

COLETALI: UM SISTEMA PARA IDENTIFICAÇÃO DE LOCAIS DE DESCARTE OU DE DOAÇÃO DE MATERIAIS

Vanessa Ribeiro Koch^{1*}

Resumo

Este artigo trata de um sistema para auxiliar a população na identificação de locais para o descarte ou doação de materiais, refletindo no tratamento desses materiais de forma que possam ser reciclados ou reutilizados, contribuindo com a diminuição dos danos provocados pelo descarte incorreto. O site e o aplicativo foram desenvolvidos em JavaScript, a linguagem mais utilizada atualmente no desenvolvimento *web*, incluindo as bibliotecas NodeJs, React e React Native. Sua principal função é a busca por locais de coletas de acordo com a localização do dispositivo do usuário, assim como, também permite que os proprietários de locais de coleta cadastrem-se e gerenciem seus serviços.

Palavras-Chave: Aplicação Web. Aplicativo Móvel. Reciclagem. Doação.

1 INTRODUÇÃO

Atualmente, devido à ascensão e popularização da *internet*, a sociedade usa o acesso à rede para obter informações, se comunicar, utilizar redes sociais e aplicativos que supram suas necessidades na vida profissional ou pessoal.

O uso de dispositivos móveis atualmente já fazem parte do cotidiano do ser humano, com eles foram construídas novas relações entre a produção, recepção e a disseminação de informação, oferecendo a interação e a comunicação a qualquer momento ou lugar.

Assim como a tecnologia evoluiu, a necessidade por produtos que tenham grande durabilidade, a industrialização, o crescimento e a exigência no consumo de produtos aumentaram significativamente no volume de materiais desperdiçados. É comum vermos pessoas acumularem produtos, mesmo em bom estado,

^{1*} Acadêmica do curso de Análise e Desenvolvimento de Sistemas do Instituto Federal de Santa Catarina. vanessa.r15@aluno.ifsc.edu.br

simplesmente por serem substituídos por novos produtos (SOARES; MADUREIRA, 2018).

Alguns materiais, ao serem descartados, podem liberar substâncias tóxicas ou ficarem dispostos no meio ambiente por anos. Eles estão cada vez mais difíceis de serem tratados da forma correta devido à baixa degradabilidade. São ações que representam uma ameaça aos recursos naturais e a saúde dos seres vivos. Portanto, nesse cenário de preocupação, as autoridades precisam melhorar a sustentabilidade para gerenciar os produtos desperdiçados da maneira correta (SONG; ZENG, 2015).

Contudo, muitas soluções propostas ainda não trazem bons resultados. Se faz necessário uma mudança de padrão, deixando de identificar administradores públicos como gestores de resíduos, passando-os para gestores de recursos e recuperando materiais para produzir bens de consumo (REICHERT; MENDES, 2014). A população também deve considerar individualmente a administração de produtos de forma que possam ser reutilizados, encaminhando o material para locais de tratamento ou doando para indivíduos vulneráveis e instituições de caridade. É preciso que a população adquira consciência sobre a gravidade do descarte de materiais no ambiente e como este fator polui o planeta e a vida no cotidiano.

Nesse cenário de desperdícios constantes em que vivemos, enviar o material para onde ele possa ser reciclado ou reutilizado ou doar um produto em bom estado para famílias em situação de vulnerabilidade é a ação mais viável para tratar dos danos ao meio ambiente ou da saúde causados pelo depósito incorreto. Porém, mesmo com tanta informação, ainda é um grande desafio para a população encaminhar o material para um destino ideal.

Sendo assim, este trabalho trata de uma aplicação para auxiliar a população nas ações de destinação de materiais descartados e que traga autonomia ao indivíduo para contribuir com o ambiente e a sociedade. Esse modelo de tecnologia, que se associa ao conhecimento popular, remete a um desenvolvimento inovador de soluções para problemas que viabilizam uma inclusão digital ou a melhoria da qualidade de vida da população (MCTIC, 2017).

1.1 Objetivos

Nesta seção são apresentados o objetivo geral e os objetivos específicos do

presente trabalho de conclusão de curso.

1.1.1 Objetivo Geral

Promover o descarte e a doação de materiais através do desenvolvimento de um sistema com o qual será possível identificar destinos ideais para estes materiais.

1.1.2 Objetivos Específicos

- Investigar e conhecer pontos de coleta ou de doação de materiais;
- Desenvolver uma aplicação *web* que permita cadastrar pontos de coleta ou de doações de materiais;
- Desenvolver uma aplicação mobile, para a plataforma Android, que possibilite descobrir a destinação ideal para o descarte ou para a doação de materiais.

2 JUSTIFICATIVA

É grande a quantidade de pessoas e empresas que rotineiramente descartam materiais de maneira incorreta, poluindo o meio ambiente, aumentando gastos com limpeza pública e botando em risco a saúde da população, que possivelmente incorrerão em crimes passíveis de multas.

A vida moderna e a praticidade que ela nos traz, com o aumento desenfreado do consumo, provoca imprudência em relação ao descarte que, muitas vezes, não é por má índole, mas sim por ausência de conhecimento e de divulgação dos pontos de coleta. O que fazer e onde descartar materiais são grandes desafios presentes na sociedade atualmente.

O descarte impróprio de determinados produtos com certas propriedades físicas, químicas e biológicas pode pôr em risco a saúde da população, assim como a fauna e a flora, causando desequilíbrio ambiental. Estes materiais também exigem uma adequada manipulação, transporte, tratamento e disposição (MATOS; SANTOS, 2016).

Portanto, é evidente a necessidade de promover a educação ambiental, alavancar reflexões sociais, o pensamento sobre o coletivo e o apoio à sociedade através de ações socioambientais. Os cidadãos precisam ser educados em relação

à sustentabilidade (CRISOSTIMO, 2011).

O material descartado no ambiente ideal tem maiores hipóteses de ter uma destinação correta, favorecendo a reciclagem, trazendo resultados ambientais positivos, reduzindo o acúmulo de lixo e a toxicidade do ambiente. Além disso, o descarte correto também pode reduzir o gasto com energia e melhorar o aproveitamento dos recursos naturais, beneficiando, conseqüentemente, a qualidade de vida das pessoas, a saúde e até mesmo podendo gerar empregos (LIMA; JESUS, 2013).

Por outro lado, enquanto certos materiais precisam ser descartados da maneira correta para que possam ser reciclados e reaproveitados, outros podem ser utilizados por outras pessoas imediatamente, através de doações. Existem organizações, campanhas e institutos que promovem ações para auxiliar estas pessoas, mas a participação é normalmente baixa, mesmo com a desigualdade social crescendo a cada ano no Brasil.

Felizmente, o material descartado de uma pessoa pode ser a riqueza de outra pessoa. Por este motivo se faz necessário considerar uma boa administração de recursos e produtos, afinal, não temos recursos globais finitos (SONG; ZENG, 2015).

A inovação tecnológica conduziu novas ideias através de um jeito de pensar diferente e comprovando que a adaptação às tecnologias trazem a possibilidade de mudanças e novas soluções para as necessidades sociais (NETO et al., 2018).

Essa popularização do uso de aplicativos e sistemas tem transformado não somente a comunicação e o relacionamento entre as pessoas, mas também as relações econômicas, trabalhistas, sociais, de entretenimento e de ensino, mas ainda verificamos uma baixa oferta de aplicativos sociais sobre sustentabilidade (MORESI, 2017). Sendo que o desenvolvimento destes aplicativos pode qualificar o senso comum e também facilitar a comunicação entre doadores ou pessoas que precisam descartar materiais, com as instituições de coleta.

Considerando a abundância de materiais atualmente descartados de forma incorreta, a proposta deste trabalho consistiu no desenvolvimento de uma ferramenta para facilitar a destinação correta e contribuir com doações a quem necessita, reduzindo assim os danos causados ao meio ambiente e, também, a pobreza. As contribuições deste estudo poderão beneficiar todas as pessoas, sejam elas físicas ou jurídicas, mas especialmente aquelas que precisam descartar materiais rotineiramente e aquelas que necessitam de doações.

3 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentadas definições sobre o tema estudado e utilizado durante o desenvolvimento do sistema, baseado em autores de áreas específicas.

3.1 Descarte e Doação de Materiais

O ser humano, mesmo em escala individual, acumula peças e materiais e frequentemente os descarta de maneira indevida, contaminando nosso solo, mares e rios. Além disso, muitas empresas, mesmo atualmente, descartam materiais industriais diretamente na natureza, causando grande desequilíbrio ambiental.

Estes materiais, caso disponham de alguma característica tóxica, irão liberar compostos químicos, podendo danificar o meio ambiente, a saúde pública e a segurança da população. Essa situação também reflete a preocupação de toda a rede epidemiológica dos municípios, agindo através da contaminação da água, do ar, e até mesmo do solo, podendo atuar também através da proliferação de vetores de transmissão de doenças (FONSECA et al., 2014).

Em 2 de agosto de 2010, foi instituída a Lei n.º 12.305, denominada Política Nacional de Resíduos Sólidos (PNRS) (GARCIA et al., 2016). Através dela foi concedido a obrigatoriedade ao poder público no estabelecimento de programas para a reciclagem e tratamento de resíduos, coleta seletiva e redução do envio destes materiais para aterros sanitários, que em sua maioria, estão saturados.

“É uma importante regulação para o gerenciamento dos resíduos sólidos, embora, o desafio nesse sentido é adotar estratégias preventivas visando à redução ou eliminação da geração dos resíduos” (FONSECA et al., 2014).

Apesar de a legislação de proteção ambiental estar completando 10 anos, atualmente o ambiente ainda vem sofrendo ameaças com o excessivo consumo da população, industrialização e principalmente em ambientes sem serviço de tratamento de material e destinação. Ela não trata somente a responsabilidade do poder público e dos setores empresariais, inclui também a responsabilidade de toda a população (LIMA; JESUS, 2013).

Hoje, muito se fala sobre o reúso de materiais e recursos. Isso vem de encontro com a frequência em que materiais são desperdiçados erroneamente, podendo ser úteis de alguma forma. Na reciclagem ou no descarte efetivo, temos

como exemplo a possibilidade de gerar energia ou calor, ou na recuperação de materiais para formação de bens que possam ser consumidos novamente (REICHERT; MENDES, 2014). A este respeito, também encontramos materiais que estão em bom estado de conservação e sem uso, que podem ser doados às instituições filantrópicas ou a indivíduos vulneráveis.

