de sistemas é a Desenvolvimento Orientado por Testes, do inglês *Test Driven Development* (TDD), que se baseia na criação de testes automatizados antes mesmo da implementação do código propriamente dito. O objetivo principal do TDD é orientar o desenvolvimento do sistema a partir dos requisitos e garantir que o código atenda a esses requisitos de forma confiável (DEVMEDIA, 2023).

O TDD segue um ciclo iterativo de três etapas: *Red-Green-Refactor* Figura 4:

- ***Red:*** Nesta etapa, o desenvolvedor escreve um teste automatizado que verifica um requisito específico que ainda não foi implementado. O teste é escrito de forma que inicialmente falhe, representando a falha do sistema em atender ao requisito.
- ***Green:*** Nesta etapa, o desenvolvedor implementa o código necessário para fazer com que o teste passe, ou seja, o sistema atenda ao requisito estabelecido. O objetivo é escrever a quantidade mínima de código necessário para que o teste seja bem-sucedido.
- ***Refactor:*** Nesta etapa, o desenvolvedor melhora o código existente, fazendo refatorações para eliminar redundâncias, melhorar a legibilidade, otimizar o desempenho, entre outras melhorias. É importante garantir que todos os testes continuem passando após as refatorações.

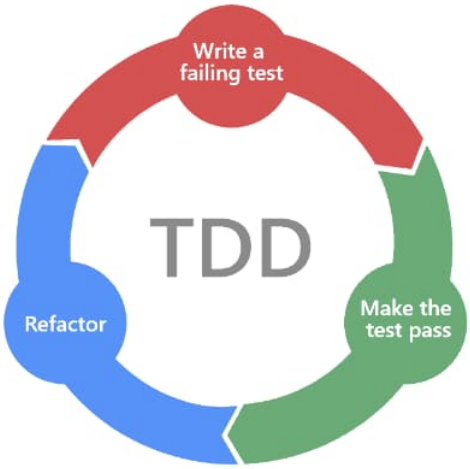


Figura 4 – Exemplo etapas TDD
Fonte: DEVMEDIA (2023).

2.2 Ferramentas de desenvolvimento

As ferramentas de desenvolvimento desempenham um papel crucial ao auxiliar os desenvolvedores na construção de sistemas computacionais. Ao utilizar ferramentas adequadas, é possível aumentar a produtividade do desenvolvimento, reduzindo a necessidade de realizar tarefas manuais e promovendo a organização e qualidade do sistema a ser desenvolvido. Nesta seção, serão apresentadas diversas ferramentas que auxiliam na construção de aplicações, contribuindo para a manutenção, organização e adoção de melhores práticas de desenvolvimento.

2.2.1 Trello

O Trello é uma ferramenta de gerenciamento de projetos online utilizando a teoria do quadro Kanban na subseção 2.1.2. Cada projeto criado é denominado de quadro e pode ser compartilhado com vários usuários. Cada quadro[1] possui colunas que contêm cartões. Os cartões são os elementos essenciais para a organização do trabalho em um quadro, utilizados para representar tarefas e histórias do usuário das metodologias ágeis, o fluxo pode ser visualizado na Figura 5. O *Trello* permite integrar ferramentas de trabalho, conectando *power-ups* que ajudam a atender a uma necessidade específica. Como exemplo, há o *Card Priority by Screenful* que permite adicionar prioridades as Histórias de Usuário e *Card Numbers by Reenhanced* que permite adicionar números de Identificação (IDs) nos cartões do quadro (TRELLO, 2023).

[1] O quadro completo pode ser acessado aqui

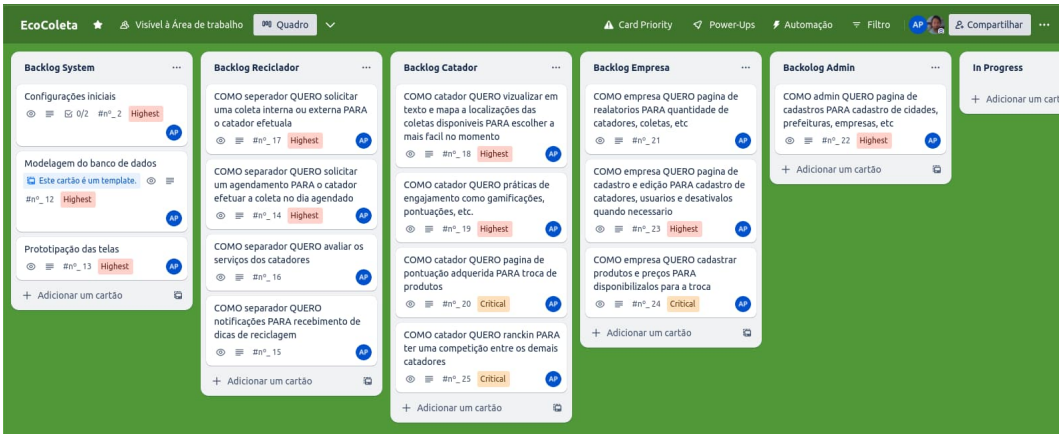


Figura 5 – Apresenta a tela da organização de um quadro
Fonte: Autoria própria (2025).

2.2.2 Git, Gitflow e GitHub

Git é um sistema de controle de versão de arquivos distribuído, gratuito e de código aberto. Foi projetado para hospedar desde projetos pequenos até grandes, mantendo a mesma velocidade e eficiência (GIT, 2023). Para gerenciar o fluxo de trabalho no *Git*, pode-se seguir um modelo chamado *Gitflow*. O *Gitflow* é um modelo abstrato do fluxo de trabalho do *Git*, que define um padrão rigoroso de ramificação com base no lançamento do projeto. Tecnicamente, o *Gitflow* estabelece um fluxo de trabalho para gerenciar as *branchs*, que são ramificações do projeto nos repositórios. Em vez de ter apenas uma *branch* principal, esse fluxo de trabalho usa duas ramificações para registrar o histórico do projeto. A *branch* principal (chamada de *main*), armazena o histórico do lançamento oficial, enquanto a ramificação de desenvolvimento (chamada de *develop*), contém o código para a próxima versão de lançamento. Cada nova funcionalidade deve ser desenvolvida em sua própria ramificação, criada a partir da *branch* de desenvolvimento. Quando uma funcionalidade é concluída, ela é mesclada na ramificação de desenvolvimento. Quando a ramificação de desenvolvimento adquire recursos suficientes para um lançamento, ela é bifurcada em uma ramificação de lançamento (chamada de *release*). Quando está pronta para ser lançada, a ramificação de lançamento é mesclada com as *branchs* principal e de desenvolvimento. Para corrigir problemas que ocorrem na versão em produção com rapidez, são criadas ramificações de manutenção ou *hotfix* baseadas na *branch* principal, uma vez que essa *branch* reflete o código em produção (ATLASSIAN, 2023). O fluxo de trabalho do *Gitflow* pode ser visualizado na Figura 6.

O GitHub é uma plataforma online de código aberto que permite hospedar projetos gratuitamente, com repositórios públicos e privados de forma ilimitada. Além disso, oferece uma série de funcionalidades que auxiliam na revisão e no gerenciamento de projetos (GITHUB, 2023). Por sua vez, o *GitHub Actions* é utilizado para automatizar, personalizar e executar fluxos de trabalho de desenvolvimento em CI e CD diretamente em projetos hospedados em repositórios do *GitHub*. O CI, que utiliza o *GitHub Actions*, oferece fluxos de trabalho que podem ser

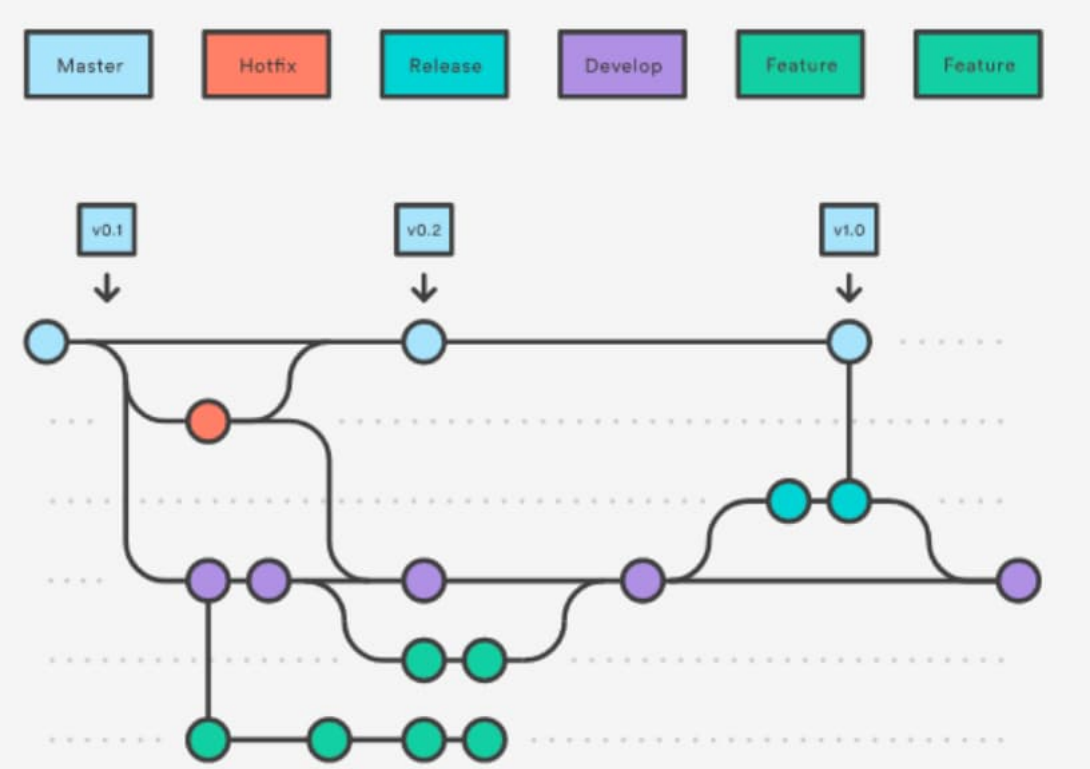


Figura 6 – Fluxo de trabalho do Gitflow
Fonte: (ATLASSIAN, 2023).

configurados quando ocorre um evento no *GitHub*, como o envio de novo código para o repositório. Dessa forma, ele pode executar os testes da aplicação. Já o CD refere-se ao lançamento automático das mudanças do repositório para a produção, o qual pode ser configurado através da mesclagem com a *branch* principal, por exemplo. O fluxo de trabalho utilizando os conceitos de CI/CD pode ser visualizado na Figura 7.

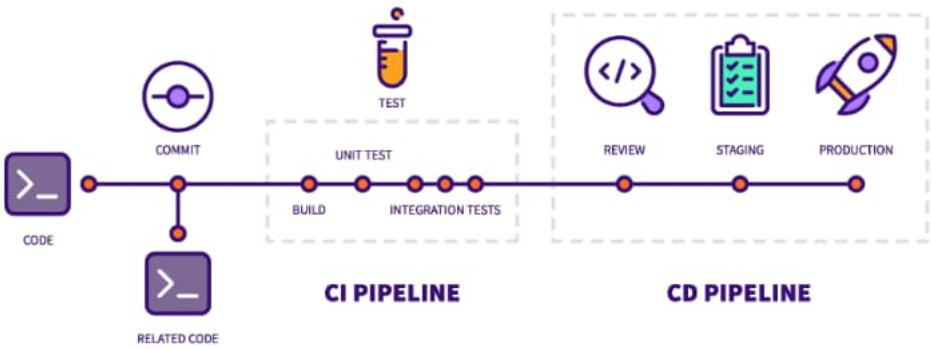


Figura 7 – Imagem do fluxo de trabalho de um projeto utilizando CI/CD.
Fonte: (GITLAB, 2023).

2.2.3 Render

Render é uma plataforma de hospedagem em nuvem que combina a simplicidade de uma Plataforma como Serviço, do inglês *Platform as a Service* (PaaS) com o desempenho e a flexibilidade de uma Infraestrutura como Serviço, do inglês *Infrastructure as a Service* (IaaS). Assim como outras soluções em nuvem, ela permite o *deploy* de aplicações de maneira rápida e fácil, sendo amplamente adotada por desenvolvedores que buscam um equilíbrio entre controle, escalabilidade e facilidade de uso.

Uma das principais vantagens do *Render* é sua flexibilidade ao suportar uma ampla gama de linguagens de programação, como *Python, Node.js, Ruby, Java* e outras. A plataforma é conhecida por oferecer integrações nativas com bancos de dados como PostgreSQL, permitindo, por exemplo, a configuração da extensão geoespacial *PostGIS* diretamente na instância do banco. Isso foi essencial para a aplicação desenvolvida, pois possibilitou operações geoespaciais otimizadas, como a geração de rotas baseadas na localização das coletas.

Além disso, o *Render* suporta tanto aplicações em contêineres (via *Docker*) quanto aplicações estáticas e dinâmicas, fornecendo escalabilidade automática com base na demanda. Isso o torna uma solução robusta para projetos de diferentes tamanhos. A plataforma também oferece recursos integrados de segurança, backup de dados e balanceamento de carga, garantindo alta disponibilidade e confiabilidade das aplicações (RENDER, 2025).

Em comparação com outras plataformas, o *Render* destacou-se durante o desenvolvimento do projeto devido à sua política de preços competitiva e ao suporte eficiente a *workflows* de CI/CD, integrando-se facilmente a repositórios de controle de versão como *GitHub* e *GitLab*. Dessa forma, o *deploy* contínuo foi automatizado, simplificando as atualizações de produção e garantindo que novas funcionalidades fossem disponibilizadas rapidamente aos usuários.

2.2.4 Flyway

O *Flyway* é uma ferramenta amplamente utilizada para o gerenciamento e controle de versões de migrações de bancos de dados relacionais. Ele permite que as alterações estruturais e dados sejam aplicados de forma ordenada e consistente, garantindo a integridade do banco de dados durante todo o ciclo de desenvolvimento e produção (SOFTWARE, 2025).

A principal função do *Flyway* é aplicar *scripts* de migração automaticamente, seguindo uma sequência ordenada e controlada por meio de versões numeradas. Cada migração é registrada internamente na tabela de metadados do *Flyway*, conhecida como *schema history table*, garantindo que apenas as mudanças necessárias sejam aplicadas e evitando a execução duplicada de *scripts*. Isso facilita o rastreamento das modificações e a reversão, quando necessário.

O processo de migração envolve as seguintes etapas:

- **Versionamento**: Cada *script* de migração recebe um identificador único, geralmente numérico, que indica a sequência em que deve ser aplicado.
- **Aplicação de migrações**: Durante a execução, o *Flyway* verifica quais migrações ainda não foram aplicadas ao banco de dados e as executa em ordem cronológica.
- **Controle de rollback**: Embora o *Flyway* não ofereça suporte automático a rollback, ele permite que sejam criados scripts manuais de reversão, caso necessário.

Além disso, o *Flyway* oferece suporte a diversos bancos de dados, incluindo *PostgreSQL*, *MySQL*, *Oracle* e *SQL Server*, e pode ser integrado a diferentes ferramentas de CI/CD, garantindo que o processo de atualização do banco de dados esteja sincronizado com o pipeline de desenvolvimento.

2.3 Tecnologias do lado cliente

O lado do cliente, também conhecido como *front-end*, é responsável pela interface visual de uma aplicação, possibilitando a interação entre o usuário e o sistema. É nessa parte que os elementos visuais, como *layouts*, botões, formulários e outros elementos interativos, são desenvolvidos. Além disso, o lado do cliente também é responsável por enviar requisições ao lado do servidor e exibir os resultados obtidos. Nesta seção, serão abordadas algumas tecnologias de desenvolvimento de aplicações utilizadas no lado do cliente, com foco na criação de interfaces visuais e na interação com o usuário.

2.3.1 Angular e Tecnologias Relacionadas

Angular é um *framework* para desenvolvimento web amplamente utilizado, criado e mantido pelo Google. Ele oferece uma abordagem baseada em componentes para o desenvolvimento de aplicativos web modernos e escaláveis. O *Angular* adota o padrão Modelo-Visão-Controlador, do inglês *Model-View-Controller* (MVC), mas com uma abordagem mais modular e orientada a componentes. Os componentes são os blocos de construção fundamentais do *Angular*, permitindo a criação de interfaces de usuário reutilizáveis e responsivas (ANDREI, 2023).

Entre as principais funcionalidades utilizadas no projeto *EcoColeta*, destacam-se:

- **Estrutura modular**: Com a divisão clara entre módulos, serviços e componentes, o *Angular* facilita a manutenção e escalabilidade do código.
- **Angular Reactive Forms**: Utilizado para a validação dinâmica e reativa dos formulários, garantindo que os dados enviados ao backend estejam corretos e completos.

- **HttpClient**: Integrado diretamente ao *Angular*, o *HttpClient* permite a comunicação eficiente entre o frontend e a API REST, utilizando requisições Protocolo de Transferência de Hipertexto, do inglês *Hypertext Transfer Protocol* (HTTP) de forma assíncrona.
- **PrimeNG**: Um conjunto de componentes de interface de usuário avançados e personalizáveis, utilizado no *EcoColeta* para a criação de tabelas, botões, painéis e outros elementos interativos. O *PrimeNG* facilitou a construção de uma interface visualmente atraente e funcional.

Esses recursos, aliados à robustez do *Angular*, proporcionaram uma base sólida para o desenvolvimento da interface do usuário, garantindo uma experiência fluida e responsiva.

2.3.2 Progressive Web App (PWA)

Aplicativo Web Progressivo, do inglês *Progressive Web App* (PWA) é um modelo de aplicação web que oferece uma experiência de uso muito próxima à oferecida pelos aplicativos móveis, sem a necessidade de instalação no dispositivo. Esse modelo foi adotado no *EcoColeta* para garantir acessibilidade, leveza e funcionamento em diferentes condições de conectividade. As aplicações PWA são leves, responsivas e seguem os padrões necessários para oferecer funcionalidades avançadas como notificações, cache local e funcionamento offline (ANTUNES, 2023).

Entre os principais recursos da tecnologia PWA usados no *EcoColeta*, destacam-se:

- **Service Worker**: Responsável por interceptar e armazenar requisições no lado do cliente, permitindo o acesso rápido aos dados e garantindo o funcionamento offline da aplicação.
- **Cache API**: Utilizada para armazenar localmente recursos estáticos e dinâmicos, otimizando a velocidade de carregamento das páginas mesmo em conexões instáveis.
- **Web App Manifest**: Um arquivo Notação de Objetos JavaScript, do inglês *JavaScript Object Notation* (JSON) que define os metadados essenciais da aplicação, como o ícone, o nome da aplicação e as configurações de inicialização, proporcionando uma experiência semelhante à de aplicativos nativos.
- **Segurança com HTTPS**: Todo o tráfego é protegido por meio do protocolo Protocolo Seguro de Transferência de Hipertexto, do inglês *Hypertext Transfer Protocol Secure* (HTTPS), garantindo a privacidade e segurança dos dados do usuário.

Ao adotar o conceito de PWA, o *EcoColeta* assegura maior confiabilidade, mesmo em ambientes de baixa conectividade, e oferece uma experiência contínua e imersiva aos seus usuários.

2.3.3 Google Maps e Serviços de Geolocalização

O *Google Maps* é uma plataforma amplamente utilizada para serviços de mapeamento e geolocalização, oferecendo uma API robusta que permite a integração de funcionalidades de mapas interativos, geocodificação e cálculo de rotas em aplicações web e móveis. Sua versatilidade facilita a implementação de recursos que utilizam informações geográficas para melhorar a experiência do usuário (Google Maps, 2025).

Entre as principais funcionalidades técnicas do *Google Maps* utilizadas no projeto, destacam-se:

- **Mapas interativos**: A API do *Google Maps* foi utilizada para exibir mapas dinâmicos e interativos, facilitando a navegação e a visualização das coletas disponíveis em tempo real.
- **Geocodificação e geocoding reverso**: A conversão de endereços físicos para coordenadas geográficas foi implementada através da *Geocoding API*, permitindo que pontos de coleta sejam marcados com precisão no mapa. O geocoding reverso foi utilizado para exibir o endereço textual correspondente às coordenadas.
- **Cálculo de rotas otimizadas**: A *Directions API* foi usada para gerar rotas otimizadas que conectam os pontos de coleta mais próximos, minimizando o tempo e a distância percorrida pelos catadores.
- **Atualização em tempo real**: A integração do *Google Maps* com os dados geoespaciais do banco de dados, via *PostGIS*, permite que os mapas sejam atualizados em tempo real, refletindo a posição atual dos catadores e as coletas concluídas.
- **Controle de camadas e marcadores**: Diferentes camadas de informações, como marcadores de pontos de coleta ativos, foram configuradas na interface, permitindo a sobreposição e personalização da visualização dos dados no mapa.

2.4 Tecnologias do lado servidor

O lado do servidor, também conhecido como *back-end*, é responsável por receber requisições e executar processamentos relacionados diretamente às regras de negócio, como armazenar dados, obter informações, realizar cálculos, definir rotas, agendamentos e outras funcionalidades. O lado do servidor está intimamente ligado ao lado do cliente, que solicita informações a ele e aguarda o processamento dessas requisições.

O lado do servidor deve retornar o resultado dessas solicitações para o cliente que as requisitou. Nesta seção, serão abordadas as tecnologias de desenvolvimento de aplicações utilizadas no lado do servidor.

2.4.1 Docker

O Docker é uma plataforma de virtualização de contêineres que possibilita o empacotamento e a execução de aplicativos de forma isolada, garantindo sua portabilidade e escalabilidade. Ao contrário da virtualização tradicional, em que cada máquina virtual possui seu próprio sistema operacional completo, o Docker utiliza os recursos do sistema operacional hospedeiro, compartilhando o *kernel* e outros componentes como ilustrado na Figura 8. Isso torna os contêineres mais leves, rápidos e eficientes, permitindo uma melhor utilização dos recursos do servidor. Com o Docker, é possível criar, implantar e gerenciar aplicativos de maneira simples e consistente, independentemente do ambiente de execução. O uso de imagens Docker, que são modelos pré-configurados para a criação de contêineres, garante a consistência do ambiente em diferentes etapas, como desenvolvimento, teste e produção. Além disso, o Docker facilita o (CI/CD), permitindo que os desenvolvedores criem pipelines automatizados para construir, testar e implantar aplicativos de forma rápida e confiável (HOSTINGER, 2023).

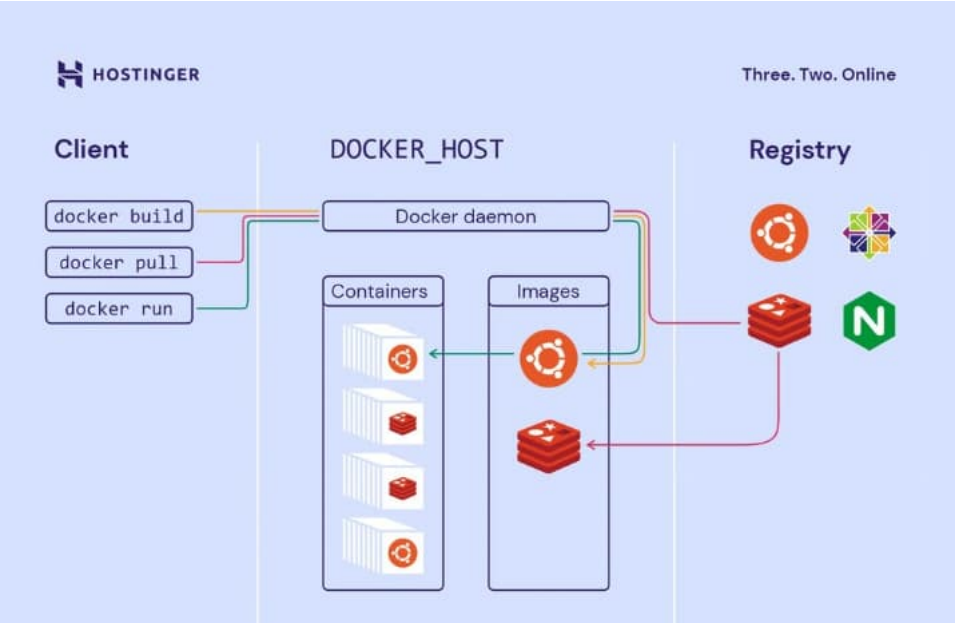


Figura 8 – Como docker funciona
Fonte: (HOSTINGER, 2023).

2.4.2 Spring Boot e Tecnologias Relacionadas

O *Spring Boot* é um *framework* Java projetado para simplificar o desenvolvimento de aplicativos robustos e escaláveis, fornecendo uma estrutura de configuração mínima e uma integração direta com o ecossistema do *Spring*. Com sua abordagem de "opinião sobre configuração", ele elimina a necessidade de configurações manuais complexas, permitindo um desenvolvimento ágil e um rápido lançamento de aplicativos (BOOT, 2023).

Entre os principais recursos oferecidos pelo *Spring Boot*, destacam-se:

- **Injeção de dependência**: Por meio do *Spring IoC Container*, o *Spring Boot* gerencia as dependências de forma automática, promovendo um código desacoplado e fácil de manter.

- **API REST**: Com o suporte integrado ao módulo *Spring Web*, o *Spring Boot* facilita a criação de *APIs RESTful*. As rotas e controladores podem ser facilmente definidos utilizando anotações como `@RestController` e `@RequestMapping`, garantindo um design limpo e modular da Interface de Programação de Aplicativos, do inglês *Application Programming Interface* (API).

- **Persistência de dados com *Spring Data JPA***: O *Spring Boot* oferece integração nativa com o *Spring Data JPA*, permitindo a interação simplificada com bancos de dados relacionais. Com base no padrão API de Persistência Java, do inglês *Java Persistence API* (JPA), é possível realizar operações de *CRUD* de forma declarativa, utilizando interfaces e anotações como `@Entity` e `@Repository`.

- **Autenticação e segurança com *Spring Security***: O *Spring Boot* possui suporte integrado ao *Spring Security*, um módulo voltado à proteção de aplicações web. A segurança é implementada de forma flexível, permitindo autenticação por meio de *tokens* Tokens Web no Formato JSON, do inglês *JSON Web Tokens* (JWT) e a configuração de políticas de autorização baseadas em papéis de usuário. O uso do JWT permite a troca segura de informações entre cliente e servidor, garantindo sessões autenticadas sem a necessidade de manter estados no servidor.

- **Configurações de ambiente e *profiles***: O *Spring Boot* permite configurar diferentes ambientes (como desenvolvimento, teste e produção) por meio do uso de arquivos de configuração personalizados (`application.properties` ou `application.yml`). Isso facilita a adaptação da aplicação às necessidades de cada fase do ciclo de desenvolvimento.

- **Suporte ao *Flyway***: O *Spring Boot* pode ser facilmente integrado ao *Flyway* para o gerenciamento de migrações de banco de dados, garantindo que as alterações estruturais sejam aplicadas de forma ordenada e consistente em todos os ambientes.

Com o uso dessas tecnologias, o backend do *EcoColeta* foi desenvolvido para atender às principais necessidades de segurança, escalabilidade e flexibilidade. O *Spring Security* garantiu a proteção das rotas da API Transferência de Estado Representacional, do inglês *Representational State Transfer* (REST), enquanto o JPA facilitou o acesso ao banco de dados relacional. A autenticação baseada em JWT contribuiu para a manutenção de sessões seguras e sem estado, ideal para uma arquitetura moderna. Além disso, o suporte ao *Flyway* permitiu a aplicação controlada das alterações no banco de dados, minimizando o risco de inconsistências em produção.

2.4.3 PostgreSQL

O PostgreSQL é um Sistema de Gerenciamento de Banco de Dados (SGBD) de código aberto e gratuito, com mais de 30 anos de desenvolvimento ativo. Além de controlar o armazenamento de dados, ele gerencia o acesso, oferece compatibilidade com a linguagem Linguagem de Consulta Estruturada, do inglês *Structured Query Language* (SQL) e possui suporte às propriedades de transação Atomicidade, Consistência, Isolamento e Durabilidade (ACID). Compatível com diversos sistemas operacionais, o PostgreSQL permite replicação de servidores, gerenciamento de múltiplas conexões e suporte nativo a Camada de Soquetes Seguros, do inglês *Secure Sockets Layer* (SSL). Destaca-se pela eficiência e confiabilidade no manuseio de grandes volumes de dados. Como um SGBD objeto-relacional, o PostgreSQL oferece recursos essenciais para garantir a integridade dos dados e proporcionar segurança nas comunicações em rede (POSTGRESQL, 2023).

2.4.4 PostGIS

O *PostGIS* é uma extensão geoespacial do *PostgreSQL*, projetada para permitir o armazenamento, a manipulação e o processamento de dados espaciais. Ele amplia as funcionalidades do *PostgreSQL*, oferecendo um conjunto de tipos de dados e funções específicas para trabalhar com informações geográficas (TEAM, 2025).

Entre as principais funcionalidades do *PostGIS*, destacam-se:

- **Armazenamento de dados geoespaciais**: Com suporte a tipos de dados como *POINT*, *LINESTRING* e *POLYGON*, o *PostGIS* permite o armazenamento de informações de localização diretamente no banco de dados.
- **Consultas geoespaciais avançadas**: Funções como `ST_Distance`, `ST_Within` e `ST_DWithin` permitem a execução de cálculos de distância, verificação de proximidade e determinação de interseções entre objetos geográficos.
- **Indexação espacial eficiente**: Utilizando índices espaciais Árvore de Busca Generalizada, do inglês *Generalized Search Tree* (GIST), o *PostGIS* otimiza consultas geoespaciais, garantindo que operações envolvendo grandes volumes de dados sejam executadas de forma rápida e eficiente.

3 TRABALHOS RELACIONADOS

Atualmente, no Brasil e no mundo existem muitos aplicativos instrutivos sobre a coleta de lixo reciclável, entretanto, há poucas opções sobre o contato do coletor do material reciclável e o produtor. A seguir, são apresentados alguns sistemas disponíveis no mercado que possuem propósito similar ao objetivo dessa proposta.

3.1 ReciclaBR

Com o ReciclaBR é possível encontrar pontos de coleta seletiva próximos a sua localização, incluindo cooperativas, pontos de entrega voluntária e postos de coleta de empresas. A empresa também tem o foco em comercialização de todos os tipos de materiais recicláveis. Além disso, o aplicativo também fornece informações sobre como separar corretamente os resíduos e como fazer o descarte de materiais específicos, como pilhas e baterias (RECICLABR, 2023).

3.2 Meu Copo Eco

Meu Copo Eco é a primeira empresa no Brasil a pensar em uma solução para diminuição do uso de copos descartáveis, assim criando uma plataforma onde é possível comprar seu copo reutilizável personalizado, o aplicativo também incentiva a coleta de copos plásticos descartáveis em empresas e eventos. A cada copo coletado o usuário acumula pontos que podem ser trocados por prêmios ou doados para instituições de caridade, desde a criação da empresa foram 145 toneladas de lixo não gerados (ECO, 2023).

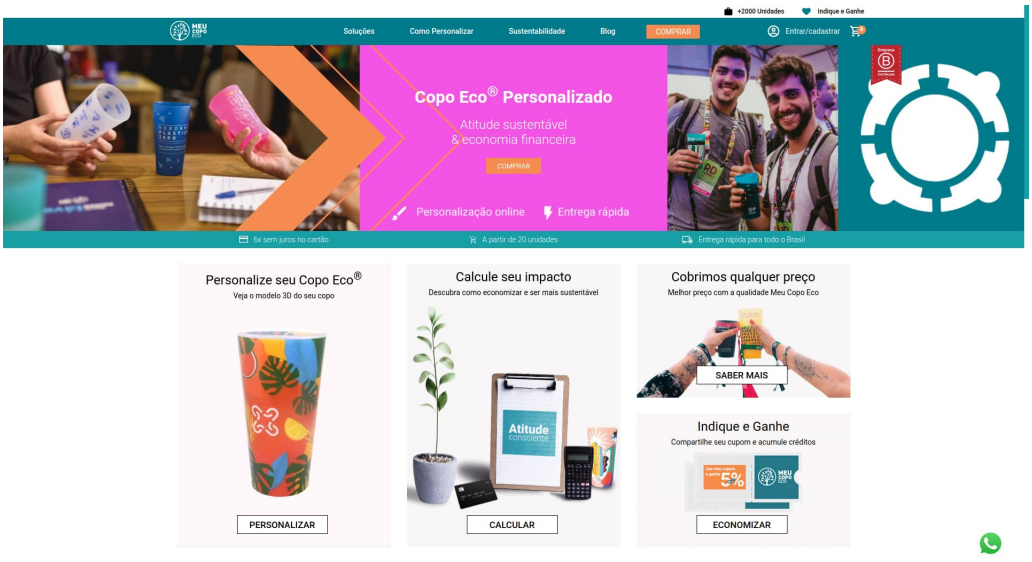


Figura 9 – Tela inicial Meu Copo Eco
Fonte: Eco (2023).

3.3 Cataki

Em território nacional, o que se destaca é o Aplicativo (APP) “Cataki” [1], produzido pela Organização não-governamental (ONG) “Pimp my carroça”, um ótimo auxiliador do catador, segundo dados do próprio site da organização o objetivo principal é trazer visibilidade e consequentemente aumentar o ganho financeiro dos catadores de materiais recicláveis por meio de arte, sensibilização, tecnologias e trabalhos coletivos. O APP desenvolvido pela ONG auxilia no contato entre o produtor de materiais e o coletor, notificando pontos de coletas, horários e tipos de materiais a serem coletados.

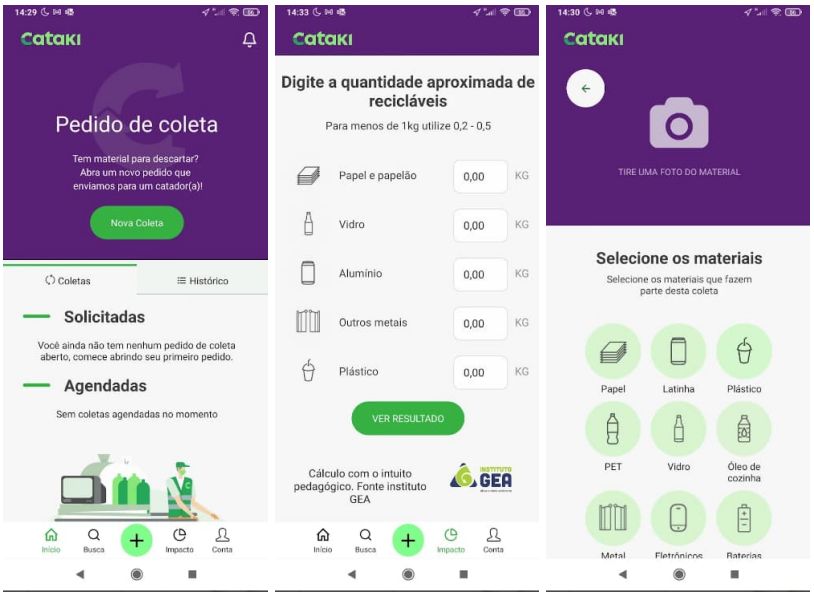


Figura 10 – Telas app Cataki
Fonte: CARROÇA (2023).

3.4 iRecycle

Nos Estados Unidos o que mais se destacou foi o aplicativo "iRecycle", produzido pela ONG "Earth911"[2] tem como missão aumentar as taxas de reciclagem dos Estados Unidos, ajudando você a classificar e a reutilizar os materiais recicláveis. O aplicativo iRecycle da dicas de quais materiais pode reciclar e procurar onde levar. A iRecycle fornece conexões de localização, telefone e site para mais de 110.000 programas de reciclagem em mais de 250.000 locais nos Estados Unidos EARTH911 (2023).