A Fidelity Charitable é uma instituição de caridade pública independente, sem fins lucrativos, que já ajudou mais de 300 mil organizações. No seu relatório de 2020 é destacado que a maioria dos doadores utiliza a plataforma *web* e o aplicativo próprio para recomendar suas doações, permitindo com que os doadores as rastreiem e incentivem novos usuários. Em 2019, quase 35 mil recomendações de doações foram iniciadas pelo aplicativo, criaram-se quase 15 mil novas contas de doações e totalizaram US \$108 milhões de dólares de doações - um aumento de mais de 50% comparado ao ano anterior. Além disso, perceberam o crescimento de doadores mais jovens e preocupados com a sociedade, principalmente pelo aumento do interesse aos investimentos e fundos de caridade investidos. Muitos aproveitaram o crescimento do mercado de ações para contribuir com ativos não monetários à instituição, o que trouxe vantagens fiscais sobre os ganhos de capital, reduzindo impostos para o doador. O que demonstra a grande vantagem que o aplicativo e a página na *web* trouxe a esta instituição (FIDELITY CHARITABLE, 2020).

Saber a melhor forma de descartar um material ainda é um desafio e, mesmo com tal conhecimento, a população necessita de incentivos para agir da forma correta.

3.2 Tecnologias e Ferramentas

Nesta seção serão apresentadas algumas definições das principais tecnologias utilizadas durante o desenvolvimento do projeto.

3.2.1 JavaScript

O JavaScript é uma linguagem de programação de alto nível que possui tipagem dinâmica e suporta estilos de programação orientada a objetos, imperativos e funcionais. Foi a única linguagem de programação utilizada em todo o projeto,

desde o desenvolvimento ao lado do servidor, do site e até mesmo do aplicativo mobile. O fato de utilizar apenas uma linguagem de programação em diversas plataformas diferentes trouxe certa otimização do tempo e evolução do conhecimento.

Ela foi criada pela Netscape na fase inicial da web e hoje é uma das linguagens mais fortes no desenvolvimento web, em razão de possibilitar criar páginas interativas com HTML e CSS (FLANAGAN, 2011; SHUTE, 2019).

No início de sua história, logo após ser introduzido pela Netscape com o Netscape Navigator 2.0 em setembro de 1995, a Microsoft lançou sua própria versão chamada de JScript com o Internet Explorer 3.0. Devido a grande semelhança entre às duas linguagens, a associação especializada em padrões e normativas ECMA - *European Computer Manufacturers Association*, definiu a linguagem ECMAScript para criar um padrão e normatizar a linguagem (HENNEN et al., 2000). Hoje o JScript e o JavaScript se baseiam no ECMAScript, mesmo que cada uma delas contenham recursos adicionais não especificados no padrão.

Seus interpretadores estão presentes nos navegadores mais modernos dos dispositivos móveis, computadores ou até console de jogos (FLANAGAN, 2011). Por estar presente nos navegadores mais comuns, o JavaScript não requer nenhuma instalação separada de um *plugin* ao lado do cliente para ser executado (HENNEN et al., 2000). Além disso, a linguagem nos oferece uma variedade de *frameworks* e bibliotecas para diversos dispositivos.

Analisando uma pesquisa realizada anualmente pela comunidade Stack Overflow, em especial a que foi realizada em fevereiro de 2020 com cerca de 65 mil desenvolvedores, vemos que o JavaScript lidera a classificação de tecnologias que vem sendo utilizadas nos últimos anos (STACK OVERFLOW, 2020).

Assim como eles, o GitHub também demonstra que o JavaScript tem o maior número de repositórios disponíveis, superando todas as outras linguagens (ANTANI et al., 2016). Ele tem sido uma das linguagens de programação mais populares no mundo, a qual se fortaleceu através do surgimento do Node.js (descrito na próxima subseção), um interpretador multiplataforma de ambiente de execução JavaScript fora do navegador (MOZILLA, 2020).

3.2.2 Node.js

Hoje o JavaScript não é apenas uma linguagem *frontend* graças ao ambiente de execução Node.js. Ele permite que os desenvolvedores que trabalham no lado do cliente também trabalhem funções ao lado da rede e do servidor, usando a mesma linguagem de programação (MOZILLA, 2020). Esse foi o principal motivo da escolha dessa ferramenta para o projeto. Ele serviu todo o *backend* da aplicação e gerenciou todos os pacotes utilizados. Com ele, foi implementado toda a regra de negócio do sistema, a implementação de rotinas de segurança, conexão com os bancos de dados, conexão com o Amazon S3 para armazenamento de imagens em produção, configuração de guardas e validações de rotas.

O Node.js tem um alto desempenho e pode trabalhar com uma grande carga de solicitações por usar o mecanismo JavaScript V8 do Google, o mesmo do navegador Google Chrome (ZAMMETTI, 2020). Ele foi lançado para criar aplicativos de rede escaláveis utilizando um modelo orientado a eventos assíncronos, não bloqueante, que o torna leve e eficiente, perfeito para aplicações com intenso fluxo de dados que rodam em múltiplos dispositivos (NODE.JS, 2020).

Com o Node.js temos a capacidade de execução paralela e redução na exigência dos recursos computacionais. Esses benefícios são proporcionados porque a ferramenta possui um ambiente de *thread* único para tratar suas requisições. Essa *thread*, chamada de *Event Loop*, trata cada requisição como um novo evento. Ou seja, ela fica aguardando novos eventos para tratar, como um *loop*, dispensando a necessidade de criar uma nova *thread* para cada requisição, como nas demais linguagens de programação (ZAMMETTI, 2020).

Diferente do modelo de concorrência comum em outras tecnologias baseadas em *threads*, nas operações de entrada e saída (E/S) com o Node.js, os processos não são bloqueados, pois a *thread* não espera que essas operações sejam concluídas para continuar sua execução (LUNDAR;GRONLI;GHINEA, 2013).

Além disso, existe um repositório amplamente utilizado no gerenciamento de pacotes e versões do Node.js com diversas dependências, chamado de *Node Package Manager*, ou simplesmente NPM.

Com exceção da interface DOM - *Document Object Model*, o Javascript possui uma biblioteca pequena, fazendo com que ele dependa de outros pacotes como o NPM do Node.js (BLANTHORN et al., 2019). Criado em 2010, um gerenciador de pacotes como o NPM, que fornece acesso a mais de 750.000 pacotes, automatiza e facilita o *download* e a instalação de pacotes individuais

(COX, 2019).

3.2.3 React

O React é uma biblioteca JavaScript de código aberto, cuja função principal é tornar a experiência do usuário mais eficiente criando *interfaces* de usuário.

A ferramenta foi utilizada para o desenvolvimento *web* do projeto e foi escolhida pela facilidade no desenvolvimento de componentes modernos, pela combinação entre eles, atualizações frequentes e pela facilidade de encontrar materiais para consulta.

Ela é baseada em componentes escritos em JavaScript e faz a atualização e a renderização apenas dos componentes necessários em tela (FACEBOOK, 2020). Esses componentes poderão, ainda, ser reutilizados em outros projetos ou até mesmo dentro de outros componentes.

Durante a criação de *interfaces* com o React, podemos implementar diversos algoritmos e tecnologias para desempenho e reutilização da renderização de componentes de tela. Diferente de outras linguagens, ele trabalha com o DOM através do seu próprio Virtual DOM, um dos seus grandes benefícios junto com a atualização do estado de componentes. Ou seja, quando precisamos alterar um estado de um componente, a alteração é declarada no VDOM. E para obter um maior desempenho de renderização, o VDOM é mantido em memória sincronizado com a imagem do DOM real, atualizando apenas os componentes que mudaram de estado (PRUSTY, 2016).

A biblioteca também possui uma sintaxe própria chamada JSX, semelhante ao XML, que combina JavaScript e HTML. Ela surgiu como uma alternativa para escrevermos nosso *template* servindo para facilitar o uso da API do componente (HORTON; VICE, 2016).

Sobre os benefícios desse padrão o autor Boduch (2017) afirma que:

A sintaxe no estilo XML facilita a descrição da aparência da interface do usuário. Ou seja, quais são os elementos HTML que este componente renderizará? Isso é chamado de programação declarativa e é muito adequado para o desenvolvimento de interfaces.

Foi criado pelo próprio React e atualmente, devido a sua popularidade, diversas bibliotecas estão utilizando.

3.2.4 React Native

Lançado pelo Facebook em 2015, logo após o sucesso do React para *web*, o React Native é uma biblioteca para a criação de aplicativos móveis nativos, que usa de tecnologias conhecidas, construídas sobre o React (MASIELLO; FRIEDMANN, 2017).

Utilizado para o desenvolvimento do aplicativo móvel do projeto, a biblioteca criou um repositório que foi executado no próprio celular do desenvolvedor e testado em emuladores do Android Studio de diversos tamanhos. Para a execução, foi utilizado uma ferramenta chamada Expo CLI que acessa a aplicação através do aplicativo Expo Go. O expo, segundo a documentação (EXPO, 2021), executa a aplicação e traz fácil acesso às API's nativas dos dispositivos Android e iOS. É um *framework* que permite criar o projeto, executar o servidor, visualizar os *logs*, trazendo inclusive mais facilidade durante o *deploy* da App Store ou da Play Store.

Com o React Native conseguimos criar aplicativos nativos para Android e iOS escrevendo a lógica de negócios em JavaScript e compondo as visualizações com a utilização do JSX (MASIELLO; FRIEDMANN, 2017; FACEBOOK, 2020). Dessa forma, é necessário compreender o React antes mesmo de trabalhar com o React Native e é possível dispensar a necessidade de utilizar outra linguagem de programação para criar aplicativos móveis.

Ao invés de manipular a DOM com Virtual DOM como faz o React, o React Native faz uma comunicação através de uma *bridge* com chamadas assíncronas para a plataforma móvel subjacente, chamando as APIs de ferramentas nativas (BODUCH, 2017). Disponibilizando assim, um conjunto de componentes nativos como o *View*, *Text*, *Image*, o acesso à câmera do dispositivo, GPS, e entre outros, sem depender da utilização de WebViews (MASIELLO; FRIEDMANN, 2017; FACEBOOK, 2020).

Da mesma forma que o React Native permite a utilização de componentes nativos essenciais, ele também possibilita a criação de componentes nativos próprios conforme a necessidade.

3.2.5 Single Page Application - SPA

O Single Page Application é uma aplicação *web* que carrega somente o

código necessário para a utilização quando a página é inicializada. Trazendo uma experiência similar aos aplicativos *desktop* e *mobile* para o cliente. Esse modelo foi utilizado nas páginas *web* com o objetivo de diminuir o custo das renderizações das páginas.

É uma página única que funciona ao lado do cliente e não é recarregada durante o acesso, apenas reescreve o conteúdo dinamicamente na mesma página.

Isso, por sua vez, aumenta a interatividade da página, a capacidade de resposta e a satisfação do usuário. Ela solicita a página e os dados de forma independente, renderizando a página no navegador e enviando em maior parte, os dados para o servidor (SOLOVEI; OLSHEVSKA; BORTSOVA, 2018). Dessa forma, na arquitetura SPA, boa parte da lógica de negócio e geração do HTML é transferida para o cliente, fazendo com que reduza a carga de recursos no servidor e fluxo de rede (PUSKARIC, 2019).