[1] Disponivel em https://pimpmycarroca.com/
[2] Disponivel em https://earth911.com/

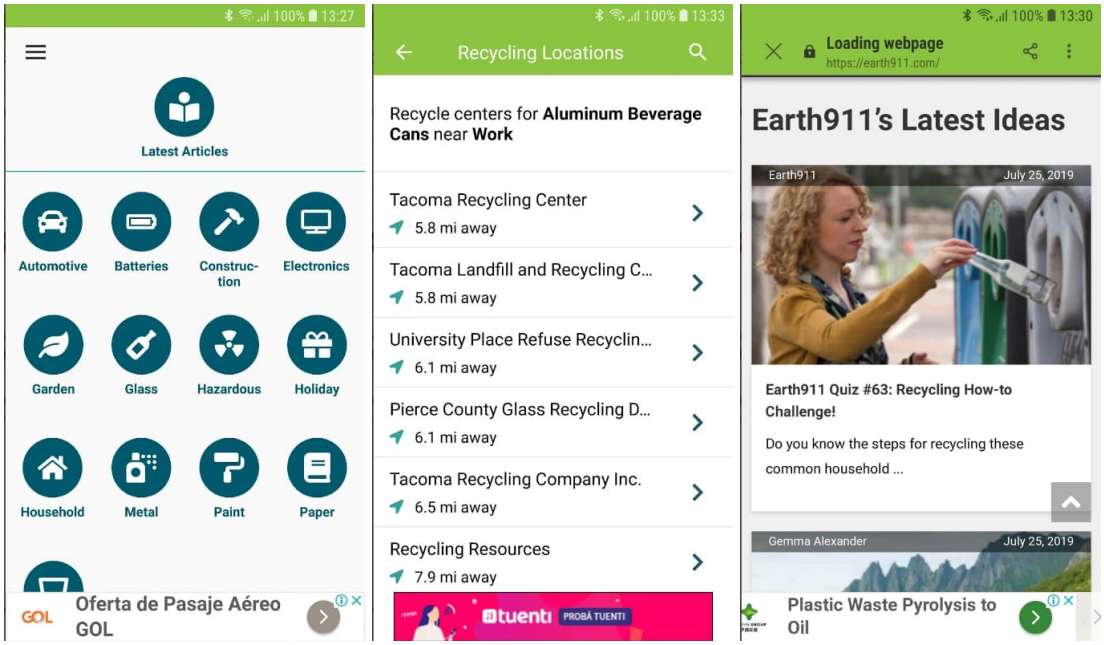


Figura 11 – Telas app iRecycle

Fonte: EARTH911 (2023).

3.5 Estudo Comparativo

Com base na análise apresentada na Seção 3, os aplicativos analisados compartilham o objetivo de aumentar a reciclagem no Brasil e no mundo. No entanto, o aplicativo Cataki se destaca por considerar a qualidade de vida do catador como um dos principais objetivos, facilitando a coleta por meio de agendamentos e recebimentos de doações.

O aplicativo proposto combina as qualidades do Cataki com novas funcionalidades, como incentivos internos para os catadores, coletas mais automatizadas, criação de rotas otimizadas para uma melhor logística e a troca de produtos por meio de parcerias.

A Tabela 1 apresenta um comparativo entre os sistemas analisados, destacando as principais diferenças entre eles.

Tabela 1 – Estudo comparativo

Funções	ReciclaBR	MeuCopoEco	Cataki	iRecycle	EcoColeta
1. Localização de pontos de coleta					
2. Dicas de reciclagem					
3. Doações de objetos					
4. Doações de dinheiro					
5. Troca de pontos por produtos					
6. Coleta mais automatizada					
7. Mobile					

Fonte: Autoria própria (2025).

4 MATERIAIS E MÉTODOS

O desenvolvimento de sistemas requer organização e a escolha adequada de ferramentas para garantir o sucesso do projeto. Neste sentido, é fundamental adotar uma metodologia que auxilie na organização dos processos e atividades, visando atender aos requisitos e entregar resultados satisfatórios.

Para este projeto, que foi desenvolvido individualmente com o apoio de professores, a metodologia incremental do *Scrum* foi utilizada como base subseção 2.1.1. O professor orientador assumiu o papel de *Scrum Master* e *Product Owner*, enquanto o aluno desempenhou o papel de *Developer*.

O desenvolvimento das atividades seguiu a abordagem de *Sprints*, que são ciclos de trabalho definidos, e foi acompanhado por meio do quadro *Trello*. Essa ferramenta proporcionará uma visão geral do projeto e seus requisitos, permitindo um acompanhamento efetivo do progresso das atividades.

Para manter um padrão na definição das histórias de usuário, foi utilizado o template *COMO...QUERO...PARA* e definidos números de IDs para cada história de usuário, utilizando a extensão *Card Numbers by Reenhanced*, a fim de facilitar a identificação e referenciação no texto. Também foi atribuída prioridade às histórias por meio da *Card Priority by Screenful*, com uma escala de 1 a 5, onde 1 representa a maior prioridade e 5 a menor. A duração de uma *Sprint* foi definida como um período de duas semanas, com o *Developer* dedicando, em média, 2 horas por dia ao projeto, totalizando 10 horas semanais de desenvolvimento.

Para a organização e gerenciamento das histórias do *backlog*, foram criadas colunas no quadro do *Trello*, nomeadas como *Backlog Separador*, *Backlog Catador*, *Backlog Empresa*, *Backlog Admin*, *Tasks*, *To Do*, *In Progress*, *Review* e *Done*, conforme mostrado na Figura 3. Essas colunas têm as seguintes funções:

- ***Product Backlog* separador, catador, empresa e admin:** contém as histórias de usuário que devem ser implementadas;
- ***Tasks*:** tarefas com funcionalidades que não estão diretamente ligadas a uma requisição de usuário;
- ***To Do*:** histórias selecionadas para a *Sprint* atual;
- ***In Progress*:** histórias que estão sendo desenvolvidas no momento;
- ***Review*:** histórias aguardando revisão;
- ***Done*:** histórias revisadas e concluídas.

Na prática, os *cards* são movidos do *Product Backlog* para *To Do*, depois para *In Progress*, iniciando a *Sprint* atual. Quando uma tarefa é finalizada, ela é movida para *Review* e, se aprovada, para *Done*.

Para a construção da aplicação, foi utilizado o *Docker* subseção 2.4.1, que facilita o desenvolvimento, a manutenção e as futuras atualizações. Durante o desenvolvimento, a aplicação foi dividida nos seguintes contêineres:

- **Contêiner WEB-BACK:** executa o servidor em Java utilizando o *Spring Boot* subseção 2.4.2. Ele concentra o código da aplicação, as regras de negócio e tecnologias do lado do servidor, como *Spring Security*, API RESTful, autenticação via JWT e persistência com JPA.
- **Contêiner WEB-FRONT:** executa o *Angular* versão 17 subseção 2.3.1, responsável pelo desenvolvimento da interface do usuário. O *PrimeNG*, também na versão 17, foi utilizado para fornecer componentes visuais avançados, como tabelas interativas, botões e gráficos.
- **Contêiner BD:** executa o banco de dados *PostgreSQL* subseção 2.4.3, utilizando a extensão *PostGIS* para cálculos geoespaciais. Durante o desenvolvimento, o banco de dados foi configurado como um contêiner no *Docker*. No entanto, no ambiente de produção no *Render*, o banco de dados foi migrado para uma instância nativa do *PostgreSQL* para garantir maior desempenho, backups automáticos e estabilidade.

No ambiente de produção, o *Render* foi utilizado para o *backend*, o *frontend* e o banco de dados. O *backend* continuou utilizando contêineres *Docker*, mas tanto o *frontend* quanto o banco de dados foram executados como instâncias nativas para garantir melhor desempenho e otimização de recursos.

Além disso, o *Angular* foi configurado para gerar uma PWA, permitindo a criação de uma aplicação web e móvel sem duplicação de código. Isso proporcionou maior portabilidade e acessibilidade, garantindo que o sistema pudesse ser acessado por diversos dispositivos.

Dessa forma, a adoção dessas tecnologias e metodologias garantiu a organização do projeto, um acompanhamento efetivo das atividades e a entrega de uma aplicação moderna, escalável e eficiente.

5 ANALISE E PROJETO DO SISTEMA

Atualmente, a aplicação apresentada neste trabalho possui as características de um PWA, tendo como principal objetivo facilitar a solicitação de coletas de lixo reciclável por meio de um aplicativo, tanto em ambientes internos quanto externos. O usuário poderá escolher um catador específico ou optar por uma seleção automática.

Dentro da aplicação, é possível informar a localização exata do lixo reciclável, seja em residências ou outros locais. Além disso, a aplicação permite a avaliação dos serviços prestados pelos catadores, bem como a visualização da localização das coletas por meio de mapas ou textos. Para incentivar a participação dos catadores, a aplicação oferece um sistema de gamificação, atribuindo pontuações que poderão ser trocadas por produtos disponibilizados pela prefeitura ou empresas parceiras.

A solução também inclui a geração de relatórios informativos para a prefeitura, auxiliando na gestão de resíduos recicláveis. Além disso, a aplicação disponibiliza avisos e conteúdos educativos à população, promovendo a conscientização sobre a importância da reciclagem. As características da aplicação são detalhadas na subseção 1.1.2.

Este capítulo apresenta a análise necessária para que haja um melhor entendimento de como ocorreu o desenvolvimento do sistema, utilizando as tecnologias e conceitos abordados no Capítulo 4.

5.1 Análise do Sistema

Para cada história, foi atribuído um grau de importância ou prioridade em uma escala de 1 a 5, onde 1 representa a maior prioridade e 5 a menor. Tal classificação foi organizada de forma decrescente, de modo que as histórias com maior grau de importância para o projeto fossem desenvolvidas primeiro.

É importante destacar que o *Product Backlog* não é estático, novas histórias podem ser adicionadas à medida que o projeto avança e os envolvidos obtêm um maior entendimento do domínio do problema. O *Product Backlog* inicial está apresentado na Tabela 2.

Tabela 2 – Histórias de usuário

ID	Prioridade	Descrição da história de usuário
17	1	COMO separador QUERO solicitar uma coleta interna ou externa PARA o catador efetuá-la
14	1	COMO separador QUERO solicitar um agendamento PARA o catador efetuar a coleta no dia agendado
18	1	COMO catador QUERO visualizar em texto e mapa a localizações das coletas disponíveis PARA escolher a mais fácil no momento
19	1	COMO catador QUERO práticas de engajamento como gamificações, pontuações, etc.
23	1	COMO empresa QUERO página de cadastro e edição PARA cadastro de catadores, usuários e desativa-los quando necessário
22	1	COMO admin QUERO página de cadastros PARA cadastro de cidades, prefeituras, empresas, etc
20	2	COMO catador QUERO página de pontuação adquerida PARA troca de produtos
25	2	COMO catador QUERO ranking PARA ter uma competição entre os demais catadores
24	2	COMO empresa QUERO cadastrar produtos e preços PARA disponibiliza-los para a troca
15	3	COMO separador QUERO notificações PARA recebimento de dicas de reciclagem
21	3	COMO empresa QUERO página de relatórios PARA quantidade de catadores, coletas, etc
16	4	COMO separador QUERO avaliar os serviços dos catadores

Fonte: Autoria própria (2025).

Durante o processo de desenvolvimento do sistema, com o melhor domínio da aplicação e uma compreensão mais profunda das necessidades do projeto, algumas histórias de usuário foram despriorizadas e outras novas foram adicionadas.

As histórias adicionadas, apresentadas na Tabela 3, refletem novas demandas identificadas ao longo do desenvolvimento, seja devido a mudanças nos requisitos ou pela descoberta de funcionalidades que agregam maior valor ao sistema.

Tabela 3 – Histórias adicionadas

ID	Prioridade	Descrição da história de usuário
26	1	COMO usuário QUERO tela de login e sessão PARA entrar na aplicação
33	1	COMO usuário QUERO tela de endereços PARA edição e cadastro de endereços
32	1	COMO catador QUERO tela de cadastro PARA se inscrever na aplicação
37	1	COMO separador QUERO tela de cadastro PARA se inscrever na aplicação
35	1	COMO catador QUERO enviar uma imagem PARA ser visualizada em meu perfil
39	1	COMO catador QUERO geração de rotas PARA minhas coletas selecionadas
41	1	COMO catador QUERO coletas marcadas no mapa
47	1	COMO catador QUERO pontuação ao finalizar coletas PARA ranking e futura troca de produtos
42	2	COMO separador QUERO catadores ativos marcados no mapa
43	2	COMO separador QUERO página de listagem de coletas atuais e anteriores PARA manipular as coletas e avaliá-las
44	3	COMO empresa QUERO página de listagem de catadores
38	3	COMO admin QUERO gráficos de quantidades de usuários
40	3	COMO admin QUERO página de listagem de empresas
48	3	COMO usuário QUERO dicas de reciclagem

Fonte: Autoria própria (2025).

5.1.1 Papéis dos Usuários no Sistema

Na aplicação EcoColeta, existem quatro papéis principais de usuários, cada um desempenhando funções específicas para garantir o bom funcionamento do sistema:

- **Separador (Resident):** É o usuário que gera os resíduos recicláveis. Ele é responsável por separar os materiais adequadamente e solicitar a coleta. O separador pode informar o tipo e a quantidade de materiais, avaliar o serviço prestado pelos catadores e acessar conteúdos educativos sobre reciclagem.

- **Catador (Waste Collector):** Desempenha o papel central na coleta de resíduos recicláveis. Ele recebe as solicitações de coleta dos separadores. Com o suporte de funcionalidades como mapas e roteirização, o catador otimiza suas rotas para aumentar a eficiência da coleta. Além disso, ele acumula pontos por boas práticas.

- **Administrador (Admin):** Esse papel é assumido por responsáveis pela gestão dos dados do sistema. O administrador pode editar e criar novos usuários, como empresas.

- **Empresa (Company):** Esse papel é geralmente assumido por prefeituras ou empresas parceiras que participam da gestão de resíduos. A empresa pode cadastrar novos catadores e gerar relatórios analíticos detalhados para monitoramento de desempenho.

5.1.2 Modelagem do Banco de Dados

Nesta seção será apresentada a modelagem do banco de dados gerada para a elaboração do projeto. O sistema gerenciador de banco de dados utilizado no desenvolvimento da aplicação foi o *PostgreSQL*, descrito no Capítulo 4. A modelagem do banco de dados foi inicialmente representada por um diagrama conceitual, contendo os atributos e relacionamentos das principais entidades do sistema[1]. A Figura 12 apresenta a versão inicial do modelo de dados. Este protótipo serviu como base para a implementação das tabelas no sistema, bem como para as iterações e ajustes ao longo do projeto.

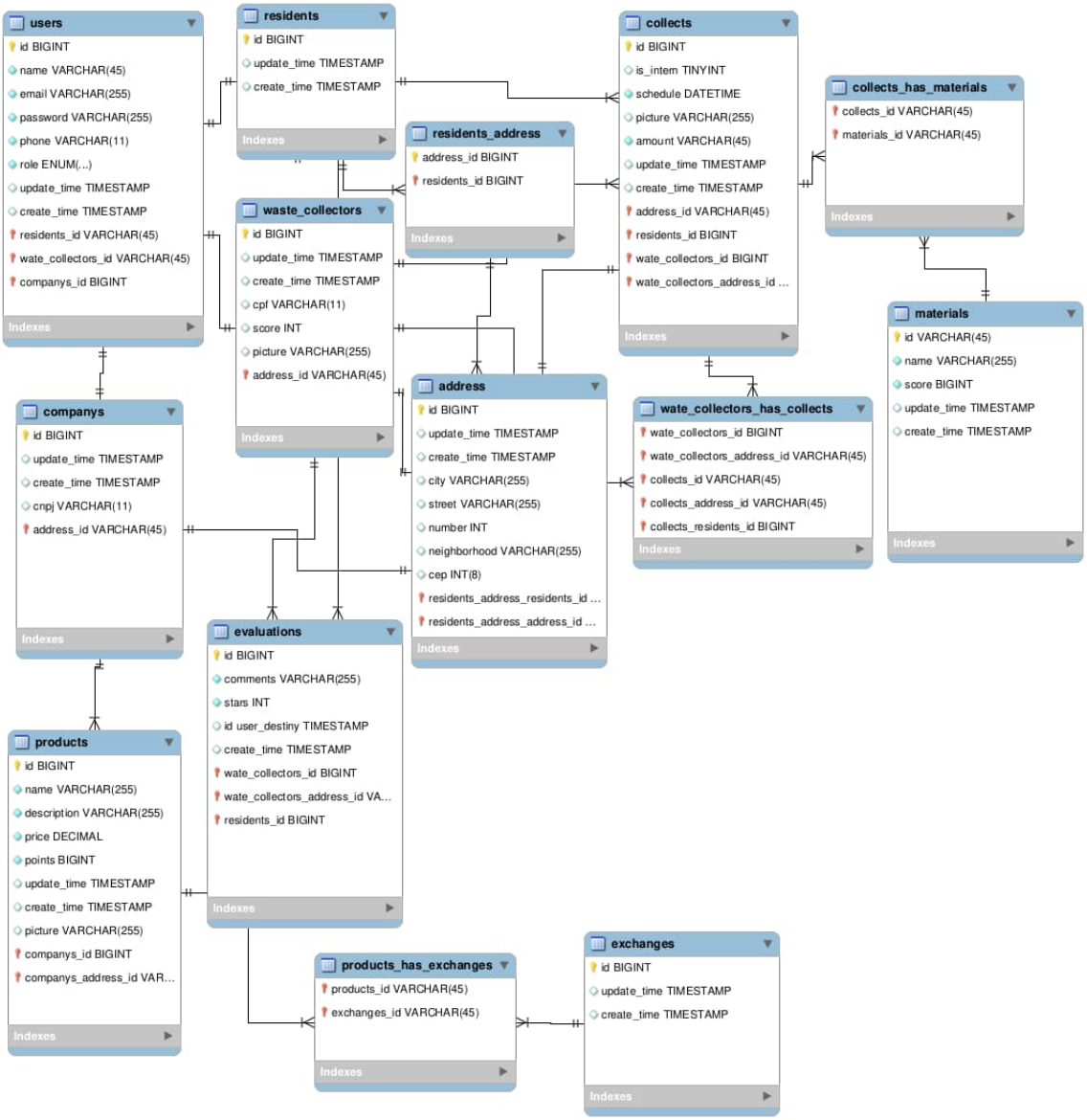


Figura 12 – Imagem inicial da modelagem do banco de dados da aplicação EcoColeta.

No modelo inicial, destacam-se as seguintes entidades e seus relacionamentos:

[1] O modelo completo do banco de dados pode ser acessado aqui

- **Tabela *users*:** Armazena informações genéricas dos usuários, como nome, telefone, e-mail, senha e *role* (papel no sistema). Dependendo do tipo de usuário (separador, catador, empresa ou administrador), dados específicos são armazenados nas tabelas especializadas: *residents*, *waste-collectors* e *company*.
- **Tabela *residents*:** Relacionada à tabela *address*, permitindo que cada separador tenha vários endereços cadastrados, contendo cidade, estado, rua, número e CEP.
- **Tabela *waste-collectors*:** Armazena dados específicos dos catadores, como CPF, foto e pontuação acumulada. Cada catador pode ter apenas um endereço associado, facilitando a navegação e o cálculo de rotas.
- **Tabela *company*:** Representa prefeituras e empresas participantes, armazenando informações como CNPJ e dados organizacionais. Inicialmente, esta tabela estava conectada à tabela *products* para gerenciar a troca de produtos por pontos.
- **Tabela *collects*:** Principal tabela de coletas, relacionando separadores e catadores em uma relação "1:n". Armazena informações como o tipo de coleta, data, fotos e quantidade coletada.
- **Tabela *materials*:** Estava relacionada à tabela *collects*, permitindo o registro dos diferentes tipos de materiais recicláveis recolhidos.
- **Tabela *evaluations*:** Armazenava a avaliação das coletas, permitindo comentários e a atribuição de estrelas aos catadores.

Durante o desenvolvimento, o modelo do banco de dados passou por ajustes significativos para atender melhor ao escopo do projeto. A versão final do diagrama de banco de dados está apresentada na Figura 13.

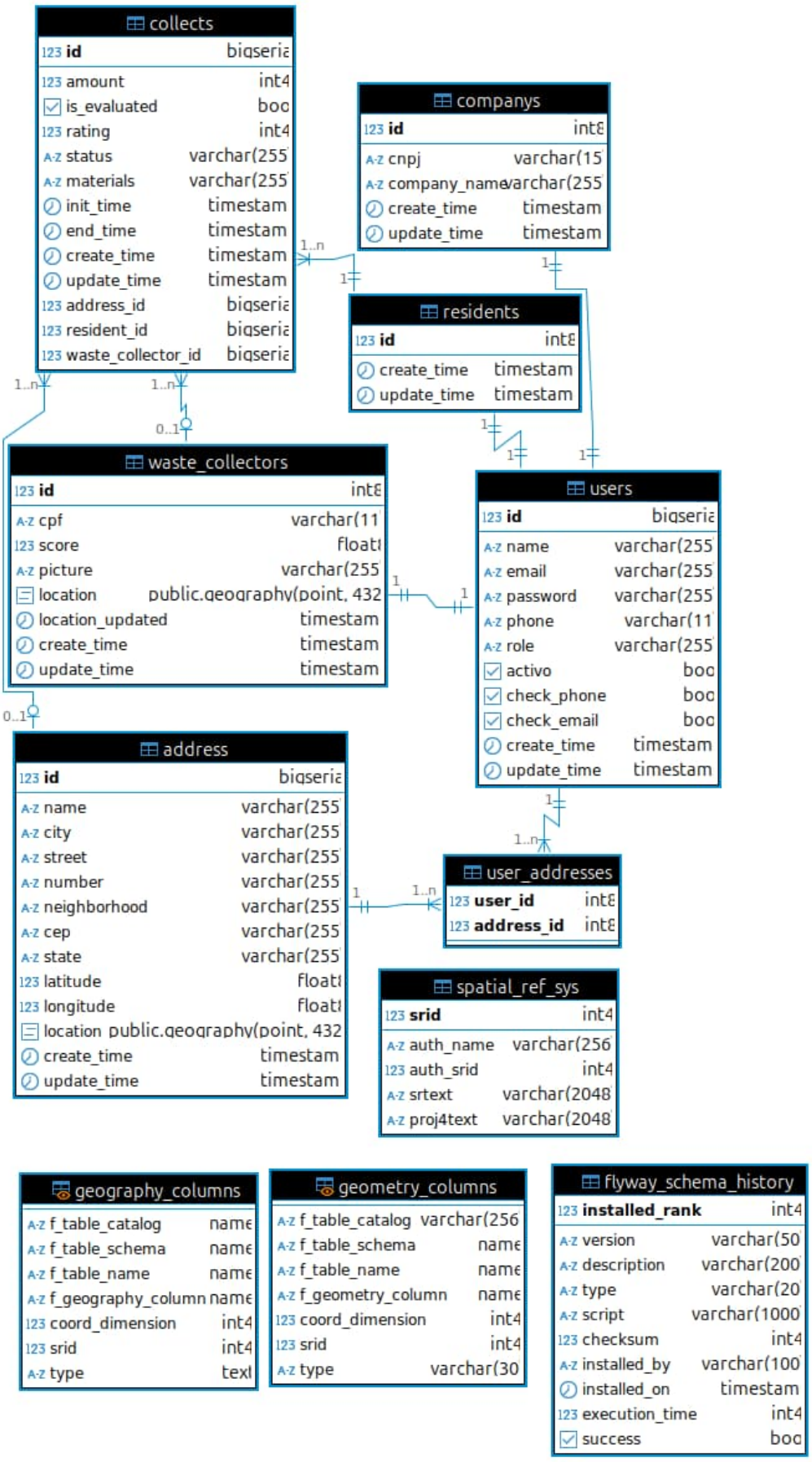


Figura 13 – Modelagem final do banco de dados da aplicação EcoColeta.

Durante as revisões do projeto, as seguintes tabelas foram removidas:

- *products*, *products_has_exchanges*, *exchanges*: A funcionalidade de troca de pontos por produtos foi considerada secundária e postergada para desenvolvimento futuro.
- *materials* e *collects_has_materials*: Os dados de materiais recicláveis foram simplificados, sendo incorporados diretamente à tabela *collects*.
- *evaluations*: As avaliações foram integradas diretamente à tabela *collects*, simplificando o modelo e reduzindo a complexidade.

Com o avanço do projeto, novas tabelas foram incorporadas ao modelo:

- **Tabela de Migração (Flyway):** Utilizada para gerenciar e versionar as migrações de banco de dados subseção 2.2.4, garantindo a atualização controlada das estruturas de dados.
- **Tabelas Geoespaciais (PostGIS):** Permitiram a execução de cálculos de proximidade e roteamento geográfico subseção 2.4.4, essenciais para a funcionalidade de exibição de rotas no aplicativo.

As mudanças realizadas foram baseadas em requisitos funcionais e melhorias de desempenho identificadas durante os testes. Por exemplo, a integração do *PostGIS* viabilizou a criação de consultas eficientes para encontrar os pontos de coleta mais próximos de um catador, utilizando funções como `ST_DWithin`. Da mesma forma, a simplificação das tabelas de materiais e avaliações reduziu a complexidade das consultas e melhorou o tempo de resposta das operações.

Com o modelo ajustado, o banco de dados final está preparado para suportar as principais funcionalidades do sistema, garantindo integridade, escalabilidade e eficiência nas operações.

5.1.3 Protótipos das Telas

Nesta seção, são apresentados os protótipos das telas desenvolvidas para a aplicação EcoColeta, fundamentais para a construção das novas funcionalidades. O desenho das interfaces foi realizado utilizando a ferramenta de design colaborativo *Figma* (INC., 2025), o que possibilitou a criação de protótipos interativos e altamente detalhados, facilitando a validação visual antes da implementação no sistema. Cada tela foi projetada com foco na usabilidade, seguindo os princípios de design responsivo para garantir uma experiência consistente em diversos dispositivos.

5.1.3.1 Tela de Login

A tela de login está ilustrada na Figura 14. Nela, os usuários podem inserir suas credenciais de forma prática. Além disso, foi implementada a opção de recuperação de senha para facilitar o acesso em casos de esquecimento.

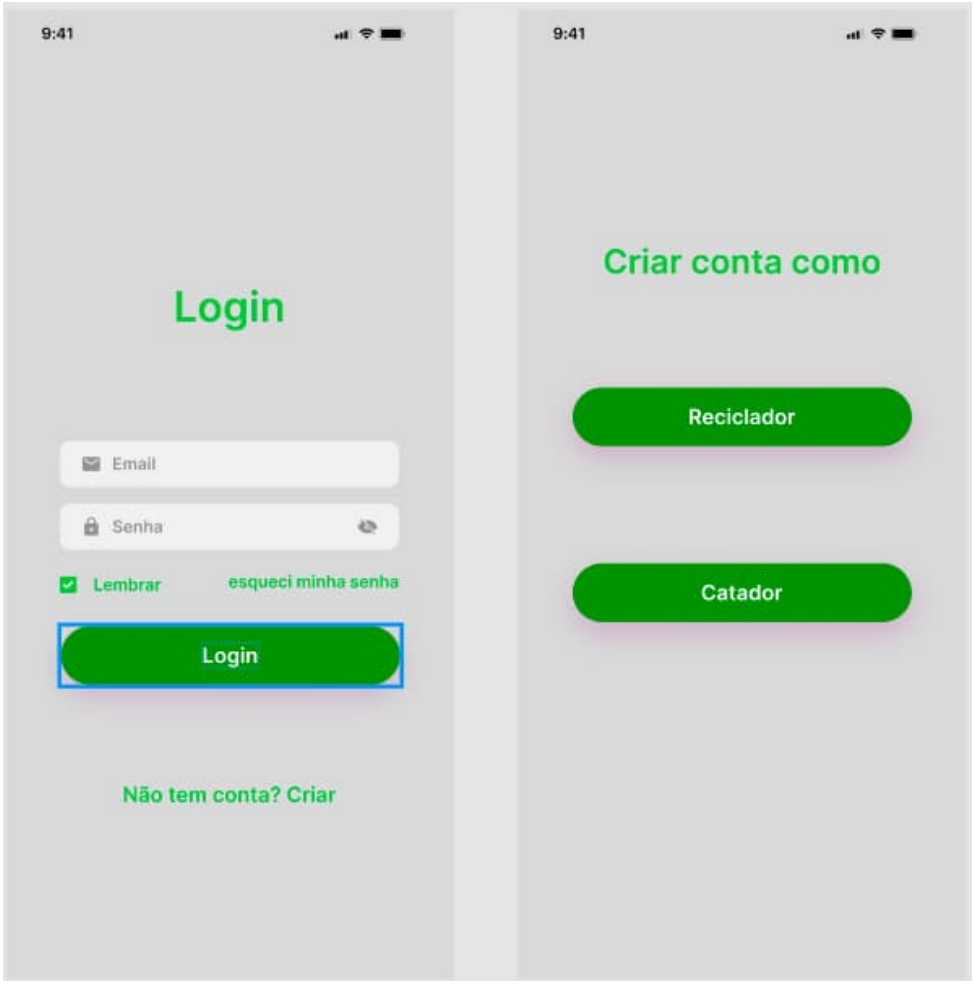


Figura 14 – Tela de login do sistema.
Fonte: Autoria própria (2025).

5.1.3.2 Formulários de Cadastro de Catador e Separador

Os formulários de cadastro, apresentados na Figura 15, foram desenvolvidos para atender aos diferentes papéis de usuários no sistema. Cada formulário contém campos específicos para coleta de informações, como nome, e-mail, endereço e senha, além de dados adicionais, como o CPF no caso dos catadores.

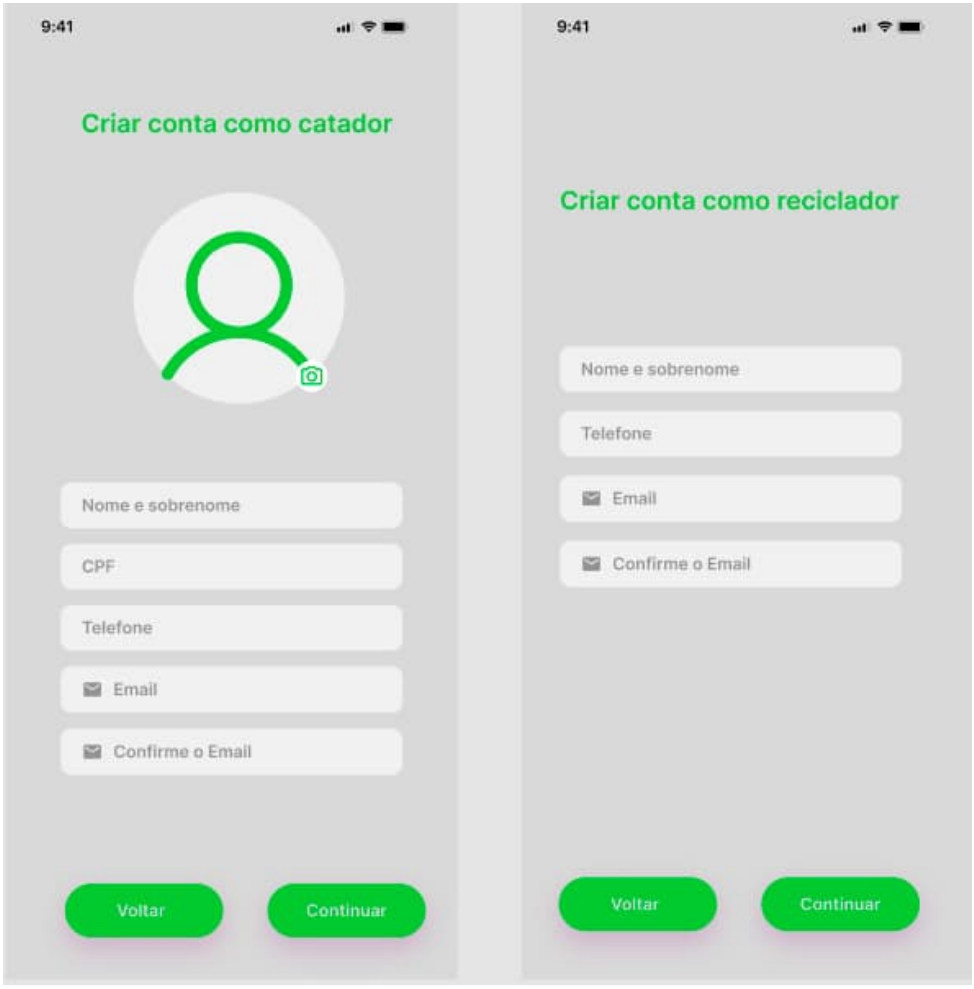


Figura 15 – Tela de cadastro de catador e separador.
Fonte: Autoria própria (2025).

5.1.3.3 Tela Inicial do Perfil de Separador e Formulário de Solicitação de Coleta

A tela inicial do separador, ilustrada na Figura 16, oferece acesso rápido às principais funcionalidades, como visualizar coletas e solicitar novas coletas. Já o formulário de solicitação de coleta, permite que o usuário informe o tipo de resíduos, a quantidade e o endereço onde a coleta deverá ser realizada.