Hoje, milhares de pessoas utilizam o SPA diariamente em aplicações como, por exemplo, o Facebook, Google Maps e o Gmail (SOLOVEI; OLSHEVSKA; BORTSOVA, 2018). É uma arquitetura relativamente nova, e já possui complementos e bibliotecas que auxiliam nos seus aspectos específicos, como o Angular e o React (PUSKARIC, 2019).

3.2.6 Banco de Dados - SQLite/MySQL

Conforme a documentação do Sqlite (2020) “O SQLite é uma biblioteca em linguagem C que implementa um mecanismo de banco de dados SQL pequeno, rápido, independente, de alta confiabilidade e completo”.

Ele foi a primeira escolha do projeto pela fácil utilização e leveza comparada a outras ferramentas de banco de dados. Mas devido à realização do *deploy* na nuvem da plataforma Heroku e o SQLite executar somente em memória e armazenar em disco, o Heroku não recomenda e restaura o banco em até 24 horas. Dessa forma, nesse projeto ele está sendo utilizado como SGBD apenas na execução local e na produção foi substituído para o MySQL, utilizando a conexão na ferramenta MySQL Workbench. Como foi utilizado a biblioteca Knex, que facilita a conexão de banco de dados e a migrar para outros bancos de dados relacionais sem muitas alterações, esse processo de migração acabou sendo rápido.

O SQLite, portanto, é um banco de dados *open source*, relacional, que salva

dados em um único arquivo localmente, composto por tabela, índice, gatilho e exibição de entradas. Por ser apenas um arquivo único, ele é fácil de ser copiado, editado e facilmente transportável. Ele suporta a maioria dos padrões ANSI/ISO SQL e garante um mecanismo de armazenamento seguro com transações ÁCID - Atômico, Consistente, Isolado, Durável (SANGJUN, 2012). Sua instrução em SQL serve para recuperar, criar, atualizar, editar e remover dados no banco de dados (ADITYA; MOHANTA; KARN, 2014).

Portátil, fácil de usar, eficiente e confiável. Dispensa a necessidade de configurar parâmetros de rede ou *firewalls*, pois tanto o cliente quanto o servidor trabalham em um único processo, eliminando problemas de configurações (BOROVSKOY et al., 2017). Ou seja, ele não requer nenhum arquivo de configuração, etapas de instalação ou configuração inicial, tem um código fonte livre para modificar e distribuir e é um único arquivo comum, sem servidor, leve e simples (ADITYA; MOHANTA; KARN, 2014). E atualmente ele já está embutido em *smartphones*, na maior parte dos computadores de mesa e é fornecido por diversas outras aplicações (SQLITE, 2020).

Já o MySQL é um projeto de código aberto que pertence à Oracle. Ele necessita de um servidor e uma quantidade maior de configurações, em contrapartida, disponibiliza muito mais recursos comparado ao SQLite. Tem maior segurança como a utilização de autenticação e gerenciamento de usuários, maior escalabilidade e é ideal para aplicações maiores.

4 RESULTADOS

Nesta seção, apresentam-se os resultados de todo o procedimento adotado.

4.1 Descrição do Sistema

O sistema facilita a busca por locais de descarte e de doações que estejam próximos ao usuário, filtrando o item que deseja entregar. Ele poderá visualizar a descrição do local e dos itens, clicar para ver a rota no Google Maps e também entrar em contato com eles via e-mail ou WhatsApp. Além disso, um usuário que possua algum ponto de coleta próprio, seja ele de doação, descarte ou reciclagem, poderá se cadastrar e alimentar a plataforma com seu próprio local, gerenciar seus

pontos de coleta, seu perfil e os itens que recebe.

4.2 Modelagem e configurações do Sistema

Para facilitar o desenvolvimento do projeto, foi utilizado bibliotecas javascript em todo o sistema, independente da plataforma. Além das ferramentas já descritas na seção anterior, o projeto também teve a importação de outras bibliotecas que foram necessárias durante o desenvolvimento.

Um exemplo é o *JsonWebToken*, uma biblioteca que cuida da autenticação no sistema através do gerenciamento de *tokens*. Ele gera um *token* aleatório durante a criação de usuário que se renova cada vez que o usuário se autentica, com um determinado prazo de validade. Com o *token* armazenado no *Local Storage* do navegador ou no cabeçalho da requisição, o usuário consegue acessar rotas privadas conforme o seu perfil sem a necessidade de se autenticar novamente.

Já a biblioteca *Knex* serviu para a conexão com os bancos de dados do projeto, ele é um *query builder* de consulta SQL flexível que facilita e abstrai a conexão com os bancos dados Postgres, MSSQL, MySQL, MariaDB, SQLite3, Oracle e Amazon Redshift através de pequenas configurações (KNEX, 2020). Ele traz maior facilidade em casos de migração de banco de dados, permitindo alterar para qualquer banco sem a necessidade de alterar as consultas.

Para a validação de caminhos de rotas de API HTTP foi utilizado a biblioteca *Celebrate*. Ela é um *middleware* que garante que as requisições sigam somente se todas as entradas digitadas pelo usuário estiverem corretas. Faz a utilização de *Joi*, uma linguagem de definição e validação para objetos JavaScript e permite criar sua própria estrutura de regras de entrada.

Na criação da API foi utilizado o *Express*, um *framework* Node.js para servidores HTTP, *open source* que otimiza a construção de aplicações *web* e API's. Sua filosofia é fornecer ferramentas pequenas e robustas para servidores HTTP, tornando-o uma ótima solução para aplicativos no modelo SPA, sites, híbridos ou APIs públicas.

O projeto foi configurado para a liberação de requisições através da biblioteca *CORs*, instalado pelo *npm*, de modo que aceite todas as requisições mesmo que estejam em domínios diferentes, habilitando ou bloqueando o *Cross-Origin Resource Sharing* que já está no cabeçalho HTTP. Afinal, ele quem determina se a página será

redirecionada ou não, evitando conflitos entre o cliente-servidor.

O Axios foi inserido no projeto ao lado do cliente para efetuar requisições para a API e faz todo o gerenciamento de requisições e respostas. Já o bcrypt serviu para transformar as senhas dos usuários em *hash* com o adição do *salt* antes de salvá-los no banco de dados, resolvendo o *hash* quando necessário e aumentando a proteção contra ataques.

O mapa utilizado no sistema *web* é de uma biblioteca javascript *open source* para mapeamento e interação com a geolocalização na *web*, chamado de leaflet. A estilização do mapa está no padrão da biblioteca, mas os marcadores e o *popup* foram customizados para este sistema.

Também foi utilizado o Multer, um *middleware* que configura *uploads* de arquivos em geral. Ele foi configurado para que localmente salvasse imagens dos pontos de coleta dentro de um arquivo nomeado de "uploads", localizado na raiz do projeto, e utilize esse arquivo para retornar a imagem do ponto de coleta. Assim como configurado localmente, ele precisou ser configurado de maneira diferente para a produção. Como o Heroku limpa imagens salvas em disco periodicamente, foi necessário criar um *bucket* AWS S3 e conectá-lo ao Node.js para armazenar imagens estáticas da aplicação.

A Tabela 1 apresenta as principais tecnologias utilizadas e as suas versões.

Tabela 1: Tecnologias e ferramentas

Tecnologias	Versão	Finalidade
Javascript	-	Linguagem de programação.
Sqlite	5.0.0	Banco de dados local.
MySQL	2.18.1	Banco de dados de produção.
Node.js	15.7.0	Ambiente de execução server-side.
React	17.0.1	Desenvolvimento de interfaces web.
React Native	0.63.2	Desenvolvimento do aplicativo Android/iOS.
Knex	0.21.8	Query builder para conexão e consulta DBs.
JsonWebToken	8.5.1	Autenticação
Celebrate	13.0.4	Validação de rotas
Express	4.17.1	Gerenciamento HTTP
Cors	2.8.5	Controla se o recurso pode ser acessado
Axios	0.21.0	Cliente HTTP para fazer requisições
BcryptJs	2.4.3	Encriptação de dados
Multer	0.21.8	Middleware de gerenciamento de arquivos
Aws-Sdk	2.x	Bucket para upload de imagem em produção
Material UI	4.11.2	Estilização das páginas
Expo	40.0.0	Facilita acesso às APIs nativas mobile
Leaflet	1.7.1	Mapa utilizado na página WEB

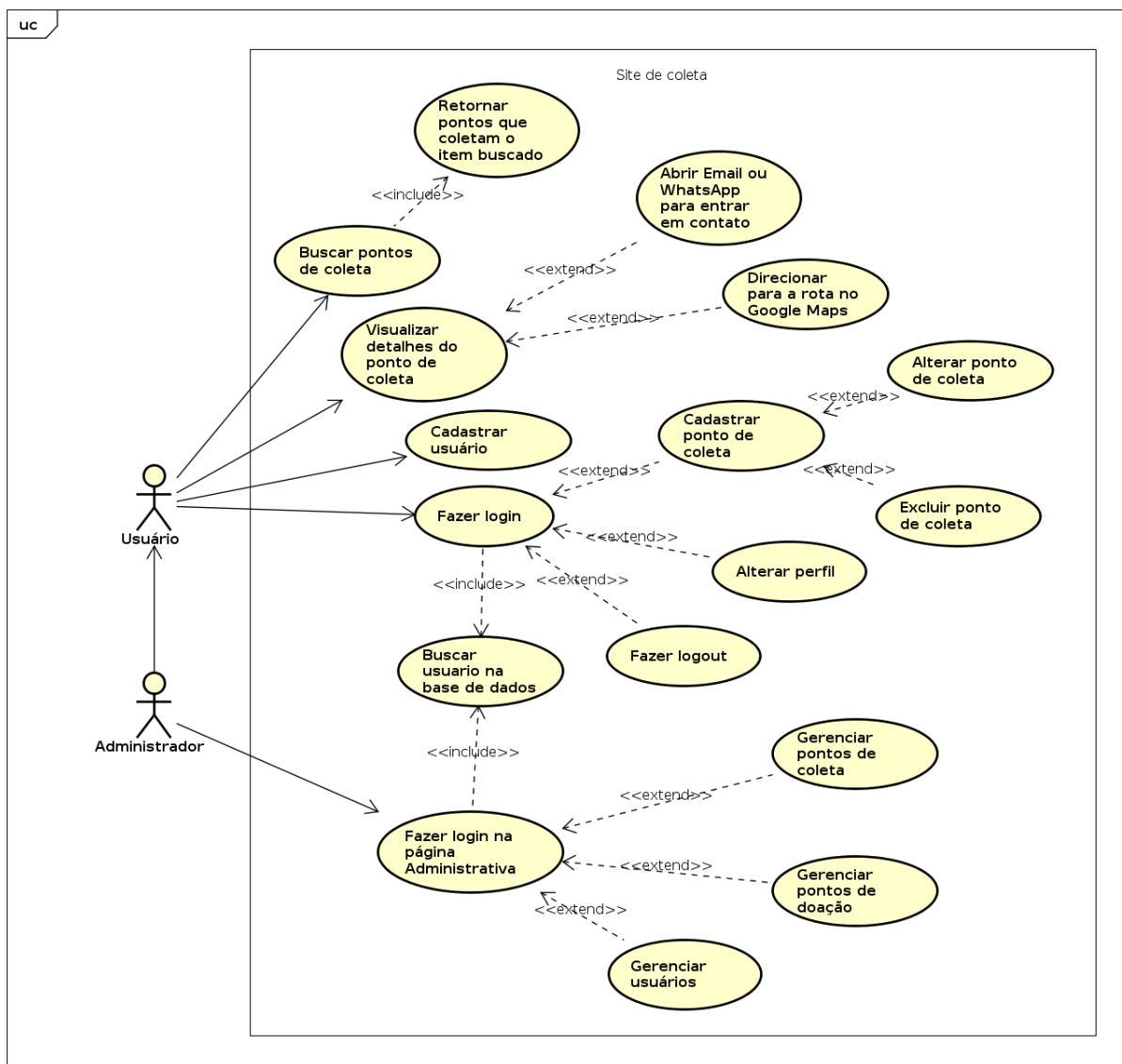
Fonte: Elaborado pela autora, 2021.

Também se fez necessário o levantamento de requisitos e a modelagem do sistema para alcançar os objetivos. Eles melhoram a visibilidade de todo o processo e dos problemas que seriam enfrentados.

Foi essencial pensar na facilidade da utilização do sistema, com uma busca rápida por pontos de coleta, eliminando dificuldades ou muitos cliques em tela. Assim como foi preciso pensar na segurança dos dados que seriam persistidos e na segurança de rotas privadas.

Para entender o funcionamento da aplicação e seus requisitos foi elaborado o diagrama de Casos de Uso do site (Figura 1), o diagrama de Casos de Uso do aplicativo (Figura 2), o modelo ER (Figura 3) e o DER (Figura 4).

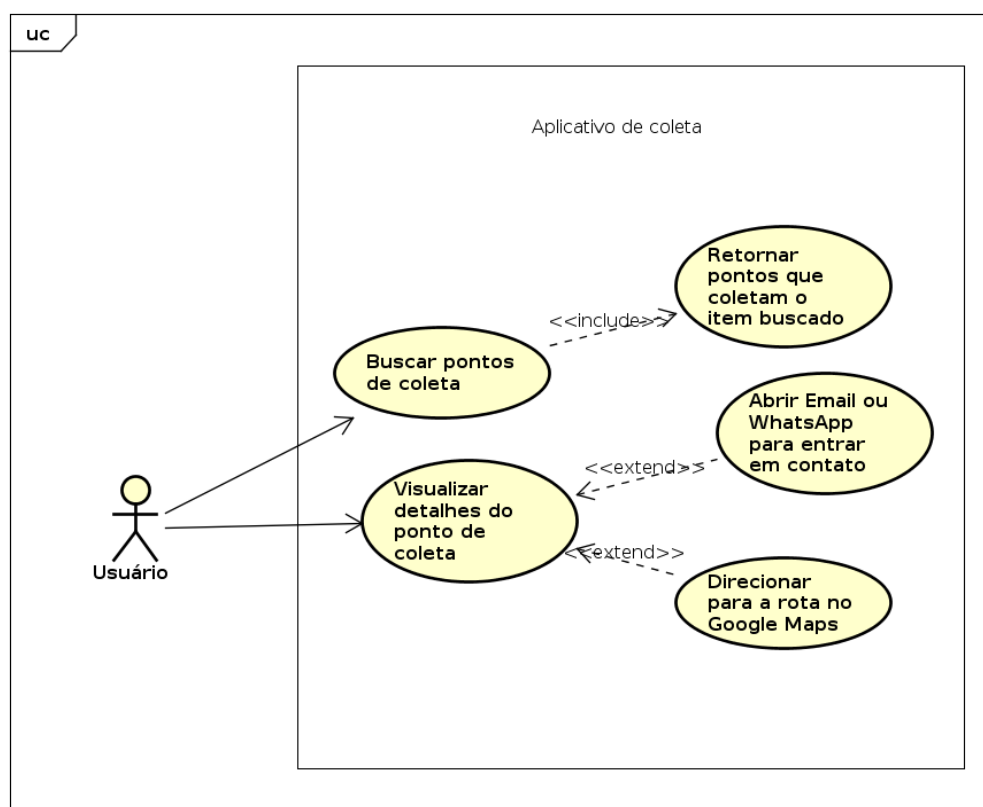
Figura 1: Diagrama de casos de uso do site



A Figura 1 demonstra as principais funcionalidades do sistema *web*, incluindo interação com as duas categorias de usuários, sendo o usuário comum e o administrativo. O usuário comum consegue buscar pontos de coletas que estejam próximos a ele, consegue ver os detalhes deste local, entrar em contato e gerar uma rota no Google Maps apresentando a distância entre eles.

Além disso, o usuário consegue se cadastrar no sistema, criar seus próprios pontos de coleta e gerenciá-los. Já o usuário que é administrador, poderá gerenciar todos os pontos de coleta e todos os usuários da aplicação.

Figura 2: Diagrama de caso de uso do aplicativo



Fonte: Elaborado pela autora, 2021.

Conforme o diagrama que representa as funcionalidades do aplicativo móvel, visto na Figura 2, percebemos serem mais simples que no sistema *web*.

O aplicativo permite que o usuário abra um mapa de pontos de coletas ou um mapa de pontos de doações e busque-os através dos itens que deseja descartar.

No mapa ele conseguirá clicar sobre o ponto, ver os detalhes sobre ele e entrar em contato se desejar. Logo, a aplicação não possui outras funcionalidades como o gerenciamento de cadastros de usuário, *login* ou gerenciamento dos pontos

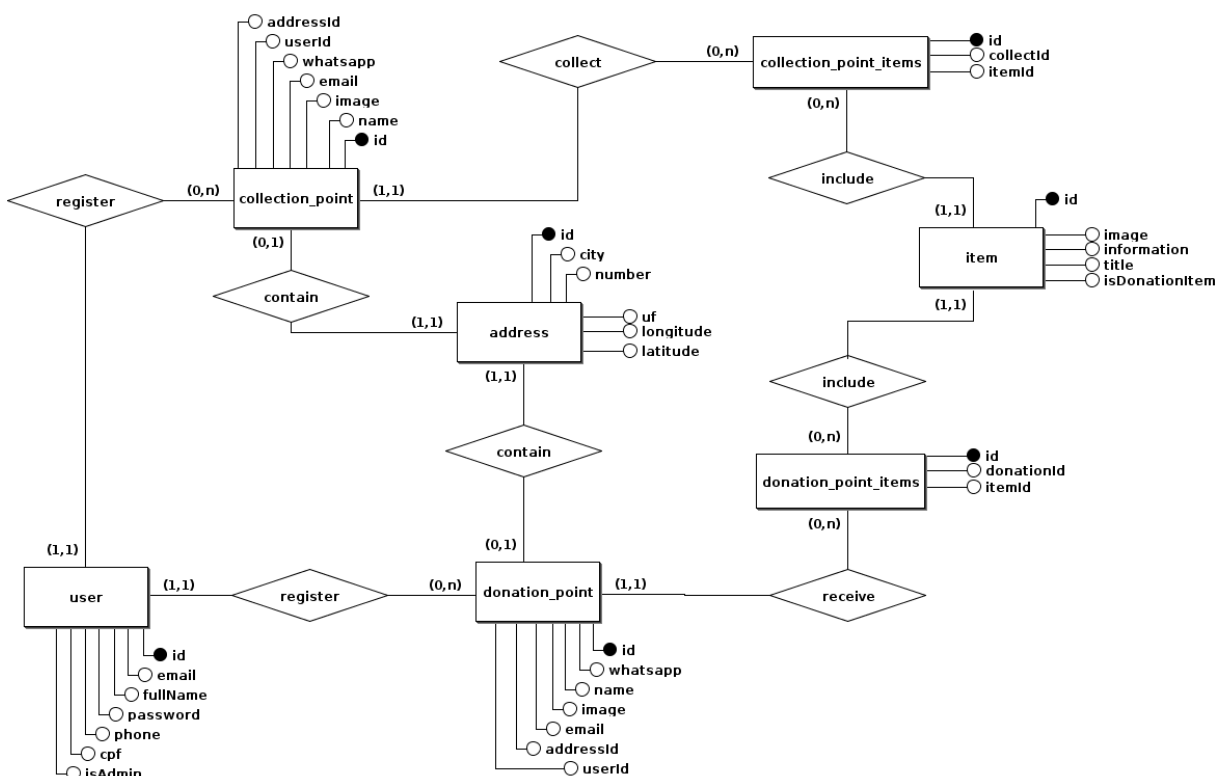
de coleta.

A respeito do modelo e do diagrama entidade relacionamento veremos as entidades criadas no banco de dados e a forma em que elas se relacionam entre si. Ou seja, o usuário é o indivíduo que registra os pontos de coleta e de doação, estes pontos possuem relacionamento com uma tabela de endereço, e se relacionam com a tabela itens.

Esse relacionamento entre o ponto de coleta e os itens é de muitos para muitos (N:N), que funciona criando outra tabela que representa essa categoria, denominados de *collection point itens* e *donation point itens*.