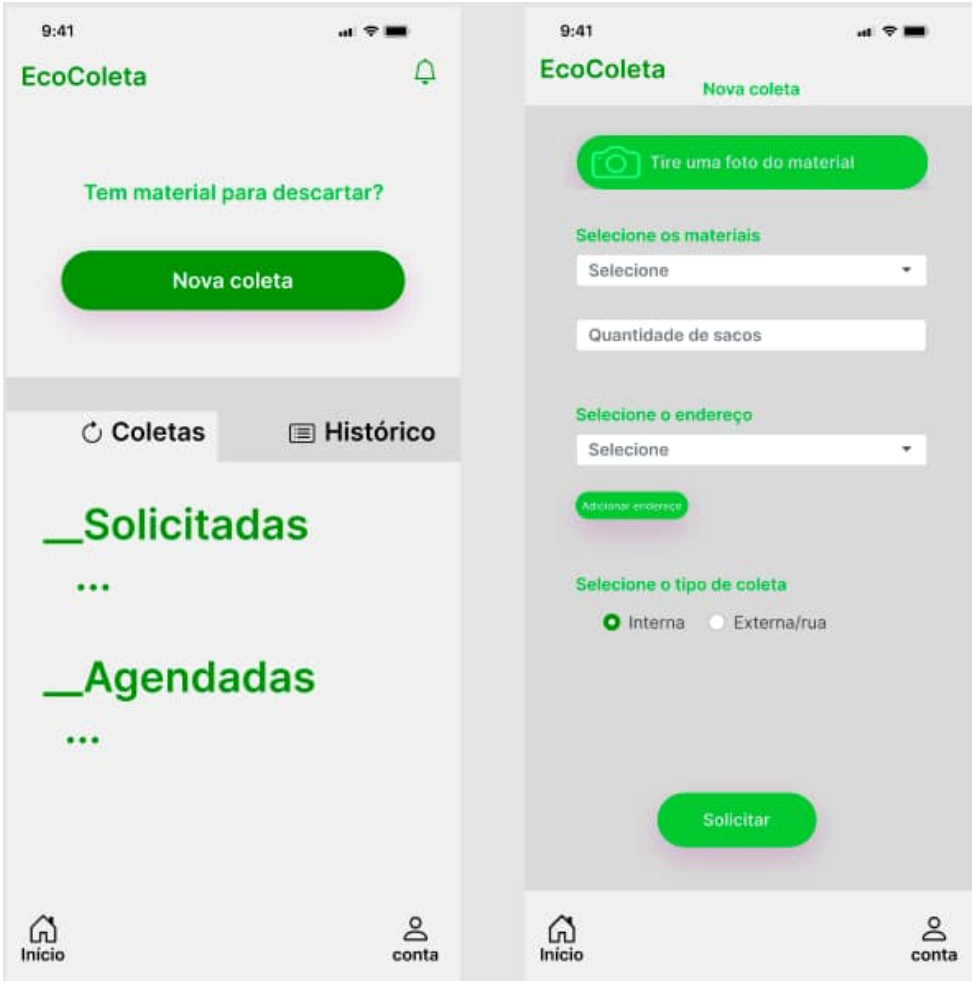
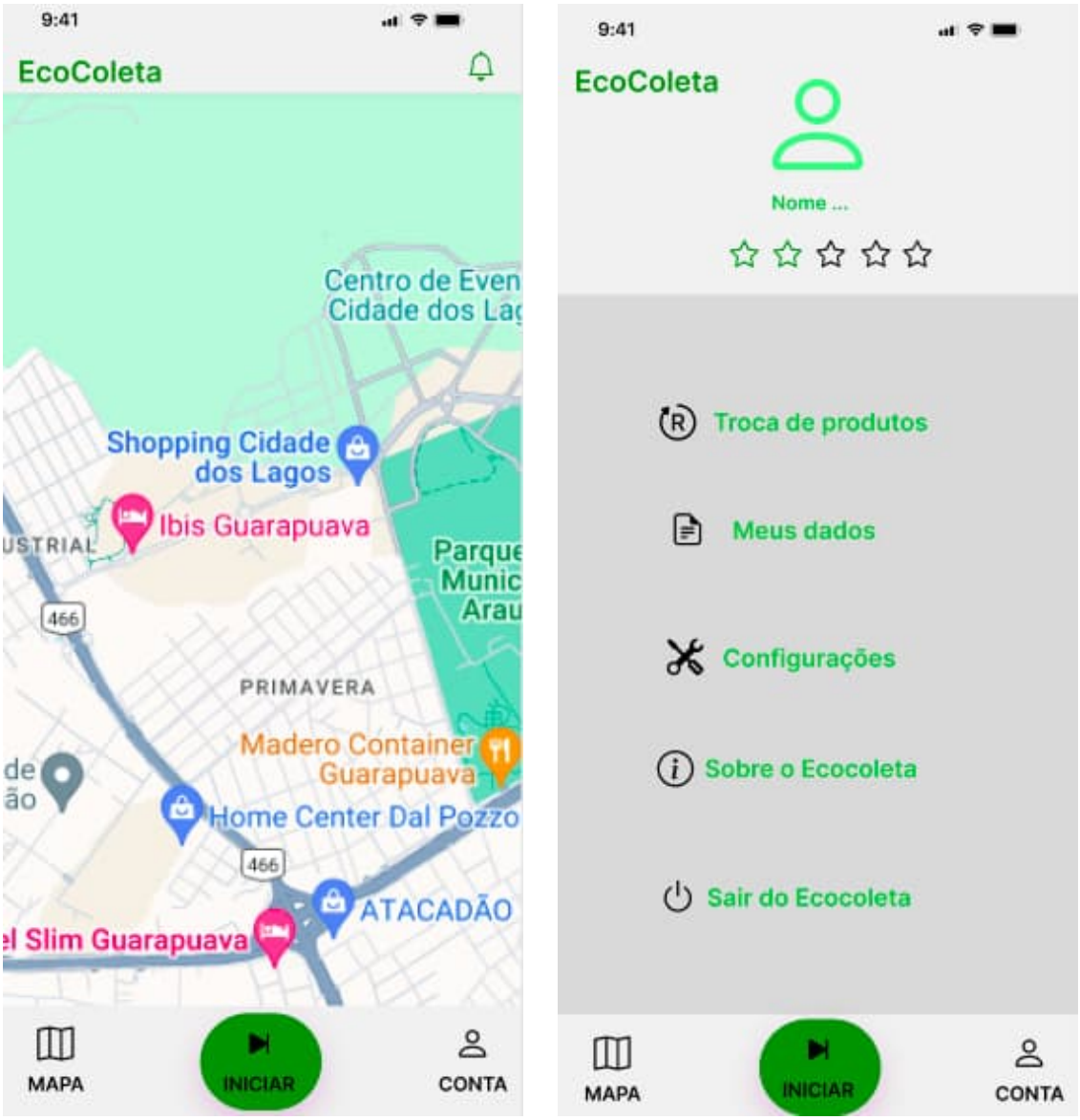


Figura 16 – Tela de solicitar coleta.
Fonte: Autoria própria (2025).

5.1.3.4 Tela do Perfil de Catador e Mapa Interativo

Para o papel de catador, a tela inicial é mostrada na Figura 17a. Nela, é apresentado o mapa interativo, integrado à *API* do *Google Maps*, que permite ao catador visualizar em tempo real os pontos de coleta próximos, facilitando a otimização das rotas. A tela de perfil do catador, representada na Figura 17b, exibe informações pessoais e a pontuação acumulada. O catador também pode acessar opções para edição de dados básicos e visualizar avaliações recebidas.



(a) Inicial **(b) Perfil**

Figura 17 – Telas inicial e perfil (catador).

Fonte: Autoria própria (2025).

5.1.3.5 Tela de Funcionalidade de Coletas e Rotas Otimizadas

A funcionalidade de coletas e rotas otimizadas é apresentada na Figura 18. Nessa tela, os pontos de coleta são destacados no mapa com base na proximidade.

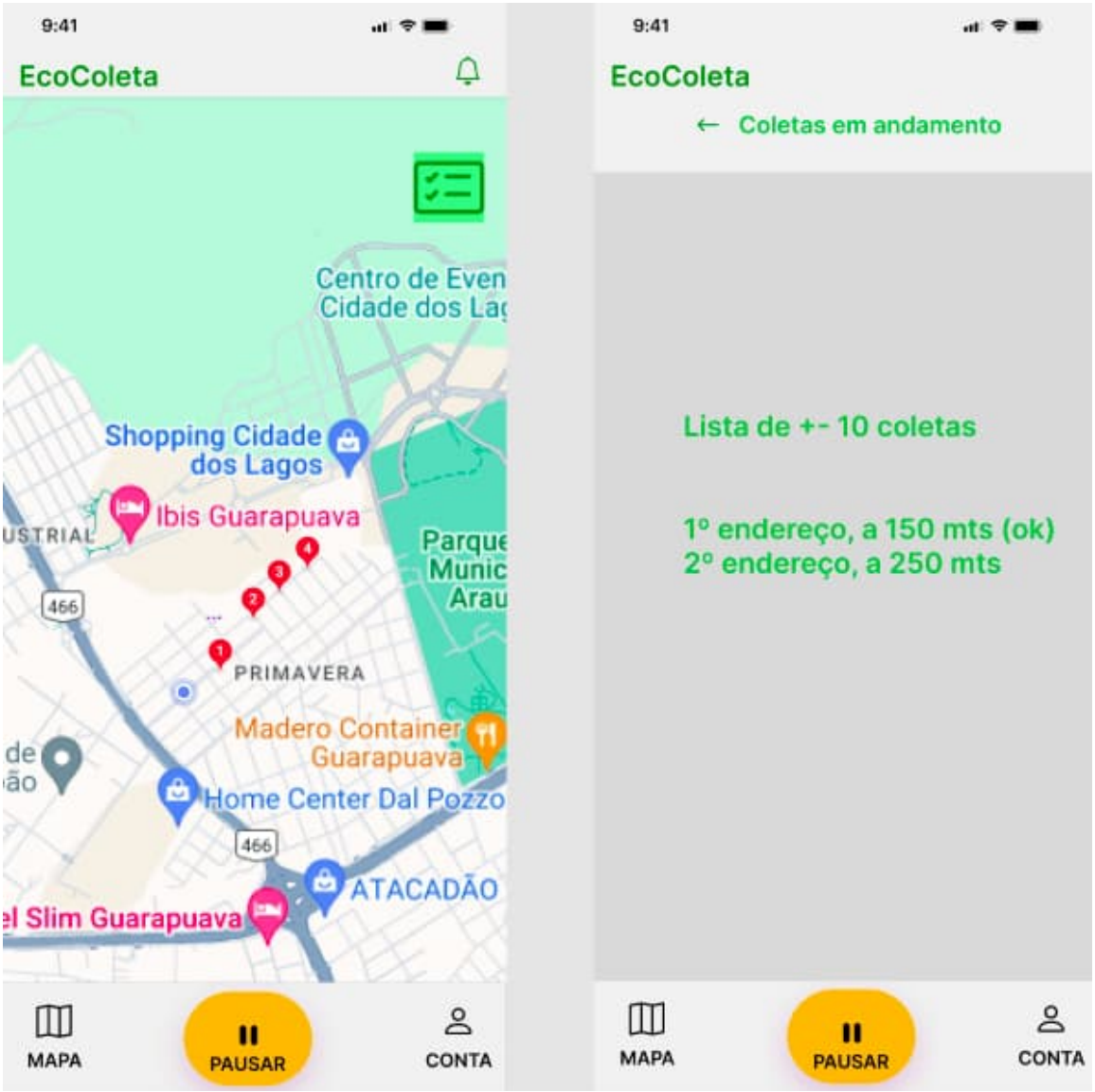


Figura 18 – Telas coletas em andamento.
Fonte: Autoria própria (2025).

6 DESENVOLVIMENTO DO SISTEMA

Neste capítulo, são detalhadas as etapas do desenvolvimento do sistema, se inspirando no framework Scrum. Cada fase será descrita minuciosamente, destacando as atividades realizadas, funcionalidades implementadas e dificuldades enfrentadas durante o projeto.

6.1 Sprint 1

Na primeira *sprint* no período inicial foi dedicado às configurações básicas do ambiente de desenvolvimento, bem como à leitura de documentações e tutoriais técnicos necessários para a implementação do sistema.

Além disso, foram realizadas as configurações iniciais do repositório de código, garantindo o versionamento adequado e o fluxo de trabalho colaborativo. Também foram criadas contas nas plataformas externas utilizadas no projeto, como a Render para o *deploy* em nuvem e a *Google Maps API* para funcionalidades de geolocalização. No banco de dados, foram configuradas a extensão *PostGIS* para suporte a operações geoespaciais e o *Flyway* para versionamento e migrações estruturais, preparando o ambiente para futuras funcionalidades complexas de localização e coletas otimizadas.

Para as funcionalidades de autenticação e segurança, foi implementada a história 26[1]. No *backend*, utilizando o *Spring Boot*, foi configurado o *Spring Security*, que gerencia a autenticação e autorização dos usuários, protegendo todas as rotas da API.

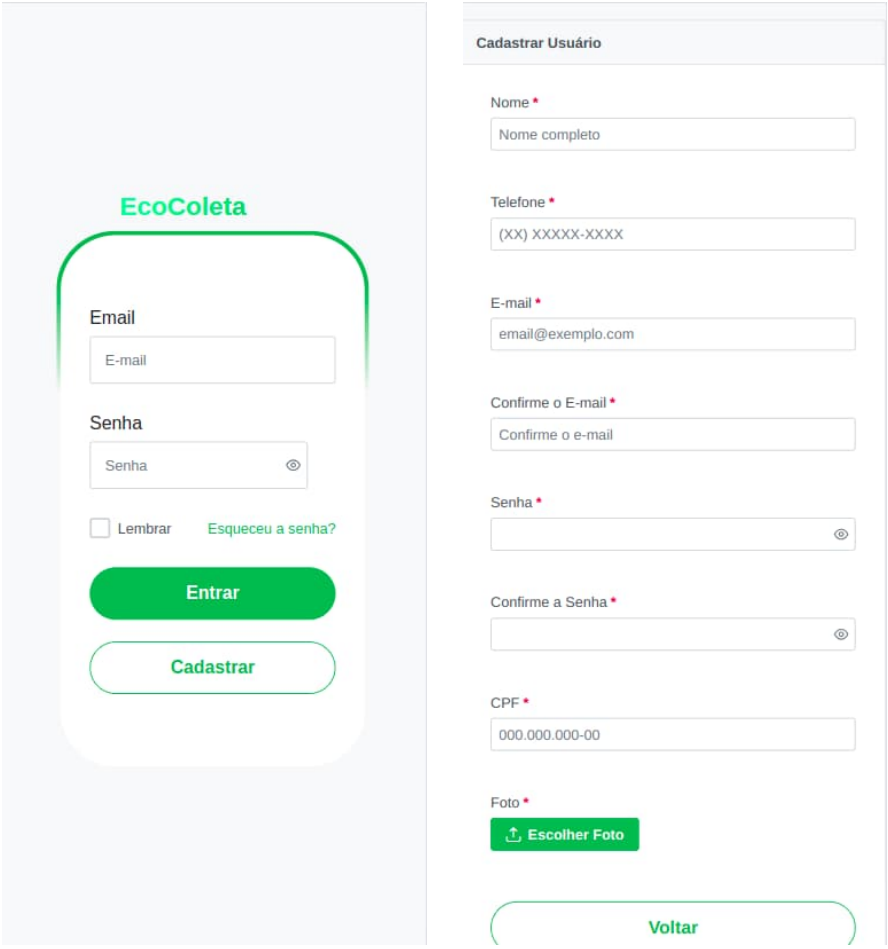
Para gerenciar a autenticação, foram implementados tokens JWT, que armazenam informações relevantes do usuário, incluindo seu papel no sistema, como exemplo o catador ou residente. Esses tokens são enviados em cada requisição ao *backend*, se o usuário dispor das permissões necessárias, o recurso solicitado é acessado; caso contrário, uma mensagem de erro é retornada.

No backend, testes unitários foram desenvolvidos para garantir que os dados fossem devidamente validados antes de serem persistidos no banco de dados. Um exemplo importante foi a verificação de armazenamento adequado de senhas criptografadas dos usuários.

No lado cliente, a validação ocorre de forma semelhante: ao acessar uma rota, o sistema verifica se o token do usuário é válido e se ele possui a permissão correta. Se as condições forem atendidas, a página correspondente ao papel do usuário é renderizada; caso contrário, o usuário é redirecionado para a página de login. A tela de login, que pode ser visualizada na Figura 19a, permite a autenticação segura do usuário para acessar as funcionalidades do sistema.

Com essas configurações iniciais e funcionalidades implementadas, a base do sistema foi preparada para o desenvolvimento das próximas etapas de forma robusta e segura.

[1] COMO Usuário QUERO tela de login e sessão PARA entrar na aplicação



(a) Login **(b) Cadastro de catador**

Figura 19 – Telas de login e cadastro de catador
Fonte: Autoria própria (2025).

6.2 Sprint 2

Foram desenvolvidas as histórias 32[2], 33[3], 35[4] e 37[5], referentes às páginas específicas de cadastro para cada papel de usuário no sistema e seus respectivos endereços. Inicialmente, o sistema permite o cadastro livre de usuários com o papel de separador e catador, enquanto os usuários com o papel de empresa só podem ser registrados pelo administrador. Para gerenciar essas informações, utilizou-se a tabela *users*, juntamente com tabelas derivadas, como *waste-collectors*, *residents* e *companies*, que armazenam os dados específicos de cada tipo de usuário.

Além disso, foi adicionada a coluna *active* à tabela *users*, permitindo o controle do status ativo ou inativo dos usuários no sistema, facilitando a administração de permissões e acessos.

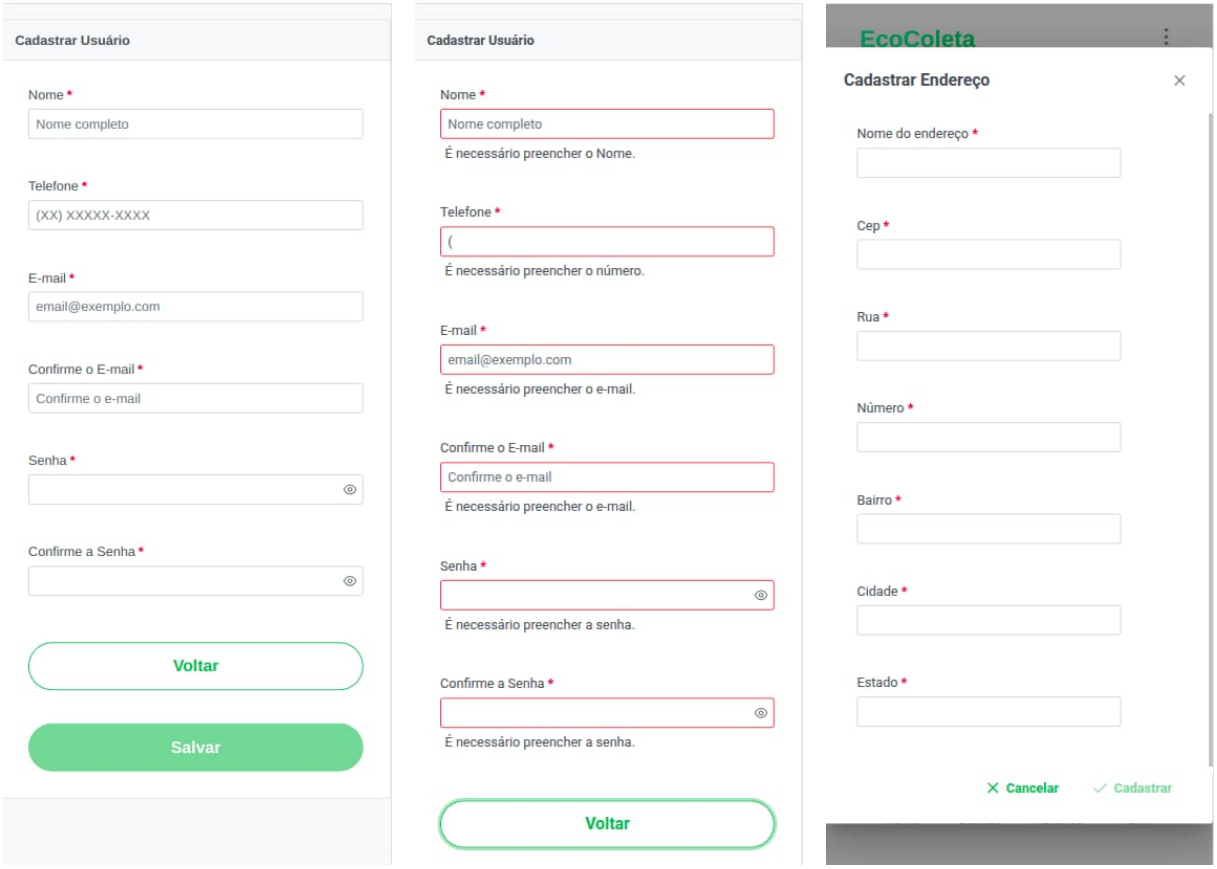
[2] COMO catador QUERO tela de cadastro PARA se inscrever na aplicação
[3] COMO usuário QUERO tela endereços PARA edição e cadastro de endereços
[4] COMO catador QUERO enviar uma imagem PARA ser visualizada em meu perfil
[5] COMO separador QUERO tela de cadastro PARA se inscrever na aplicação

Para o gerenciamento de endereços, foi criada a tabela *user-addresses*, que implementa uma relação muitos para muitos (N:N) entre os usuários e os endereços. Na tabela *addresses*, foram adicionadas as colunas *location*, longitude e latitude, com o objetivo de registrar a localização exata dos endereços diretamente no banco de dados. Essa estrutura foi planejada para permitir a fácil recuperação da localidade das coletas e a geração de rotas no mapa.