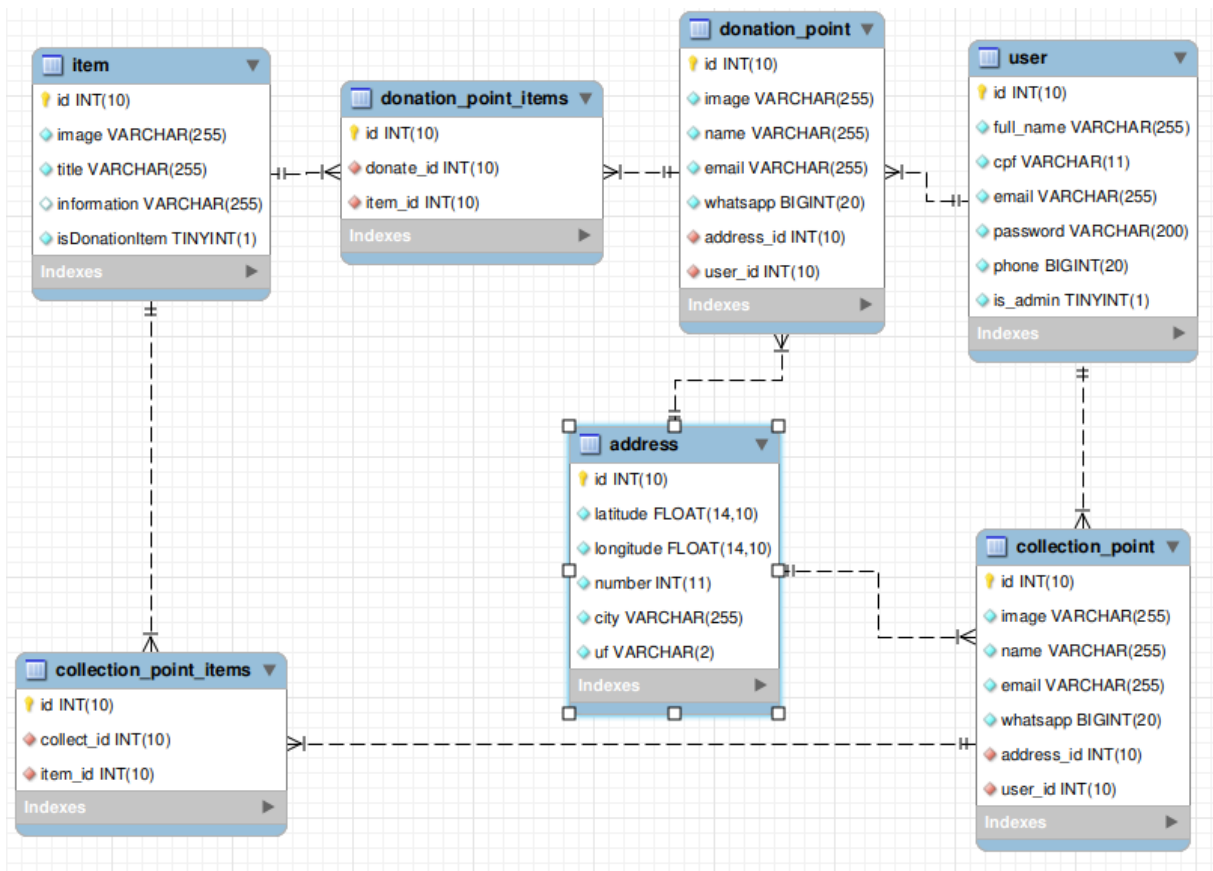
No armazenamento de itens, ao todo são 9 categorias de itens para filtragem de descarte, e 6 tipos para doação. A tabela item é a única entidade que não possui os dados alterados durante a execução do sistema. Por serem generalizados, eles são inseridos todos na tabela durante a primeira conexão com o banco de dados.

Figura 3: Modelo entidade relacionamento (MER) do banco de dados



Fonte: Elaborado pela autora, 2021.

Figura 4: Diagrama entidade relacionamento do banco de dados



Fonte: Elaborado pela autora, 2021.

O diagrama apresentado na Figura 4 foi gerado através da ferramenta MySQL Workbench. Nele, conseguimos analisar como o banco de dados está estruturado conforme o relacionamento das entidades, os dados que cada entidade armazena e o tipo deste dado.

Quando os repositórios estavam prontos, foi decidido pelo *deploy* em uma plataforma em nuvem, gratuita e de fácil utilização, chamada Heroku (Ver Figura 5). Tanto o servidor quanto a página *web* foram configurados nesta plataforma. Para isso, foi necessário refatorar as configurações do projeto e separar configurações locais com a de produção. Foi criado em cada um dos projetos um arquivo de variável de ambiente *.env* (oculto) onde é controlado o que cada ambiente vai utilizar.

Essas variáveis incluem credenciais do banco de dados, link da API que vai servir os dados, a porta em que o servidor vai ouvir localmente ou em produção, o link do direcionamento do *upload* de imagens (local ou Amazon S3), as chaves de acesso do AWS e também a chave que resolve *tokens* de autenticação. Quanto ao

aplicativo mobile, ele está publicado e disponível no Google Play aguardando atualização.

Figura 5: Página inicial do Heroku demonstrando o servidor

The screenshot shows the Heroku dashboard for the 'coletali-server' application. At the top, there's a search bar and navigation links for 'Personal' and 'coletali-server'. Below that, there are tabs for 'Overview', 'Resources', 'Deploy', 'Metrics', 'Activity', 'Access', and 'Settings'. The main content area is divided into several sections: 'Installed add-ons' (ClearDB MySQL), 'Dyno formation' (free dynos), 'Collaborator activity' (vanessa.r15@aluno.ifsc.edu.br), and 'Latest activity' (a list of recent events like 'Set config var', 'Remove config var', 'Deployed', and 'Build succeeded').

Fonte: Elaborado pela autora, 2021.

Figura 6: Página do bucket Amazon S3

The screenshot shows the Amazon S3 console for the 'coletali-bucket'. The page has a navigation bar with tabs for 'Objetos', 'Propriedades', 'Permissões', 'Métricas', 'Gerenciamento', and 'Pontos de acesso'. The main content area is titled 'Objetos (10)' and contains a list of objects. The table below shows the details of three objects:

	Nome	Tipo	Última modificação	Tamanho	Classe de armazenamento
<input type="checkbox"/>	002cc8d6fa93e707818eca41c6cdbfc6-photo-1611284446314-60a58ac0deb9.jpeg	Jpeg	31 Mar 2021 08:18:28 PM -03	46.5 KB	Padrão
<input type="checkbox"/>	0ade8db52d59f296c51112fe5d9beadc-photo-1593113630400-ea4288922497.jpeg	Jpeg	27 Mar 2021 06:36:17 PM -03	42.2 KB	Padrão
<input type="checkbox"/>	17b2400e3982cab717fdb5b9913c65bb-photo-1611284446314-60a58ac0deb9.jpeg	Jpeg	31 Mar 2021 08:18:26 PM -03	46.5 KB	Padrão

Fonte: Elaborado pela autora, 2021.

Através da imagem do painel do Heroku, vemos que o servidor Node.js está utilizando um *add-on* chamado ClearDB MySQL. Uma solução simples oferecida pela plataforma para a hospedagem do MySQL gratuitamente. Foi ele quem criou o banco de dados MySQL da aplicação, através das configurações Knex.js do projeto, e ofereceu as credenciais para poder conectar em qualquer ferramenta.

Já na sequência vimos a página do *bucket* do Amazon S3 (Figura 6), que é onde estão sendo armazenadas as imagens de *upload* dos pontos de coleta em produção. Para a configuração dele foi necessário liberar permissões de CORs e de acesso público. Além disso, foi necessário configurar um usuário IAM da Amazon - *Identity and Access Management*, associá-lo ao *bucket*, liberar permissão para gerenciamento S3 e obter credenciais de acesso para conectá-lo à API.

4.3 Apresentação do Sistema

Como mencionado anteriormente, o desenvolvimento do sistema se deu início pelo servidor. Com ele foi implementado todas as rotas que a API iria servir, conexão com o banco de dados e testado todas as requisições como cliente dentro de um *workspace* do projeto na ferramenta Insomnia.

O site foi iniciado logo após a conclusão do servidor, com a utilização de dados estáticos para que a estilização das telas fossem priorizadas na responsividade e na usabilidade. As cores utilizadas no site foram definidas antes do desenvolvimento, assim como foi definido um padrão de *layout* globalmente.

Durante o desenvolvimento web, foi aplicado um conhecimento anteriormente adquirido da biblioteca React. Também foi obtido suporte de uma biblioteca de componentes, chamada de Material-UI, para a estilização. A aplicação possui componentes que foram estilizados pela ferramenta e também componentes que foram de criação própria. As imagens disponíveis na página *web* foram buscadas através de bancos de imagens gratuitas como a Unsplash e a ManyPixels e também foram utilizados ícones que estão disponíveis na plataforma Flaticon.

A Figura 7 mostra a página inicial do site. Ele permite ao usuário acessar diretamente a aplicação ou as páginas administrativas. Para acessar a aplicação não é necessário fazer a autenticação, já o administrador será redirecionado para inserir suas credenciais e acessar as páginas administrativas.

Figura 7: Landing Page da aplicação web

COLETALI

Reuse, Reduza, Doe e Recicle

Ajudamos as pessoas a encontrar pontos próximos e eficazes para o descarte ou doação de materiais. Colaborando com as pessoas e o meio ambiente :)

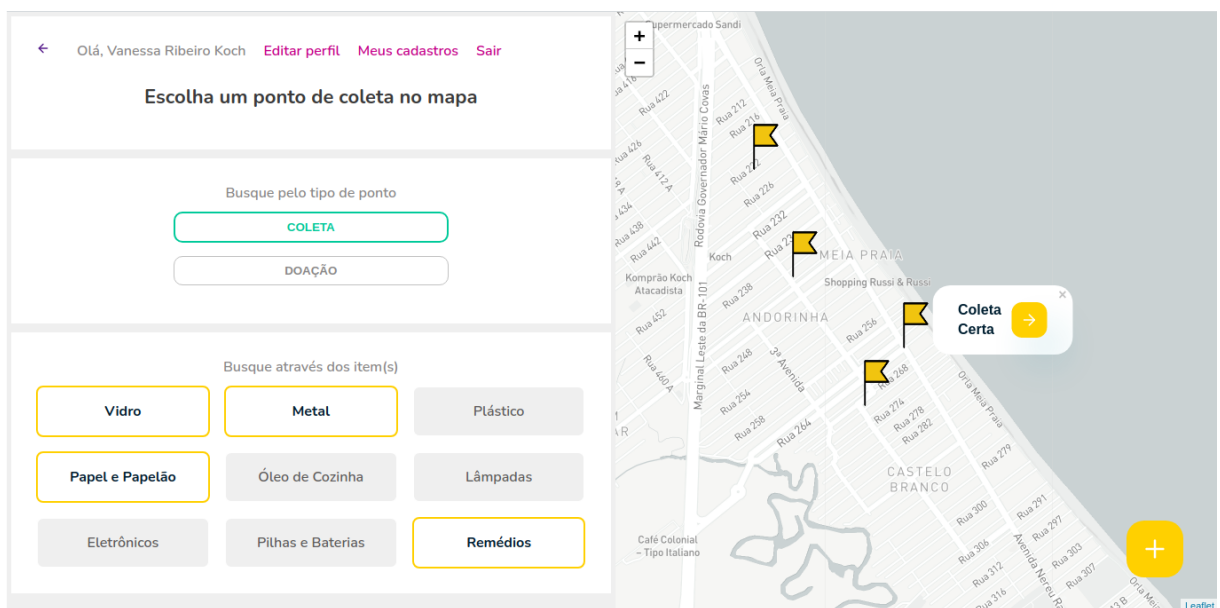


→ Acessar

🔑 Sou administrador

Fonte: Elaborado pela autora, 2021.

Figura 8: Mapa da aplicação web



Fonte: Elaborado pela autora, 2021.

A Figura 8 representa a página principal do site, onde o usuário pode interagir, independente da autenticação. Ao abrir esta página, o navegador solicitará uma permissão para a localização do dispositivo, para que então o mapa seja carregado conforme a localização de quem estiver acessando.

Nessa página o usuário poderá filtrar por locais que recebem doações ou

coletas, conforme os itens que desejam descartar. Os locais que coletam o item selecionado vão aparecer no mapa à direita e, ao ser clicado, realizará o redirecionamento para a página de detalhes, onde consta os dados desse local.

Além disso, vemos no menu superior links que redirecionam para as páginas de editar o perfil e gerenciar seus próprios pontos de coleta. No primeiro o usuário poderá alterar todos os seus dados, incluindo nome, e-mail, senha, telefone e correção do CPF. No último, será apresentado uma tabela com todos os pontos de coletas ou de doações criadas pelo usuário logado, permitindo editar ou remover estes pontos da aplicação. Porém, estes links somente aparecerão para o usuário que estiver autenticado no sistema.

Para que o usuário veja os detalhes de um ponto de coleta, ele deverá clicar no *popup* do local correspondente dentro do mapa e será redirecionado para outra página que contém estes dados.

Através dela, o usuário conseguirá visualizar a imagem do local e todos os itens que ele recebe. Poderá também clicar no botão de mapa para redirecioná-lo para as rotas do Google Maps, permitindo ver o caminho entre ele e o coletor.

Além das rotas, a página também permite que o usuário clique no botão do whatsapp e abra uma conversa através do whatsapp *web* ou pressione o botão do e-mail para entrar em contato com o local. Neste botão, o usuário será redirecionado para a plataforma do gmail com uma nova mensagem para ser enviada ao local de coleta. Essa página de detalhes de um ponto coleta será demonstrada a seguir na figura 9 e na sequência teremos a figura 10 demonstrando a página de cadastro de um ponto de coleta.

Para cadastrar um novo ponto de coleta, seja ele de doação, reciclagem ou descarte, o usuário deve fazer o cadastro prévio na página *web* e se autenticar. O sistema permite somente um cadastro por e-mail e um usuário poderá cadastrar quantos pontos de coleta acharem necessários. O *login* deve ser feito antes de acessar a página de cadastro e, sem ele, a rota privada será redirecionada à página de *login* novamente.

Todos os dados do coletor devem ser preenchidos. O usuário ao cadastrar deve selecionar exatamente o local do ponto de coleta no mapa, podendo alterar se for preciso. Os dados que estão nos campos de estado e cidade vem da API do IBGE e também devem ser selecionados corretamente.

Além disso, o usuário deve definir a categoria de ponto de coleta em que ele

se encaixa, doação ou apenas coleta. Ele deve dispor de uma foto do ambiente e selecionar os itens que o estabelecimento recolhe, podendo ser mais de um item. Somente após o processo de inserção concluído que o ponto de coleta aparecerá no mapa da web e no mapa da aplicação móvel.

Figura 9: Detalhes do ponto de coleta da aplicação web

COLETALI

← Página Inicial

Coleta Certa

Itens de coleta

Metal
Alumínio; Cobre; Ferro; Aço.

Óleo de Cozinha
Armazenar a sobra da fritura em um recipiente ou garrafa PET e entregar ao posto de coleta.

Vidro
Coloque-o em caixas de papelão, jornais ou sacos grossos, lembrando de etiquetá-los descrevendo o conteúdo.

Localização

Itapema - SC, Número 2222

[Ver a rota no Google Maps](#)

Contato

[WhatsApp](#) [Enviar e-mail](#)

Fonte: Elaborado pela autora, 2021.

Figura 10: Cadastro de ponto de coleta da aplicação web




Cadastro do ponto de coleta

Dados

Nome


E-mail WhatsApp

Endereço Selecione o endereço no mapa

A map showing a residential area with streets labeled: Rua 700, Rua 406, Rua 128, Rua 142, Praia Central, V. Est. 1000, Rua 615, and Rua 606. A restaurant named "Restaurante Rech" is also visible. The area is labeled "TABULEIRO DOS OLIVEIRAS".







Número Estado (UF) Cidade

Tipo da localização



Clique ou arraste uma imagem para o ponto de coleta

Ítems de coleta Selecione um ou mais ítems abaixo

 Alimentos	 Brinquedos	 Roupas
 Calçados	 Cobertores	 Móveis

Ícones feitos por Freepik em www.flaticon.com

Fonte: Elaborado pela autora, 2021.

O usuário que estiver autenticado conseguirá navegar até a lista de locais de coleta que ele mesmo tenha inserido no sistema, conforme demonstrado na Figura 11. Nessa lista, além de visualizar todos os pontos de coleta ou de doação que tenha adicionado, ele também consegue editar todos os dados do ponto ou apagar o registro.

Figura 11: Lista dos pontos de coleta do usuário

COLETALI

← Página inicial

Pontos cadastrados por você

Selecione o tipo do local

Coleta

Nome ↑	E-mail	WhatsApp	Cidade	Estado	Ações
Coleta Certa	certa@gmail.com	47996339690	Itapema	SC	EDITAR DELETAR
Farmacia Atual	farmacia_atual@gmail.com	47996339690	Itapema	SC	EDITAR DELETAR
Recicla Aqui	aqui@gmail.com	47999999999	Itapema	SC	EDITAR DELETAR
Reciclaria	reciclaria@gmail.com	47995522114	Itapema	SC	EDITAR DELETAR

Rows per page: 5 1-4 of 4

Fonte: Elaborado pela autora, 2021.

Na criação de usuário para o sistema, qualquer um poderá se cadastrar, mas não poderá conter cadastros com o mesmo e-mail. A página de *login* do sistema se assemelha com a página de cadastro de usuário, demonstrada na Figura 12.

Figura 12: Formulário de cadastro de usuário

[← Página inicial](#)



Cadastro de Usuário

[Já possui uma conta? Entrar](#)

Fonte: Elaborado pela autora, 2021.

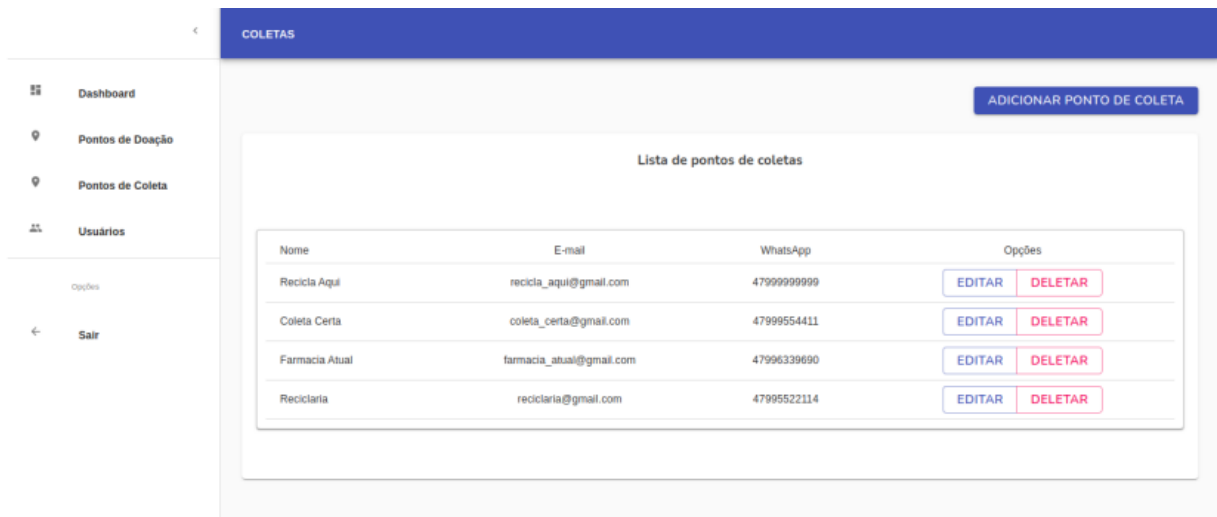
Por questões de segurança, quando o usuário cadastrar sua senha, o sistema gera um *salt* com 10 caracteres aleatórios e um *hash* a partir de uma chave, fazendo a união dos dois para então salvá-los no banco de dados no lugar da senha. Ou seja, no momento de *login* quando o usuário inserir suas credenciais, a sua senha digitada será comparada ao *hash* do banco de dados, durante uma função de verificação. Isso permite uma maior segurança a ataques e que não seja salvo dois valores iguais no banco. Tanto para o cadastro do usuário quanto para o *login*, o sistema gera um *token* novo quando submetido, com 30 minutos de duração. O token é armazenado no *local storage* do navegador e quando o usuário acessar uma rota privada, o sistema vai verificar se o *token* é válido ou se o *token* pertence a um administrador. Caso o *token* não seja válido e o usuário acesse rotas privadas, o sistema não permitirá e redireciona o usuário para a rota de *login* pertencente, podendo ser a do *login* administrativo ou a do sistema padrão.

Todos os formulários do sistema *web* estão com validações, seja como *middleware* de rotas ou dos próprios formulários, e retornam notificações de erros ou de sucesso para auxiliar o usuário e salvar dados corretamente.

As páginas administrativas do sistema só podem ser acessadas através do *login* de um usuário administrador. Nela o administrador consegue visualizar todos os pontos de coleta, doações e todos os usuários da aplicação. Além de visualizar os registros de todos os dados, ele consegue editar, remover ou até adicionar um novo registro. O sistema possui 10 rotas administrativas que suprem essas funções,

um exemplo está na Figura 13.

Figura 13: Página administrativa: Lista de todas as coletas da aplicação



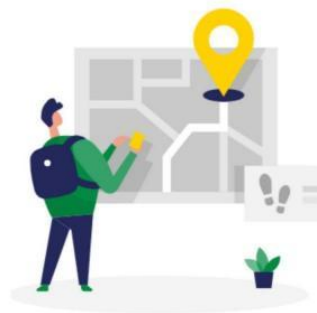
The screenshot shows a web interface for managing collection points. On the left is a sidebar menu with options: Dashboard, Pontos de Doação, Pontos de Coleta, Usuários, Opções, and Sair. The main content area is titled 'COLETAS' and contains a button 'ADICIONAR PONTO DE COLETA'. Below this is a table titled 'Lista de pontos de coletas' with the following data:

Nome	E-mail	WhatsApp	Opções	
Recicla Aqui	recicla_aqui@gmail.com	47999999999	EDITAR	DELETAR
Coleta Certa	coleta_certa@gmail.com	47999554411	EDITAR	DELETAR
Farmacia Atual	farmacia_atual@gmail.com	47996339690	EDITAR	DELETAR
Reciclaria	reciclaria@gmail.com	47995522114	EDITAR	DELETAR

Fonte: Elaborado pela autora, 2021.

Após a conclusão das páginas web e conexão com a API do servidor, foi iniciado o desenvolvimento do aplicativo mobile com o React Native.

Figura 14: Página inicial do aplicativo



COLETALI

Seja bem vindo :)

O que deseja fazer ?

Doar

Descartar

Fonte: Elaborado pela autora, 2021.