No *frontend*, foram desenvolvidos testes unitários para garantir que os dados retornados da API de cep fossem devidamente validados antes de serem persistidos no banco de dados.

Além disso, foram implementadas as interfaces de cadastro de usuários. As telas estão representadas, respectivamente, nas Figura 19b, Figura 20a. A tela de cadastro de endereços pode ser visualizada na Figura 20c.

Caso o usuário tente criar uma conta com informações incompletas ou inválidas, o sistema exibe uma mensagem de erro indicando o campo que precisa de correção. A tela que exibe essas mensagens de validação pode ser visualizada na Figura 20b.



(a) Cadastro de separador **(b) Cadastro de usuário com erro** **(c) Cadastro de endereços**

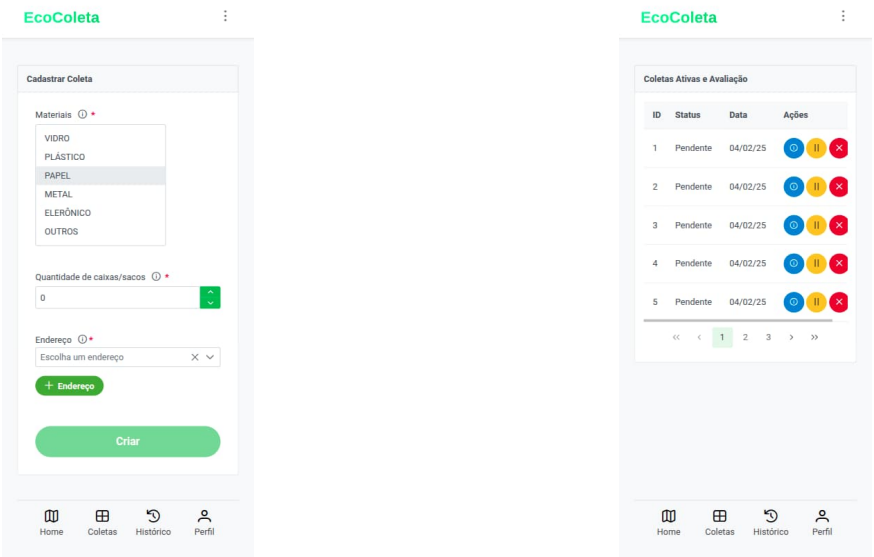
Figura 20 – Telas de cadastro: separador, usuário com erro e endereços.
Fonte: Autoria própria (2025).

6.3 Sprint 3

Nesta fase de desenvolvimento, foram implementadas funcionalidades essenciais relacionadas ao processo de solicitação e acompanhamento de coletas pelos usuários separadores.

Inicialmente, a funcionalidade prevista na história 14[6] contemplava a possibilidade de agendamento de coletas, permitindo que o separador especificasse uma data e hora para o atendimento. No entanto, durante o desenvolvimento e com um melhor entendimento das necessidades da aplicação, essa abordagem foi revisada. Optou-se por modificar o agendamento para uma funcionalidade de pausa da coleta já solicitada, visando melhorar a experiência de uso tanto para os separadores quanto para os catadores.

Essa mudança foi motivada por fatores importantes, como a simplicidade no uso do sistema e a segurança dos usuários. Com o recurso de pausa, o separador pode interromper temporariamente uma coleta sem a necessidade de cancelá-la, caso precise se ausentar. Isso evita a exposição desnecessária de informações de rotina e horários na aplicação. Uma coleta pausada não ficará disponível para entrar na rota de um catador até que o separador a reative. Para implementar essa funcionalidade, foi utilizado um *Enum* de status de coleta, permitindo a mudança dinâmica de estados, como *PENDING*, *PAUSED*, *IN-PROGRESS*, *COMPLETED* e *CANCELLED*.



(a) Solicitação de coleta **(b) Listagem de coletas atuais**

Figura 21 – Telas de solicitação de coleta e listagem das coletas atuais.
Fonte: Autoria própria (2025).

[6] COMO separador QUERO solicitar um agendamento PARA o catador efetuar a coleta no dia agendado

A história 17[7] foi implementada através da criação de uma tela de solicitação de coletas, conforme mostrado na Figura 21a. Nessa tela, o usuário separador pode preencher informações como o tipo de material a ser coletado (por exemplo, plástico, papel ou metal), a quantidade (em sacos ou caixas) e o endereço para a coleta. Inicialmente, havia a ideia de criar uma tabela separada para armazenar os materiais relacionados às coletas. No entanto, como o sistema envolve poucos tipos de materiais, optou-se por adicionar uma nova coluna *materials* diretamente na tabela *collects*, utilizando uma lista de *Enum* para representar os tipos de material. Isso simplificou a estrutura do banco de dados e facilitou a manipulação desses dados.

Além disso, para garantir o funcionamento eficaz das coletas, foram adicionados atributos às coletas na tabela *collects*, como *initTime* (tempo de início) e *endTime* (tempo de término). Esses atributos são essenciais para rastrear quando a coleta foi iniciada e concluída, melhorando a gestão e a rastreabilidade do processo.

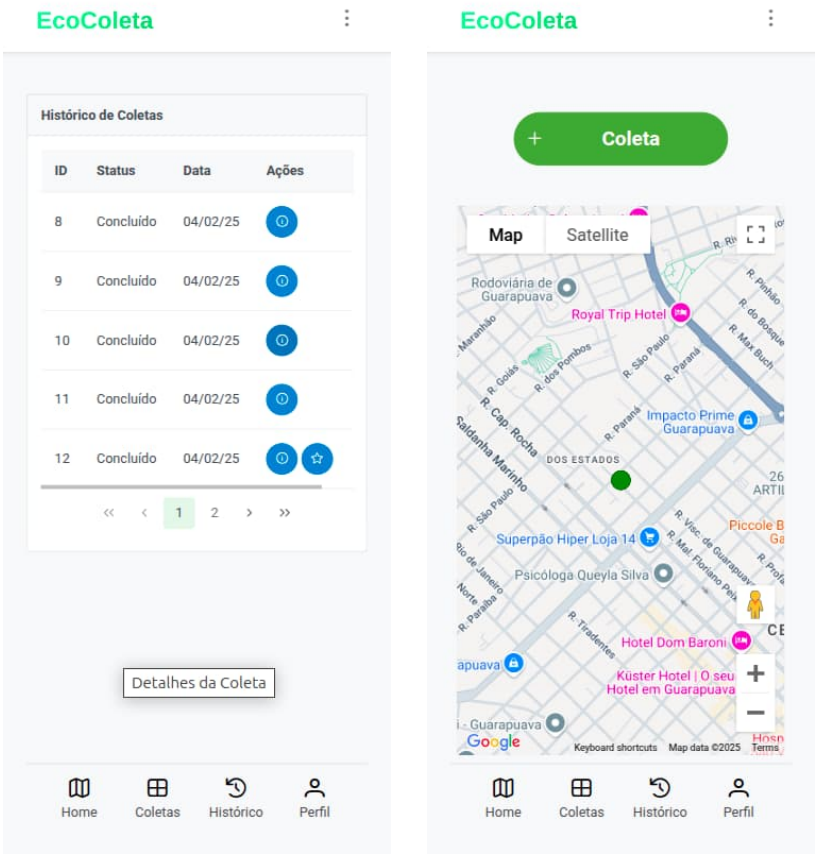
A história 43[8] resultou na criação de uma página de listagem de coletas solicitadas, organizada em duas telas principais. A primeira tela apresenta as coletas em andamento ou pausadas, conforme ilustrado na Figura 21b. A segunda tela é dedicada ao histórico de coletas, mostrada na Figura 22a, onde o usuário pode visualizar suas coletas finalizadas, canceladas ou ainda não avaliadas. A partir dessa tela, o separador pode acessar detalhes adicionais e avaliar as coletas realizadas.

Por fim, a história 42[9] levou ao desenvolvimento de uma tela principal com um mapa interativo, que exibe a localização dos catadores ativos em tempo real, conforme mostrado na Figura 22b. Para suportar essa funcionalidade, a tabela *waste-collectors* foi modificada com a adição das colunas *location* e *location-updated*. Quando o catador faz login na aplicação, um monitoramento automático de sua localização é iniciado, e os dados atualizados são armazenados no banco de dados. Esses dados são então utilizados para apresentar a posição atual dos catadores no mapa, proporcionando ao separador uma visão clara e atualizada.

[7] COMO separador QUERO solicitar uma coleta interna ou externa PARA o catador efetuá-la

[8] COMO separador QUERO página de listagem de coletas atuais e anteriores PARA manipular as coletas e avaliá-las

[9] COMO separador QUERO catadores ativos marcados no mapa



(a) Listagem de coletas **(b) Mapa com catadores ativos**

Figura 22 – Telas listagem das coletas e visualização de catadores ativos no mapa.
Fonte: Autoria própria (2025).

6.4 Sprint 4

Nesta etapa do desenvolvimento, foram implementadas funcionalidades voltadas à experiência dos catadores, oferecendo-lhes ferramentas para visualizar e gerenciar suas coletas de forma eficiente.

Inicialmente, na história 18[10], a ideia dessa funcionalidade era apresentar uma lista simples de coletas disponíveis, com informações básicas, permitindo ao catador selecionar manualmente qual coleta gostaria de atender. No entanto, durante o desenvolvimento, essa abordagem foi aprimorada para uma apresentação mais dinâmica e otimizada.

Com o melhor entendimento da aplicação, foi criada a história 41[11], que aprimorou a visualização das coletas no sistema. Agora, as coletas disponíveis são exibidas diretamente em um mapa, conforme ilustrado na Figura 23a. O mapa apresenta pontos de coleta em tempo real, possibilitando ao catador visualizar facilmente as opções mais próximas. Ao iniciar uma coleta, o sistema gera automaticamente uma rota otimizada para os três pontos de coleta mais próximos,

[10] COMO catador QUERO visualizar em texto e mapa a localização das coletas disponíveis PARA escolher a mais fácil no momento

[11] COMO catador QUERO coletas marcadas no mapa

conforme especificado na história 39[12], permitindo um fluxo contínuo e eficiente de trabalho. Conforme o catador conclui uma coleta, o sistema registra automaticamente a finalização e atualiza o status no banco de dados, possibilitando que o usuário separador avalie o serviço diretamente na interface.

Para garantir uma experiência fluida e sincronizada entre os diferentes componentes da aplicação (como o mapa, status das coletas, localização do catador e atualização de marcadores), foi desenvolvido um *service state* utilizando o padrão de gerenciamento de estado centralizado e o poder reativo do Extensões Reativas para JavaScript, do inglês *Reactive Extensions for JavaScript* (RxJS) no frontend Angular.

Esse serviço centraliza as informações dinâmicas, como:

- A lista de marcadores exibidos no mapa.
- A localização atual do catador.
- As coletas em andamento e seu status atual.
- A rota otimizada gerada e as atualizações em tempo real.

O RxJS foi utilizado para criar *Observables* que permitem que diferentes componentes da interface se inscrevam para receber atualizações automáticas e sincronizadas. Por exemplo, quando a localização do catador é atualizada, o mapa recebe automaticamente essa nova informação e atualiza o marcador correspondente sem a necessidade de recarregar a página. Além disso, botões de interação, como “iniciar coleta” e “parar”, são reativos, garantindo que as mudanças de estado sejam propagadas de forma instantânea para toda a aplicação, melhorando a experiência do usuário.

A funcionalidade de geração de rotas foi essencial para otimizar a rotina dos catadores, garantindo que as coletas sejam atendidas de forma eficiente.

Quando o catador clica em “iniciar uma coleta”, sua localização atual é enviada ao *backend*, onde o sistema utiliza o *PostGIS* para calcular os três pontos de coleta ativos mais próximos. Para identificar os pontos de coleta próximos, é utilizada a função geoespacial `ST_DWithin`, que encontra os pontos dentro de um determinado raio da localização atual do catador. A função calcula a menor distância utilizando o sistema de coordenadas geográficas armazenado no banco de dados.

No sistema, a coluna *location* na tabela *address* desempenha um papel fundamental nesse processo. Essa coluna armazena as coordenadas geográficas (latitude e longitude) de cada endereço no formato de ponto espacial (*geometry*). Para garantir a eficiência nas consultas, essa coluna é indexada utilizando um índice espacial do tipo GIST, que é otimizado para operações geoespaciais. Com essa indexação, o banco de dados pode executar buscas rápidas e precisas ao identificar os pontos de coleta mais próximos, mesmo quando o volume de dados é grande.

[12] COMO catador QUERO geração de rotas PARA minhas coletas selecionadas

O fluxo de cálculo das rotas funciona da seguinte forma:

- **Localização inicial do catador:** Ao clicar em “iniciar coleta”, a localização (latitude e longitude) do dispositivo do catador é enviada ao *backend*.
- **Consulta no banco de dados com PostGIS:** A função `ST_DWithin` verifica quais pontos de coleta armazenados na coluna *location* da tabela *address* estão dentro de um raio pré-definido (5 km) e seleciona os três pontos mais próximos.
- **Envio das coordenadas para o Google Maps:** As coordenadas geográficas dos três pontos de coleta selecionados são enviadas para a API do Google Maps, que gera uma rota otimizada.
- **Exibição da rota em tempo real:** A rota gerada é exibida no mapa na interface do catador, conforme ilustrado na Figura 23b. À medida que o catador conclui cada ponto de coleta, a rota é atualizada automaticamente.

Uma das principais funcionalidades do state service é garantir que, mesmo ao navegar entre diferentes páginas da aplicação, as informações cruciais da rota e das coletas em andamento sejam mantidas. O service state permanece ativo em segundo plano para garantir o monitoramento preciso do progresso do catador. O sistema acompanha constantemente a localização do dispositivo e, quando o catador se aproxima a menos de 30 metros de um ponto de coleta, o sistema registra automaticamente a conclusão daquela coleta e atualiza seu status no banco de dados. Essa automação evita a necessidade de o catador marcar manualmente a conclusão e melhora a eficiência geral do processo.

Caso o catador precise parar a coleta ou desistir das selecionadas, a interface permite parar a rota atual, oferecendo flexibilidade no planejamento diário.

Outro aspecto importante implementado é o tratamento de situações de perda de conexão ou saída inesperada da aplicação. Caso o catador perca a conexão com a internet ou feche a aplicação durante a execução de uma rota, o componente é capaz de recuperar as coletas em andamento ao retornar para a tela do mapa.

Ao iniciar o componente do mapa novamente, o service state verifica se há uma rota ativa ou coletas em andamento. Se houver, a rota anterior é restaurada, e o catador pode continuar de onde parou sem a necessidade de iniciar uma nova coleta. Esse processo inclui a recuperação das informações de localização, dos marcadores e do status atual de cada ponto de coleta.



(a) Mapa e pontos de coleta **(b) Rota gerada**

Figura 23 – Telas mapa com pontos e rota para coletas
Fonte: Autoria própria (2025).

6.5 Sprint 5

Foram desenvolvidas nesta sprint as histórias 16[13], 19[14], 47[15] e 20[16]. Após a conclusão de uma coleta, o sistema atualiza automaticamente o status no banco de dados, possibilitando que o usuário separador avalie o serviço diretamente na interface, conforme ilustrado na Figura 24a. Essa avaliação atribui pontos ao catador com base na qualidade das coletas finalizadas.

Inicialmente, foi implementada uma funcionalidade básica de pontuação, na qual o separador pode avaliar cada coleta concluída de 1 a 5 pontos (representados por estrelas). A coleta fica disponível para avaliação assim que for concluída e visível na interface do separador. A pontuação final do catador é gerada automaticamente por meio da média das avaliações obtidas em suas coletas, garantindo uma atualização dinâmica e confiável do desempenho geral. Essa

[13] COMO separador QUERO avaliar os serviços dos catadores
[14] COMO catador QUERO práticas de engajamento como gamificação, pontuações, etc.
[15] COMO catador QUERO pontuação ao finalizar coletas PARA ranking e futura troca de produtos
[16] COMO catador QUERO página de pontuação adquirida PARA troca de produtos

pontuação é persistida na tabela *collects*, no atributo *rating*. Após a inserção dessa avaliação, a média é calculada e o valor final é armazenado no atributo *score* da tabela *waste-collectors*, refletindo a evolução do desempenho do catador ao longo do tempo.