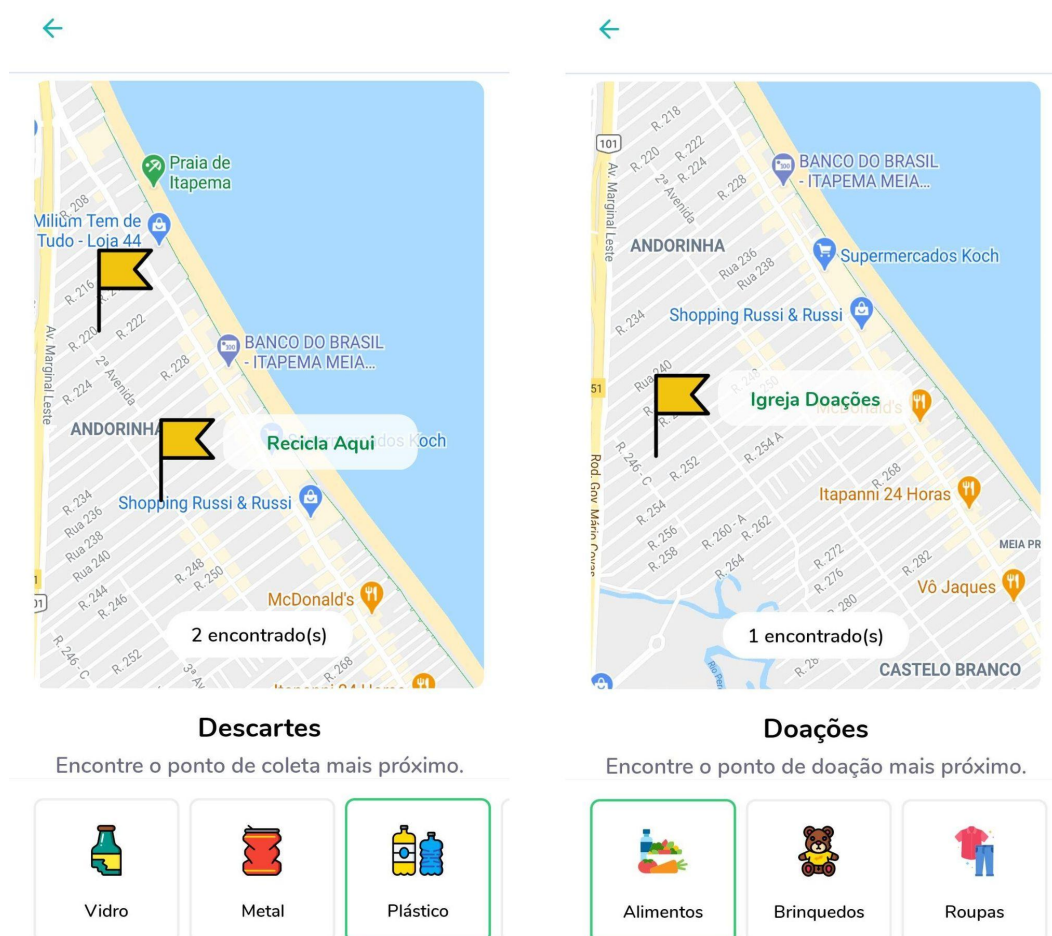
O aplicativo foi desenvolvido de maneira mais rápida por possuir menos

funcionalidades. A página inicial segue com as mesmas cores utilizadas na *web* e com o mesmo estilo de imagem. Na Figura 14 vemos que ela contém apenas dois botões, um para navegar até o mapa de pontos de coletas e outro para navegar até o mapa de pontos de doações. Diferente da página *web*, que possui uma rota que compõem as duas categorias de objetos.

Após clicar em um dos botões, o dispositivo vai solicitar permissão para a localização, para que assim, como na *web*, o aplicativo abra o mapa conforme a localização atual do usuário.

As duas rotas que contém o mapa são idênticas, mudam apenas o tipo do ponto a ser encontrado e os itens que pertencem ao filtro. Diferentemente da *web*, que foi utilizado o mapa da biblioteca leaflet nas páginas, aqui na aplicação mobile foi utilizado o próprio Google Maps, disponibilizado pelo React Native.

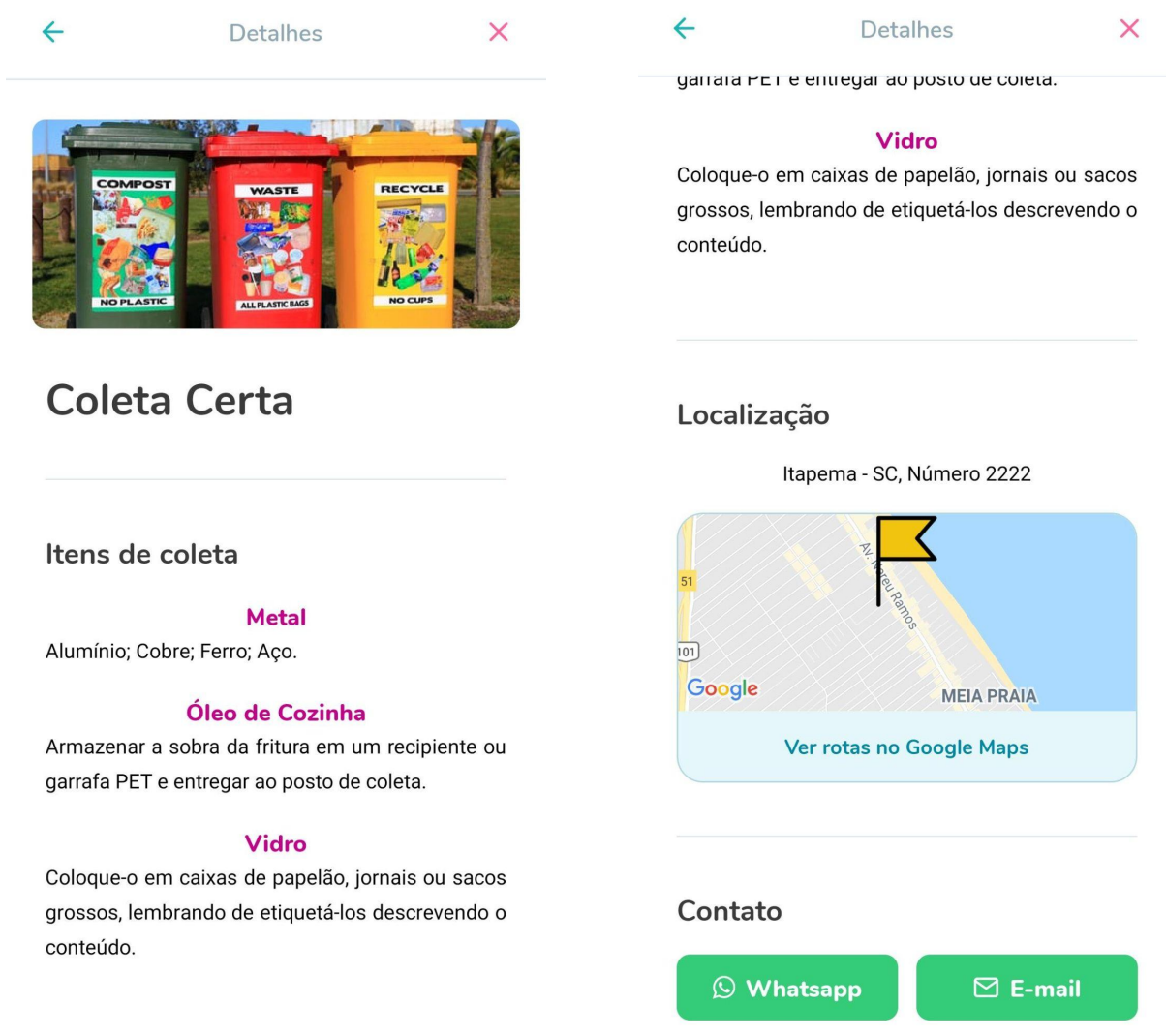
Figura 15: Mapa dos pontos de coleta da aplicação mobile.



Fonte: Elaborado pela autora, 2021.

O mapa do aplicativo (Figura 15) também retorna os pontos conforme a busca na seleção dos itens, além dos marcadores no mapa, ele possui um indicador de quantos pontos coletam o item buscado. O marcador ao ser clicado também abrirá um *popup*, que ao clicar vai abrir a página de detalhes e informações daquele local. O espaço que dispõe dos itens dessa página é um *scroll* horizontal. Ao rolar os itens para a esquerda, aparecerão outros itens da categoria escolhida para selecionar.

Figura 16: Detalhes de um ponto de coleta no aplicativo.



← Detalhes ×

← Detalhes ×

garrafa PET e entregar ao posto de coleta.

Vidro

Coloque-o em caixas de papelão, jornais ou sacos grossos, lembrando de etiquetá-los descrevendo o conteúdo.

Coleta Certa

Itens de coleta

Metal

Alumínio; Cobre; Ferro; Aço.

Óleo de Cozinha

Armazenar a sobra da fritura em um recipiente ou garrafa PET e entregar ao posto de coleta.

Vidro

Coloque-o em caixas de papelão, jornais ou sacos grossos, lembrando de etiquetá-los descrevendo o conteúdo.

Localização

Itapema - SC, Número 2222

Ver rotas no Google Maps

Contato

WhatsApp E-mail

Fonte: Elaborado pela autora, 2021.

Assim como na *web*, o usuário consegue ver as mesmas informações na página de detalhes de um local cadastrado (Figura 16). A fotografia do local, os itens que ele coleta e também inclui o botão de rotas do Google Maps, que vai

direcioná-lo para o aplicativo oficial e calcular a rota para o usuário até o ponto de coleta. No final da página também inclui um botão de WhatsApp e de e-mail que abrirão suas respectivas aplicações para que o usuário entre em contato com o ponto de coleta.

Os resultados desse trabalho trouxeram comunicação instantânea entre a página *web* e o aplicativo. Os usuários podem personalizar a pesquisa e encontrar os locais de coleta ideais através de um computador ou celular, contando com inúmeros benefícios à população em casos sociais de carência, saúde e meio ambiente.

5 CONCLUSÃO

Percebemos que o sistema *web* contém toda a regra do negócio e muito mais funcionalidades que o aplicativo, mas assim mesmo, os dois estão conectados na mesma API e trazem a principal funcionalidade do sistema.

Um dos maiores desafios do projeto foi a customização de permissões de caminhos de rotas HTTP, uma para o acesso do usuário autenticado e outra para o usuário administrativo. Além dele, outro desafio foi a configuração dos três repositórios para o *deploy*, que se fez necessário alterar o projeto para utilização do banco de dados MySQL em produção, e foi necessário alterar a forma de armazenamento de *upload* de imagens dos pontos de coleta.

O desenvolvimento do presente projeto e a utilização do sistema, auxilia o usuário comum a conhecer o local correto em que deve descartar seus itens, contribuindo no processo de reutilização, reciclagem, transformação ou até na doação de itens que estejam em bom estado. Além de dar um melhor destino para os itens desperdiçados, evitando jogar em locais impróprios, também poderá entregar nos locais mais próximos de onde o usuário estiver.