Na história 47, o sistema de gamificação foi projetado inicialmente com uma pontuação básica baseada na quantidade de coletas realizadas e nas avaliações fornecidas pelos separadores. Essa abordagem incentiva os catadores a manterem um desempenho consistente e de qualidade. Cada coleta concluída e avaliada contribui para o acúmulo de pontos, criando um mecanismo de *feedback* positivo.

No entanto, conforme previsto no escopo de implementação futura, a história 19[17] prevê uma expansão desse sistema. A pontuação será enriquecida com fatores adicionais, como a consideração dos tipos de materiais coletados e o cumprimento de metas diárias ou semanais. Esse aprimoramento proporcionará uma experiência de gamificação mais abrangente e motivadora.

Os pontos acumulados pelo catador poderão ser utilizados em programas de benefícios, como trocas por produtos ou prêmios, incentivando sua participação contínua e a melhoria constante do desempenho. A interface de acompanhamento da pontuação, ilustrada na Figura 24b, permite ao catador monitorar seu progresso e visualizar seu desempenho ao longo do tempo (história 20[18]).

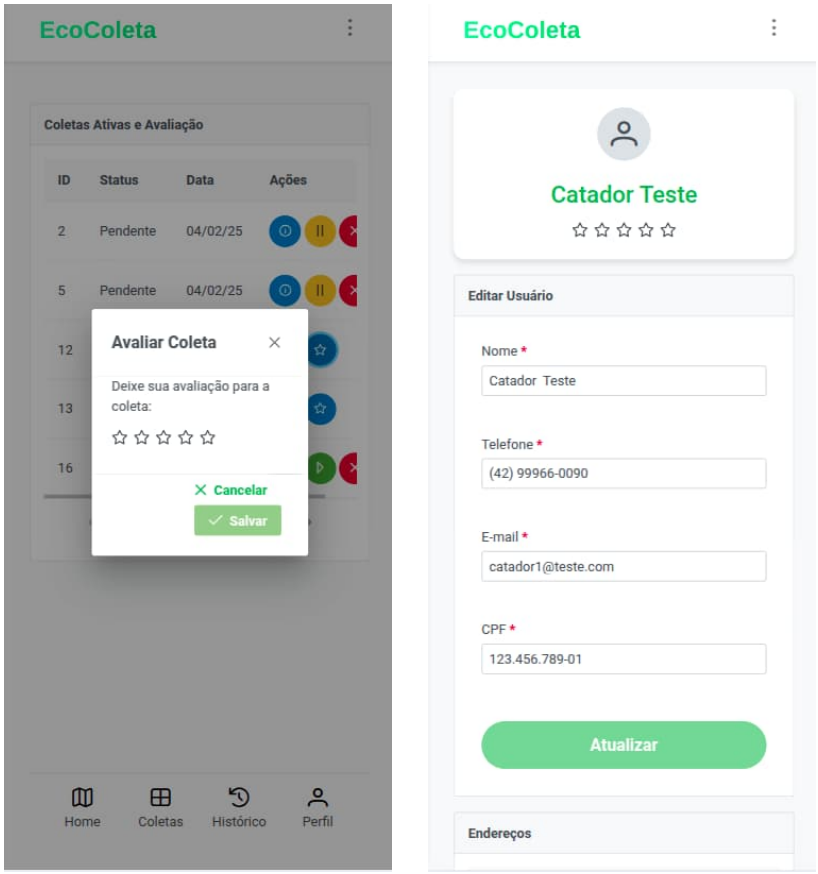
6.6 Sprint 6

Nesta *sprint*, foram desenvolvidas as histórias 23[19], 44[20] e 21[21], com o objetivo de fornecer funcionalidades administrativas completas para as Prefeituras gerenciarem os catadores e monitorarem o desempenho operacional de maneira eficaz.

Foi criado um painel inicial (*dashboard*) para o usuário com papel de empresa. O painel exibe métricas essenciais, como a quantidade de coletas finalizadas e canceladas no mês atual, o status das coletas do dia e a quantidade de usuários ativos. Essas informações ajudam as Prefeituras a monitorar e avaliar suas operações em tempo real e a tomar decisões mais informadas.

A página de relatórios (história 21), conforme ilustrado na Figura 25a, fornece uma visão detalhada sobre o desempenho dos catadores, incluindo a quantidade de coletas realizadas, canceladas e finalizadas, e outros dados operacionais relevantes. Essa funcionalidade permite análises mais aprofundadas e facilita a criação de estratégias de melhoria contínua.

[17] COMO catador QUERO práticas de engajamento como gamificação, pontuações, etc.
[18] COMO catador QUERO página de pontuação adquirida PARA troca de produtos
[19] COMO empresa QUERO página de cadastro e edição PARA cadastro de catadores, usuários e desativá-los quando necessário
[20] COMO empresa QUERO página de listagem de catadores
[21] COMO empresa QUERO página de relatórios PARA quantidade de catadores, coletas, etc



(a) Avaliação de coletas. **(b) Perfil do catador**

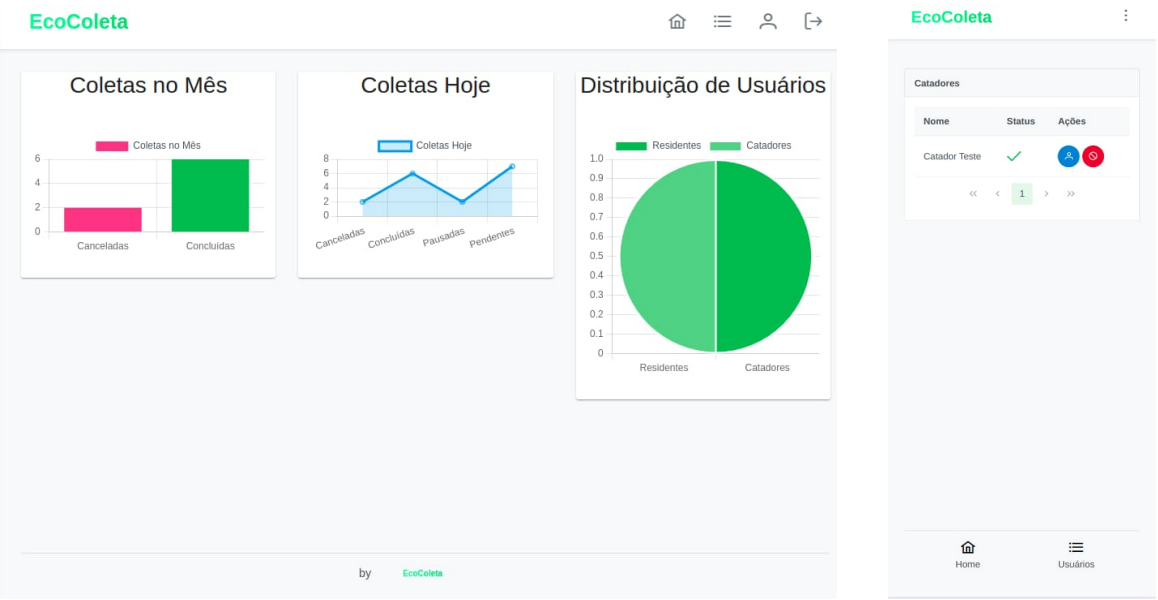
Figura 24 – Telas interface de avaliação e pontuação da coleta.
Fonte: Autoria própria (2025).

Também foi criada uma página que lista todos os catadores cadastrados no sistema (história 44), conforme ilustrado na Figura 25b. Nessa página de listagem, botões de ações estão disponíveis para permitir operações administrativas, como visualizar detalhes, editar informações ou desativar um catador. A empresa pode facilmente registrar novos catadores, editar dados existentes ou desativá-los, garantindo o controle efetivo sobre quem pode atuar no sistema, conforme especificado na história 23.

A listagem de catadores apresenta informações essenciais, como nome e status (ativo/inativo), fornecendo ao administrador uma visão geral da atuação de cada catador.

No *frontend*, as operações de cadastro e edição foram integradas com validações de formulários baseadas no Angular *Reactive Forms*, garantindo que os dados inseridos estejam corretos antes de serem enviados ao *backend*. Alertas e mensagens de *feedback* foram implementados para informar o usuário sobre o sucesso ou falha das operações, melhorando a experiência do usuário e a confiabilidade do sistema.

Para evitar inconsistências e duplicidade de dados, foi configurada uma validação adicional no *backend*, utilizando regras específicas para verificar se um catador com o mesmo e-mail já está cadastrado.



(a) Relatório empresa **(b) Listagem de catadores**

Figura 25 – Telas relatórios e listagem de catadores.
Fonte: Autoria própria (2025).

Essas funcionalidades foram desenvolvidas seguindo boas práticas de design de interface, validação de dados e segurança. Dessa forma, o sistema oferece uma experiência fluida e confiável para o administrador, garantindo que as Prefeituras possam gerenciar de forma eficaz os catadores vinculados às suas regiões e monitorar o desempenho das coletas por meio de relatórios detalhados.

6.7 Sprint 7

Nesta *sprint*, foram desenvolvidas as histórias 22[22], 38[23] e 40[24], com o objetivo de fornecer ao administrador do sistema funcionalidades robustas para o gerenciamento centralizado de prefeituras (*company*).

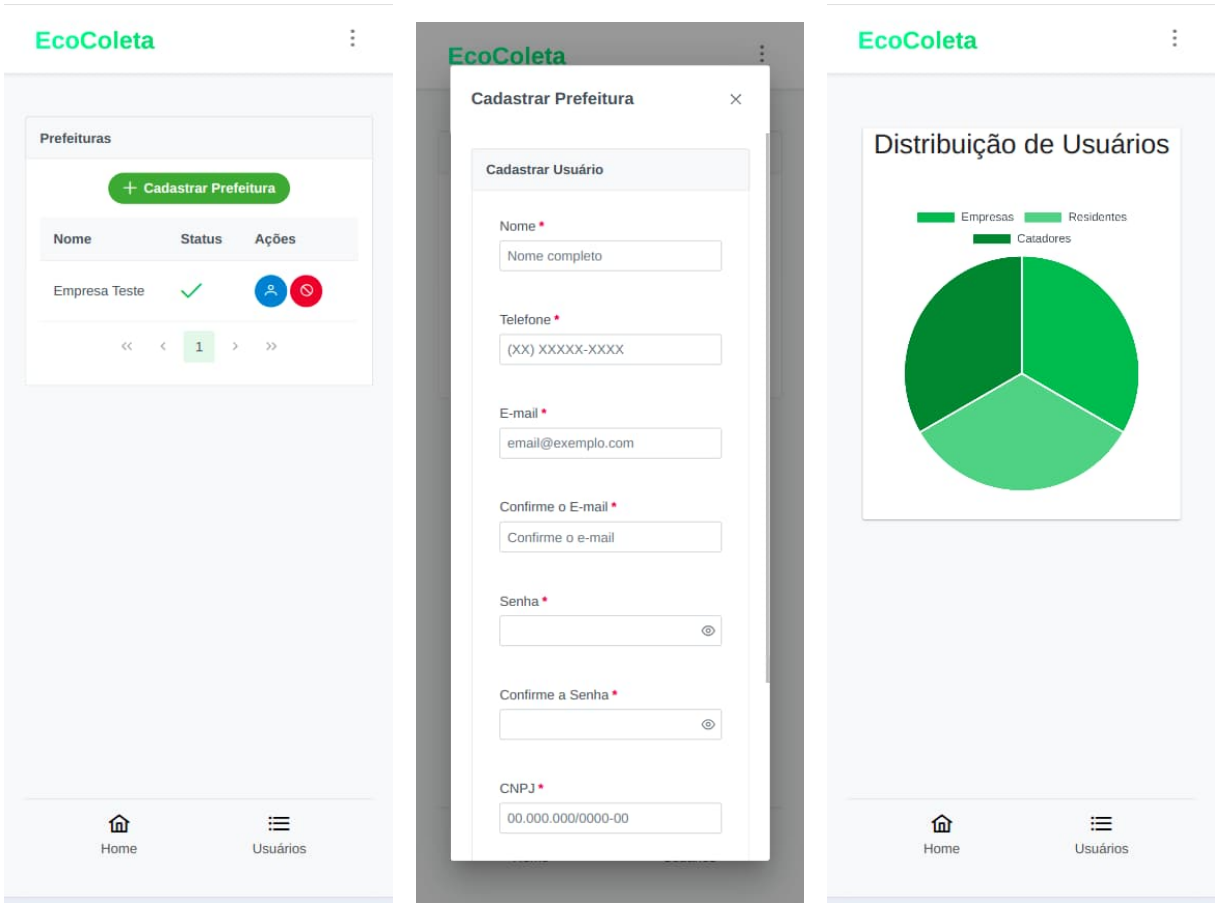
Seguindo a mesma padronização adotada para a tela de empresa na *sprint* anterior, foi desenvolvida uma página de listagem de empresas (história 40), conforme ilustrado na Figura 26a. Nesta página, o administrador pode visualizar informações detalhadas de cada empresa, incluindo nome, status. Além disso, a interface oferece botões de ações que permitem editar informações existentes, desativar empresas e adicionar novas empresas ao sistema. Essa abordagem garante um controle efetivo e centralizado das entidades empresariais.

Para facilitar a inclusão e gerenciamento de novas prefeituras e cidades, foi desenvolvida uma página de cadastro (história 22), conforme mostrado na Figura 26b. Essa funcionalidade

[22] COMO admin QUERO página de cadastros PARA cadastro de cidades, prefeituras, empresas, etc
[23] COMO admin QUERO gráficos de quantidades de usuários
[24] COMO admin QUERO página de listagem de empresas

permite ao administrador adicionar registros de prefeituras de forma rápida e eficiente. A validação dos dados foi implementada utilizando os formulários reativos do Angular, garantindo que informações inválidas ou inconsistentes sejam identificadas antes de serem enviadas ao *backend*.

Além disso, para proporcionar uma visão geral do sistema, foi implementado um *dashboard* dinâmico na página inicial do usuário com papel de administrador. Esse painel apresenta gráficos interativos que exibem informações importantes, como a quantidade de usuários cadastrados, conforme especificado na história 38. A Figura 26c ilustra essa interface, destacando a disposição clara e intuitiva dos dados.



(a) Listagem empresas **(b) Cadastro de Prefeitura** **(c) Dashboard admin**

Figura 26 – Telas de admin: listagem, edição e criação de prefeituras

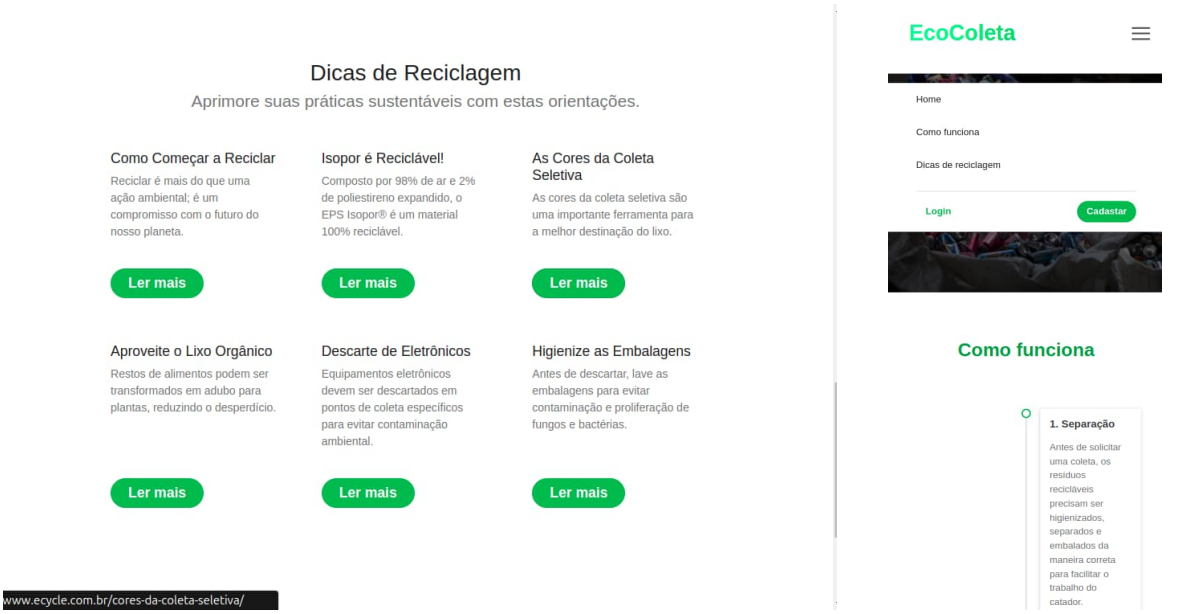
Fonte: Autoria própria (2025).

6.8 Sprint 8

Nesta *sprint*, foi desenvolvida a história 48[25], incorporando informações sobre práticas de reciclagem diretamente na página inicial da aplicação (*landing page*). Inicialmente, a ideia era implementar a história 15, que previa o envio de notificações com dicas de reciclagem para os usuários. No entanto, após análise do *Product Owner*, essa funcionalidade foi postergada para uma fase futura de desenvolvimento, e optou-se por uma abordagem mais simples e imediata, conforme especificado na história 48.

A *landing page* foi projetada para oferecer uma experiência atrativa e intuitiva, garantindo que as informações sobre práticas de reciclagem sejam facilmente acessíveis aos usuários. As dicas são apresentadas utilizando um design visual limpo e envolvente, que busca estimular o aprendizado e o engajamento dos usuários com a causa sustentável. As dicas incluem informações gerais sobre reciclagem, separação correta de materiais e orientações práticas para começar a utilizar o sistema, conforme ilustrado na Figura 27a.

Adicionalmente, a *landing page* apresenta dicas iniciais para orientar o usuário sobre como começar a usar a aplicação, promovendo uma introdução amigável ao sistema, conforme mostrado na Figura 27b. Essas orientações visam facilitar o primeiro contato do usuário com a plataforma e garantir uma experiência de navegação eficiente.



(a) Dicas de reciclagem **(b) Introdução da aplicação**

Figura 27 – Telas de dicas e introdução da aplicação.
Fonte: Autoria própria (2025).

[25] COMO usuário QUERO dicas de reciclagem

6.9 Sprint 9

Nesta *sprint*, o foco foi consolidar o desenvolvimento e otimizar o ambiente de produção, preparando a aplicação para o *deploy* final. Com um melhor entendimento da aplicação e das necessidades específicas do ambiente de produção, foram realizadas importantes configurações e melhorias no *backend*, *frontend* e banco de dados, garantindo maior desempenho, escalabilidade e estabilidade.

O projeto do *frontend*, desenvolvido em Angular, foi configurado para ser implantado como uma aplicação estática no *Render*, um serviço de nuvem gerenciado. Essa escolha permitiu uma configuração simplificada, alta disponibilidade e tempos de resposta rápidos. O *Render* oferece suporte eficiente para servir aplicações estáticas, garantindo que os usuários finais tenham uma experiência de navegação fluida.

Para atender às crescentes necessidades de desempenho e funcionalidade, o banco de dados anteriormente configurado como um *container* foi migrado para uma instância nativa do *PostgreSQL* no *Render*. Essa migração trouxe benefícios importantes, como:

- Melhor desempenho nas consultas e transações de dados.
- Suporte nativo à extensão *PostGIS*, essencial para cálculos geoespaciais.
- Funcionalidades de backup automatizado e escalabilidade nativa.

A extensão *PostGIS* foi devidamente configurada para garantir que as operações geoespaciais, como a geração de rotas e localização de coletas, fossem processadas com eficiência.

Dado que o *Render* não oferece suporte nativo para aplicações Java, o *backend* foi configurado utilizando *Docker*, o que permitiu maior flexibilidade na configuração das dependências. Foram ajustadas as *properties* específicas de produção.

6.10 Considerações Finais do Desenvolvimento

O desenvolvimento do projeto EcoColeta envolveu mais do que apenas programar funcionalidades. Foi um processo que exigiu planejamento, superação de desafios técnicos e adaptações. O objetivo era entregar um MVP funcional e alinhado com as necessidades reais da coleta seletiva.

6.10.1 Desafios e Adaptações

Desde o começo, um dos maiores desafios foi criar um sistema que otimizasse a comunicação entre catadores e separadores, mantendo a interface simples e fácil de usar. Isso exigiu ajustes nas *sprints*, reavaliações de funcionalidades e mudanças na modelagem do banco de

dados. Algumas tabelas inicialmente planejadas, como as relacionadas à troca de pontos por produtos (*products* e *exchanges*), foram removidas para simplificar o escopo do MVP, sem prejudicar o objetivo principal do projeto.

Além disso, aprendemos muito durante o desenvolvimento, principalmente ao trabalhar com tecnologias específicas, como o *PostGIS*, usado para geolocalização eficiente. No lado do *backend*, o *Spring Security* garantiu a segurança das rotas da API RESTful usando JWTs. Já no *frontend*, o Angular, configurado como PWA, permitiu o desenvolvimento de uma aplicação única, capaz de funcionar tanto em navegadores quanto em dispositivos móveis.

6.10.2 Aprendizado Técnico e Integrações

O desenvolvimento foi uma oportunidade prática de explorar várias tecnologias. O *PostGIS* trouxe soluções robustas para trabalhar com dados geoespaciais. Com ele, conseguimos encontrar pontos de coleta próximos (`ST_DWithin`) e calcular distâncias (`ST_Distance`) de forma rápida e eficiente.

A integração com a *API* do *Google Maps* complementou o *PostGIS*, permitindo a exibição de mapas interativos, a geocodificação de endereços e o cálculo de rotas otimizadas com o *Directions API*. Isso ajudou a reduzir o tempo de deslocamento dos catadores e aumentou a eficiência nas coletas.

6.10.3 Integração e Deploy no Ambiente de Produção

O processo de deploy foi um momento chave no projeto. Durante o desenvolvimento, usamos contêineres *Docker* para garantir um ambiente de testes controlado. No entanto, ao migrar para o ambiente de produção no *Render*, algumas mudanças foram feitas: o banco de dados *PostgreSQL* e o *frontend* Angular foram implantados como instâncias nativas para melhorar o desempenho e facilitar backups automáticos. Já o *backend* continuou utilizando contêineres *Docker*, o que proporcionou flexibilidade na configuração e fácil manutenção.

6.10.4 Mudanças no Banco de Dados e Refinamento do Modelo

Durante o desenvolvimento, a modelagem inicial do banco de dados sofreu ajustes para se adaptar às mudanças no escopo. Tabelas não prioritárias, como *products* e *exchanges*, foram removidas. Por outro lado, tabelas essenciais, como *collects* (coletas) e *addresses* (endereços), foram ajustadas para suportar a geolocalização e a avaliação das coletas.

Essas mudanças foram importantes para garantir que as informações geoespaciais fossem armazenadas e recuperadas de forma rápida e precisa. A tabela de endereços, por exem-

plo, passou a armazenar coordenadas geográficas no formato `POINT`, facilitando consultas eficientes com as funções do *PostGIS*.

6.10.5 Impacto do PWA no Projeto

Optar pelo desenvolvimento do *frontend* como uma PWA foi uma decisão acertada. Isso eliminou a necessidade de criar aplicativos nativos separados para Android e iOS. Com o mesmo código, conseguimos uma aplicação acessível em diferentes dispositivos, otimizando o tempo de desenvolvimento e facilitando futuras manutenções.

7 CONCLUSÃO

De fato, o descarte incorreto de resíduos tem impactos negativos tanto no meio ambiente quanto na sociedade. No caso específico do município de Guarapuava - PR, a quantidade de lixo produzida aumentou devido ao crescimento populacional e à falta de conscientização da população em relação à separação adequada dos resíduos. Isso resultou em um aumento nos custos de descarte e nos impactos ambientais na região.

A implementação de lixeiras coletivas na forma de contêiner foi uma solução adotada para simplificar e incentivar a coleta seletiva. No entanto, os desafios associados a essa abordagem evidenciaram a necessidade de uma solução tecnológica complementar, capaz de melhorar a comunicação entre os separadores e os catadores. Nesse contexto, surgiu a proposta de desenvolver o *EcoColeta*, uma aplicação web no formato de um MVP, com o objetivo de oferecer uma alternativa inovadora para tornar o processo de coleta seletiva mais eficiente.

Conforme mostrado no Capítulo 5, foi realizado o planejamento inicial da arquitetura do sistema, incluindo a modelagem do banco de dados, que sofreu alterações durante o processo de desenvolvimento. As tabelas inicialmente planejadas foram ajustadas para se adequar ao escopo reduzido do MVP, conforme descrito na subseção 5.1.2. No Capítulo 4, foi detalhado o uso de contêineres *Docker* durante o desenvolvimento e o posterior deploy no ambiente de produção no *Render*, onde o *backend* foi mantido como contêiner, enquanto o *frontend* e o banco de dados foram executados como instâncias nativas para melhor desempenho.

No Capítulo 6, o processo de desenvolvimento da aplicação envolveu uma abordagem prática, dividida em diversas *sprints*, cada uma abordando diferentes funcionalidades essenciais para o sistema. Inicialmente, foram implementados componentes como a autenticação segura via *Spring Security*, a comunicação geoespacial com o *PostGIS* e a integração com a *API* do *Google Maps* para geração de rotas otimizadas.

O EcoColeta foi desenvolvido como um MVP, cumprindo os objetivos gerais definidos no início do projeto, que incluíam promover a coleta seletiva, melhorar a comunicação entre separadores e catadores e oferecer uma solução escalável e eficiente. No entanto, como o sistema foi projetado em sua fase inicial, alguns aspectos de usabilidade e funcionalidades adicionais ainda não foram completamente implementados.

Entre os principais resultados alcançados, destacam-se:

- **Comunicação aprimorada**: O sistema permite que os catadores visualizem pontos de coleta em tempo real e otimizem suas rotas de trabalho, melhorando a eficiência operacional.
- **Autenticação segura**: O uso do *Spring Security* garantiu que apenas usuários autorizados possam acessar as funcionalidades sensíveis do sistema.

- **Integração geoespacial**: A combinação do *PostGIS* com a *API* do *Google Maps* proporcionou uma experiência precisa e confiável no gerenciamento de localizações e rotas.

- **Escalabilidade**: O PWA desenvolvido no *Angular* permitiu que a aplicação fosse acessível por diversos dispositivos sem a necessidade de desenvolvimento de aplicativos nativos.

Apesar das conquistas, o projeto ainda não foi apresentado aos usuários finais, como separadores e catadores, uma vez que o MVP atual representa uma versão inicial que precisa ser testada em ambientes reais pelo desenvolvedor e então, refinada. Porém, a aplicação foi desenvolvida considerando a possibilidade de futuras expansões.

Para concluir, devido a restrições de prazo, algumas funcionalidades não foram completamente implementadas. Assim, essas melhorias e outras funcionalidades adicionais serão descritas na seção 7.1.

7.1 Trabalhos Futuros

Com a conclusão do desenvolvimento do MVP, o sistema EcoColeta já apresenta uma base sólida, mas ainda há diversas oportunidades para evolução e aprimoramento. As melhorias planejadas envolvem a expansão de funcionalidades, ajustes de usabilidade e integração de novos recursos, conforme descrito na Tabela 4.

Entre as principais melhorias futuras, destaca-se a implementação da gamificação e do sistema de troca de pontos. Esse recurso permitirá que os catadores acumulem pontos com base no número de coletas realizadas e nas avaliações recebidas, possibilitando a troca por produtos ou benefícios. Para isso, serão desenvolvidos mecanismos de atribuição e gerenciamento de pontos, um catálogo de produtos disponíveis para troca e um histórico para acompanhamento do saldo. Esse sistema tem como objetivo aumentar o engajamento dos usuários e incentivar boas práticas de coleta seletiva.

Além disso, está prevista a criação de um sistema de notificações personalizadas para manter os usuários informados sobre coletas, mudanças de status e outras informações relevantes. As notificações serão enviadas diretamente pelo aplicativo, contribuindo para uma comunicação mais eficiente e uma melhor experiência do usuário.

Outras funcionalidades também foram identificadas como oportunidades de aprimoramento do sistema, conforme detalhado na Tabela 4. Com essas evoluções, o EcoColeta poderá oferecer uma experiência mais completa e engajante, atendendo melhor às necessidades da comunidade e fortalecendo a conscientização sobre a importância da coleta seletiva.

Tabela 4 – Histórias de Implementação futura

ID	Prioridade	Descrição da história de usuário
19	1	COMO catador QUERO práticas de engajamento como gamificações, pontuações, etc.
46	3	COMO separador QUERO receber notificações push ao finalizar uma coleta
15	3	COMO separador QUERO notificações PARA recebimento de dicas de reciclagem
25	4	COMO catador QUERO ranking PARA ter uma competição entre os demais catadores
24	4	COMO empresa QUERO cadastrar produtos e preços PARA disponibiliza-los para a troca
34	5	COMO Usuário QUERO telas de recuperação de senha e validação de e-mail e número
36	5	COMO empresa QUERO tela e formulário de cadastro de cidades PARA relacionar prefeituras e catadores
45	5	COMO empresa QUERO listagem de usuários da minha cidade

Fonte: Autoria própria (2025).

REFERÊNCIAS

ANDRADE, R. M. de; FERREIRA, J. A. **A Gestão de Resíduos Sólidos Urbanos no Brasil Frente às Questões da Globalização**. [*S.l.*], 2011. 16 p. Disponível em: http://www.revistarede.ufc.br/rede/article/view/118. Acesso em: 22 de Abril de 2023.

ANDREI. **O Que é Angular? Guia para Iniciantes**. [*S.l.*], 2023. 1 p. Disponível em: https://www.hostinger.com.br/tutoriais/o-que-e-angular. Acesso em: 28 de Abril de 2023.

ANTUNES, D. **O Que é Angular? Guia para Iniciantes**. [*S.l.*], 2023. 1 p. Disponível em: https://www.gobacklog.com/blog/progressive-web-apps/. Acesso em: 15 de Junho de 2023.

ARTIA. **Kanban, guia completo para entender o sistema de gestão visual**. [*S.l.*], 2023. 1 p. Disponível em: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-3.0.pdf. Acesso em: 28 de Abril de 2023.

ATLASSIAN. **Git Flow Workflow**. 2023. Acesso em: 16 de Junho de 2023. Disponível em: https://www.atlassian.com/br/git/tutorials/comparing-workflows/gitflow-workflow.

BOOT, S. **Spring Boot: Framework para Desenvolvimento de Aplicações Java**. 2023. Acesso em: 16 de Junho de 2023. Disponível em: https://spring.io/projects/spring-boot.

BRONOSKI, B.; MENDES, U. **Aterro sanitário de Guarapuava tem apenas mais quatro anos de vida útil**. [*S.l.*], 2017. 1 p. Disponível em: https://globoplay.globo.com/v/6094042/. Acesso em: 27 de Abril de 2023.

CARROÇA, P. M. **Sistema de Reciclagem**. [*S.l.*], 2023. 1 p. Disponível em: https://pimpmycarroca.com/. Acesso em: 27 de Abril de 2023.

DEVMEDIA. **Test Driven Development: TDD Simples e Prático**. [*S.l.*], 2023. 1 p. Disponível em: https://www.devmedia.com.br/test-driven-development-tdd-simples-e-pratico/18533. Acesso em: 15 de Junho de 2023.

EARTH911. **iRecycle App**. [*S.l.*], 2023. 1 p. Disponível em: https://earth911.com/irecycle/. Acesso em: 18 de Maio de 2023.

ECO, M. C. **Nossa missão é substituir o descartável e criar uma nova cultura de consumo acessível a todos**. [*S.l.*], 2023. 1 p. Disponível em: https://www.meucopoeco.com.br/site/sustentabilidade-meu-copo-eco. Acesso em: 27 de Abril de 2023.

GANDHI, M. **Minha vida e minhas experiências com a verdade**. 2. ed. São Paulo: Martin Claret, 2008.

GIT. **Git: Sistema de Controle de Versões**. 2023. Acesso em: 16 de Junho de 2023. Disponível em: https://git-scm.com/about.

GITHUB. **GitHub: Plataforma de Hospedagem de Código**. 2023. Acesso em: 16 de Junho de 2023. Disponível em: https://github.com/.

GITLAB. **GitLab: Plataforma de Integração Contínua e Desenvolvimento**. 2023. Acesso em: 16 de Junho de 2023. Disponível em: https://about.gitlab.com/.

Google Maps. **Google Maps Platform Documentation**. 2025. Acesso em: 4 fev. 2025. Disponível em: https://developers.google.com/maps/documentation.

GUARAPUAVA, S. D. M. A. P. de. **Secretaria De Meio Ambiente De Guarapuava Amplia Coleta Seletiva De Lixo**. [*S.l.*], 2023. 1 p. Disponível em: https://www.guarapuava.pr.gov.br/noticias/secretaria-de-meio-ambiente-de-guarapuava-amplia-coleta-seletiva-de-lixo/. Acesso em: 27 de Abril de 2023.

HOSTINGER. **O Que é Docker e Como Ele Funciona?** 2023. Acesso em: 16 de Junho de 2023. Disponível em: https://www.hostinger.com.br/tutoriais/o-que-e-docker.

INC., F. **Figma: The Collaborative Interface Design Tool**. 2025. https://www.figma.com. Acesso em: 09 fev. 2025.

JUNIOR, A. B. de C. *et al.* **Catadores de materiais recicláveis: análise das condições de trabalho e infraestrutura operacional no Sul, Sudeste e Nordeste do Brasil**. [*S.l.*], 2013. 10 p. Disponível em: https://www.scielo.br/j/csc/a/RDFvSTprvh8CBzXrsZNrPQN/?lang=pt&format=pdf. Acesso em: 22 de Abril de 2023.

MINDMASTER. **Scrum: A Metodologia Ágil Explicada de forma Definitiva**. [*S.l.*], 2023. 1 p. Disponível em: https://mindmaster.com.br/scrum/. Acesso em: 27 de Abril de 2023.

PISANO, V.; DEMAJOROVIC, J.; BESEN, G. R. **Política Nacional de Resíduos Sólidos do Brasil: perspectivas das redes de cooperativas de catadores**. [*S.l.*], 2022. 23 p. Disponível em: https://www.scielo.br/j/asoc/a/GrwgLFsQbf7dNSXB64GqVTg/?lang=pt&format=pdf. Acesso em: 22 de Abril de 2023.

POSTGRESQL. **O Que é PostgreSQL?** 2023. Acesso em: 16 de Junho de 2023. Disponível em: https://www.postgresql.org/about/.

RECICLABR. **Empresa De Transporte, Beneficiamento E Descarte De Resíduos**. 2023. Acesso em: 27 de Abril de 2023. Disponível em: https://reciclabr.com.br/.

RENDER. **Render - The Easiest Cloud for All Your Apps**. 2025. Acesso em: 4 fev. 2025. Disponível em: https://render.com.

SCHWABER, K.; SUTHERLAND, J. **O Guia do Scrum**. [*S.l.*], 2020. 16 p. Disponível em: https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-3.0.pdf. Acesso em: 27 de Abril de 2023.

SOFTWARE, R. **Flyway Documentation**. 2025. Acesso em: 04 fev. 2025. Disponível em: https://flywaydb.org/documentation/.

SOUZA, M. T. S. de; PAULA, M. B. de; PINTO, H. de S. **O Papel Das Cooperativas de Reciclagem Nos Canais Reversos Pós-Consumo**. [*S.l.*], 2011. 17 p. Disponível em: https://www.scielo.br/j/rae/a/ryBnGwKxMFymv3YrVwfFTdp/?format=pdf&lang=pt. Acesso em: 22 de Abril de 2023.

TEAM, T. P. D. **PostGIS Manual**. 2025. Acesso em: 04 fev. 2025. Disponível em: https://postgis.net/documentation/.

TRELLO. **Plataforma de Gerenciamento de Projetos**. 2023. Acesso em: 15 de Junho de 2023. Disponível em: https://trello.com/.