Vemos benefícios também para aqueles usuários que possuem um local de coleta próprio, que faz a reciclagem desse item, o tratamento, a transformação ou que recolha doações e que queiram divulgar ou facilitar o encontro do seu estabelecimento. Além disso, a utilização do sistema e o cadastro desse ponto de coleta, poderá aumentar significativamente a quantidade de itens recebidos.

Mesmo o projeto tendo atingido o seu propósito, ele ainda poderá melhorar com o desenvolvimento de todas as funcionalidades presentes na *web* para o

aplicativo. Poderia ser substituído a linguagem de programação de JavaScript para TypeScript, usufruindo da tipagem da linguagem. Assim como, também sugiro o desenvolvimento de testes que auxiliem nas manutenções do projeto.

COLETALI:

A SYSTEM FOR IDENTIFYING DISPOSAL OR DONATION SITES

Abstract: This article presents a system to assist the population in identifying places for the disposal or donation of materials, reflecting on the treatment of these materials, so they can be recycled or reused, therefore contributing to the reduction of damages caused by incorrect disposal. A website and an application were developed using Javascript, currently the most used language in web development, including the use of NodeJs, React and React Native libraries. Its main function is to search for collection sites according to the location of the user's device, as well as allowing collection site owners to register and manage their services.

Keywords: Web Application. Mobile Application. Recycling. Donation.

REFERÊNCIAS

ADITYA, S. K.; MOHANTA, P.; KARN, V. K. **Android SQLite Essentials**. Packt Publishing, 2014. isbn: 9781783282951.

ANTANI, V.; TIMMS, S.; MANTYLA, D. **JavaScript: Functional Programming for JavaScript Developers**. Packt Publishing, 1a ed, isbn: 978-1-78712-466-0, 2016.

BLANTHORN, O. A.; CAINE, C. M.; NAVARRO-LÓPEZ, E. M. **Evolution of communities of software: using tensor decompositions to compare software ecosystems**. Applied Network Science, 2019. Disponível em: <https://appliednetsci.springeropen.com/articles/10.1007/s41109-019-0193-5>. Acesso em: 10 ago. 2020.

BODUCH, Adam. **React and React Native**. Packt Publishing, isbn: 9781786465658, 2017.

BOROVSKOY, I. G.; SHELMINA, E. A.; ZARIKOVSKAYA, N. V. **Text Data Storage Method Modification for SQLite Relational Database**. Journal of Physics: Conference Series, 2017, p. 012025. doi: 10.1088/1742-6596/803/1/012025. Disponível em: <https://iopscience-iop.ez130.periodicos.capes.gov.br/article/10.1088/1742-6596/803/1/012025/pdf>. Acesso em: 16 ago. 2020.

COX, Russ. **Surviving Software Dependencies**. Communications of the ACM, 2019, p. 36–43, issn: 00010782. Disponível em: <http://search-ebscohost-com.ez130.periodicos.capes.gov.br/login.aspx?direct=true&db=bsu&AN=138293522&lang=pt-br&site=ehost-live>. Acesso em: 11 ago. 2020.

CRISOSTIMO, Ana L. **Educação ambiental, reciclagem de resíduos sólidos e responsabilidade social: Formação de educadores ambientais**. Conexão UEPG, 2011. Disponível em: <https://www.revistas2.uepg.br/index.php/conexao/article/view/3687>. Acesso em: 20 jun. 2020.

EXPO. **Expo CLI**. Expo CLI - Documentation, 2021. Disponível em: <https://docs.expo.io/>. Acesso em: 12 ago. 2020.

FACEBOOK. **React**. React documentation and related resources, 2020. Disponível em: <https://reactjs.org/>. Acesso em: 12 ago. 2020.

FACEBOOK. **React Native**. React Native, 2020. Disponível em: <https://reactnative.dev/>. Acesso em: 12 ago. 2020.

FIDELITY CHARITABLE. **Giving Report**. New York: Fidelity Charitable, 2020. Disponível em: <https://www.fidelitycharitable.org/content/dam/fc-public/docs/insights/2020-giving-report.pdf>. Acesso em: 9 ago. 2020.

FLANAGAN, David. **JavaScript: O Guia Definitivo**. 6a ed. O'Reilly, isbn: 978-0-596-80552-4, 2011.

FONSECA, M. et al. **Os riscos relacionados ao ambiente e à atividade de coleta de resíduos sólidos urbanos**. Revista Verde de Agroecologia e Desenvolvimento Sustentável, 2014, p. 96–100. issn: 1981-8203. Disponível em: <https://www.gvaa.com.br/revista/index.php/RVADS/article/view/1978>. Acesso em: 8 ago. 2020.

GARCIA, M. B. S. et al. **Aspectos legais da coleta seletiva de lixo**. Revista Brasileira De Ciências Ambientais, 2016, p. 86–96. issn: 2176-9478. doi: 10.5327/Z2176-947820160065. Disponível em: http://rbciamb.com.br/index.php/Publicacoes_RBCIAMB/article/view/141/108. Acesso em: 8 ago. 2020.

HENNEN, D. S.; RAMACHANDRAN, S.; MAMRAK, S. A. **The Object JavaScript language**. Software: Practice and Experience, 30.14, p. 1571–1585, 2000.

HORTON, A.; VICE, R. **Mastering React**. Community Experience Distilled. Packt Publishing, 2016. isbn: 9781783558568.

KNEX. **Knex.js - A SQL Query Builder for Javascript**. Documentation, 2020. Disponível em: <https://knexjs.org/>. Acesso em: 16 ago. 2020.

LIMA, G. F. C. A.; JESUS, E. **Contribuição do Centro de Compostagem e Reciclagem na formação do Técnico em Ambiente e na educação ambiental**. Revista Brasileira de Ciências Ambientais, 2013, p. 34–44. Disponível em: http://rbciamb.com.br/index.php/Publicacoes_RBCIAMB/article/view/296. Acesso em: 20 jun. 2020.

LUNDAR, J. A; GRONLI, T.; GHINEA, G. **Performance evaluation of a modern web architecture**. Journal of Information Technology and Web Engineering, 2013. Disponível em: <https://link-gale.ez130.periodicos.capes.gov.br/apps/doc/A34821>

6707/AONE?u=capes&sid=AONE&xid=9198ec63. Acesso em: 10 ago. 2020.

MASIELLO, E.; FRIEDMANN, J. **Mastering React Native**. Packt Publishing, isbn: 9781785885785, 2017.

MATOS, E. T. A. R.; SANTOS F. R. A. **Impactos da rastreabilidade dos resíduos sólidos perigosos na atividade empresarial**. Revista de Direito Ambiental e Socioambientalismo, 2016, p. 107–129. Disponível em: <https://indexlaw.org/index.php/Socioambientalismo/article/view/1618>. Acesso em: 20 jun. 2020.

MCTIC. **Tecnologia Social**. Ministério da Ciência, Tecnologia e Inovações, 2017. Disponível em: <https://bit.ly/3cb2ID1>. Acesso em: 14 jun. 2020.

MORESI, E. A. D. et al. **Tecnologia Social e Empoderamento: O Caso do Aplicativo Solidarius**. Revista de Sistemas, Cibernética e Informática, 2017. Disponível em: <https://doaj.org/%20article/6abe5e980%2088f46508c1aca050c04aaf0>. Acesso em: 8 ago. 2020.

MOZILLA e colaboradores individuais. **MDN web docs**. Mozilla, 2020. Disponível em: <https://developer.mozilla.org/>. Acesso em: 10 ago. 2020.

NETO, C. A. A.; ROLT, C. R.; ALPERSTEDT, G. D. **Acessibilidade e Tecnologia na Construção da Cidade Inteligente**. Revista de Administração Contemporânea, 2018, p. 291–310. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-65552018000200291&nrm=iso. Acesso em: 21 jun. 2020.

NODE.JS. **Sobre Node.js**. OpenJS Foundation, 2020. Disponível em: <https://nodejs.org/>. Acesso em: 10 ago. 2020.

PRUSTY, Narayan. **Modern JavaScript Applications**. Community Experience Distilled. Packt Publishing, 2016, isbn: 9781785881442.

PUSKARIC, H. et al. **Development of web based application using SPA Architecture**. Proceedings on Engineering Sciences, 2019, p. 457–464. doi: 10.24874/PES01.02.044. Disponível em: <https://doaj.org/article/c217ef96d5734eeba118ab01427d8635>. Acesso em: 15 ago. 2020.

REICHERT, G. A; MENDES, C. A. B. **Avaliação do ciclo de vida e apoio à decisão em gerenciamento integrado e sustentável de resíduos sólidos urbanos**. Engenharia Sanitaria e Ambiental, 2014, p. 301–313. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-41522014000300301&nrm=iso. Acesso em: 14 jun. 2020

SANGJUN, J. et al. **A recovery method of deleted record for SQLite database**. Personal and Ubiquitous Computing, 2012, p. 707–715. issn: 16174909. Disponível em: <http://search-ebsohost-com.ez130.periodicos.capes.gov.br/login.aspx?direct=true&db=iib&AN=78030578&lang=pt-br&site=ehost-live>. Acesso em: 16 ago. 2020.

SHUTE, Zachary. **Advanced Javascript**. Packt Publishing, 2019. isbn: 978-1-78980-010-4.

SOARES, L. S. V.; MADUREIRA, A. S. **Cenários que desafiam a implementação emergencial da política nacional de resíduos sólidos: Uma questão ambiental e de saúde**. Revista de Direito e Sustentabilidade, 2018. Disponível em: <https://indexlaw.org/index.php/revistards/article/view/4954>. Acesso em: 13 jun. 2020.

SOLOVEI, V.; OLSHEVSKA, O.; BORTSOVA, Y. **The difference between developing Single Page Application based on mechatronics robot laboratory onaft application**. Automation technological and business processes, 2018. doi: 10.15673/atbp.v10i1.874. Disponível em: https://www.researchgate.net/publication/324380010_THE_DIFFERENCE_BETWEEN_DEVELOPING_SINGLE_PAGE_APPLICATION_AND_TRADITIONAL_WEB_APPLICATION_BASED_ON_MECHATRONICS_ROBOT_LABORATORY_ONAFT_APPLICATION. Acesso em: 15 ago. 2020.

SONG, Q.; LI, J.; ZENG X. **Minimizing the increasing solid waste through zero waste strategy**. Journal of Cleaner Production 104, 2015, p. 199–2010. Disponível em: <https://www-sciencedirect.ez130.periodicos.capes.gov.br/science/article/pii/S095965261400849X>. Acesso em: 13 jun. 2020.

SQLITE. **What Is SQLite?** SQLite Open Source, 2020. Disponível em: <https://www.sqlite.org/>. Acesso em: 15 ago. 2020.

STACK OVERFLOW. **Developer Survey, 2020**. Stack Overflow, 2020. Disponível em: <https://insights.stackoverflow.com/survey/2020>. Acesso em: 9 ago. 2020.

ZAMMETTI, Frank. **Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, and Docker**. 6a ed, isbn: 978148-4257388, Apress, 2020.