

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM INOVAÇÕES TECNOLÓGICAS  
MESTRADO EM INOVAÇÕES TECNOLÓGICAS

MARCOS ROGÉRIO FERREIRA

**COLETA SELETIVA E EDUCAÇÃO AMBIENTAL:  
DESENVOLVIMENTO E UTILIZAÇÃO DE UM APLICATIVO  
MÓVEL NA IMPLANTAÇÃO DA POLÍTICA NACIONAL DE  
RESÍDUOS SÓLIDOS EM UM MUNICÍPIO**

CAMPO MOURÃO  
2020

MARCOS ROGÉRIO FERREIRA

**COLETA SELETIVA E EDUCAÇÃO AMBIENTAL:  
DESENVOLVIMENTO E UTILIZAÇÃO DE UM APLICATIVO  
MÓVEL NA IMPLANTAÇÃO DA POLÍTICA NACIONAL DE  
RESÍDUOS SÓLIDOS EM UM MUNICÍPIO**

Dissertação apresentada ao Programa de Pós-graduação em Inovações Tecnológicas da Universidade Tecnológica Federal do Paraná, como parte das exigências para a obtenção do título de Mestre em Inovações Tecnológicas.

Orientador: Prof. Dr. Gilson Junior Schiavon

CAMPO MOURÃO  
2020

---

### Dados Internacionais de Catalogação na Publicação

---

Ferreira, Marcos Rogério

Coleta seletiva e educação ambiental: desenvolvimento e utilização de um aplicativo móvel na implantação da política nacional de resíduos sólidos em um município [recurso eletrônico] / Marcos Rogério Ferreira. – Campo Mourão, 2020.

1 arquivo eletrônico (64 f.) : PDF ; 2,7 MB

Modo de acesso: World Wide Web

Orientador: Gilson Junior Schiavon

Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Inovações Tecnológicas, Campo Mourão, 2020.

Inclui bibliografia: f. 63-64

1. Gestão integrada de resíduos sólidos. 2. Coleta seletiva de lixo. 3. Sustentabilidade e meio ambiente. 4. Inovações Tecnológicas – Dissertações. I. Schiavon, Gilson Junior, orient. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Inovações Tecnológicas. III. Título.

CDD(22.ed.) 658.514

---

### Biblioteca da UTFPR - Câmpus Campo Mourão

Bibliotecária/Documentalista:

Andréia Del Conte de Paiva – CRB-9/1525



Ministério da Educação  
**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
 DIRETORIA DE PESQUISA E POS-GRADUACAO-CM  
 PROG POS GRAD EM INOVACOES TECNOLOGICAS



## TERMO DE APROVAÇÃO DE DISSERTAÇÃO Nº 02

A Dissertação de Mestrado intitulada **COLETA SELETIVA E EDUCAÇÃO AMBIENTAL: DESENVOLVIMENTO E UTILIZAÇÃO DE UM APLICATIVO MÓVEL NA IMPLANTAÇÃO DA POLÍTICA NACIONAL DE RESÍDUOS SÓLIDOS EM UM MUNICÍPIO**, defendida em sessão pública pelo candidato **Marcos Rogério Ferreira**, no dia 25 de junho de 2020, foi julgada para a obtenção do título de Mestre em **Inovações Tecnológicas**, área de concentração **Inovações Tecnológicas**, linha de pesquisa **Desenvolvimento de Equipamentos, Tecnologias e Sistemas Eletrônicos**, e aprovada em sua forma final, pelo Programa de Pós-Graduação em **Inovações Tecnológicas**.

BANCA EXAMINADORA:

Prof. Dr. Gilson Junior Schiavon - UTFPR	
Prof. Dr. Eduardo Giometti Bertogna - UTFPR	
Prof. Dr. Glaucio Pedro de Alcantara - UEM	“Participação por videoconferência”

A via original deste documento encontra-se arquivada na Secretaria do Programa, contendo a assinatura da Coordenação após a entrega da versão corrigida do trabalho.

Assinado eletronicamente por

Coordenador do Programa  
**Prof. Dr. Wyrllen Everson de Souza**



Documento assinado eletronicamente por **GILSON JUNIOR SCHIAVON, PROFESSOR(A) ORIENTADOR(A)**, em 27/06/2020, às 18:11, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **EDUARDO GIOMETTI BERTOGNA, PROFESSOR ENS BASICO TECNOLÓGICO**, em 27/06/2020, às 20:25, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [https://sei.utfpr.edu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.utfpr.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **1497809** e o código CRC **B4A74659**.

## RESUMO

Com a promulgação da Lei 12.305/2010 que instituiu a Política Nacional de Resíduos Sólidos, estão sujeitos a observação desta lei todos os responsáveis, direta ou indiretamente, pela geração de resíduos sólidos a desenvolverem ações relacionadas à gestão integrada ou ao gerenciamento de resíduos sólidos. Nesse contexto, a coleta seletiva se apresenta como uma ferramenta fundamental para o cumprimento da legislação vigente. Esta pesquisa analisa as principais leis e decretos que deverão nortear os gestores públicos municipais para a criação dos Planos Municipais de Gestão Integrada de Resíduos Sólidos. Este trabalho apresenta um protótipo, para a criação de um aplicativo móvel, que poderá ser implementado por gestores públicos e ser utilizado como um canal de comunicação entre a prefeitura e seus munícipes. Se o planejamento tem como base a coleta seletiva, os principais atores nesse cenário são os cidadãos, então criar uma tecnologia que facilite a comunicação apresentando uma agenda de coleta seletiva, pontos de coleta especializada e um espaço para promover a educação ambiental deve ser prioridade.

**Palavras-chave:** Sustentabilidade. Coleta Seletiva. Tecnologias. Aplicativos Móveis.

## **ABSTRACT**

With the enactment of Law 12.305/2010 that instituted the National Solid Waste Policy. And all those responsible, directly or indirectly, for the generation of solid waste to carry out actions related to integrated management or solid waste management are subject to the observation of this law. In this context, selective collection is a fundamental tool for compliance with current legislation. This research analyzes the main laws and decrees that should guide the municipal public managers for the creation of the Municipal Plans for Integrated Management of Solid Waste. This work presents a prototype, for the creation of a mobile application, which can be implemented by public managers and be used as a communication channel between the city and its residents. If planning is based on selective collection, the main actors in this scenario are the citizens, so creating a technology that facilitates communication by presenting a selective collection agenda, specialized collection points and a space to promote environmental education should be a priority.

**Keywords:** Sustainability. Garbage Collection. Technology. Mobile App.

## LISTA DE FIGURAS

Figura 1 – Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal. . . . .	15
Figura 2 – Existência de telefones nos domicílios. . . . .	15
Figura 3 – Ambiente de Trabalho da cooperativa Recicla Naviraí . . . . .	18
Figura 4 – Descarga de recicláveis . . . . .	19
Figura 5 – Esteira de transporte . . . . .	19
Figura 6 – Separação de materiais . . . . .	19
Figura 7 – Prensa de materiais . . . . .	19
Figura 8 – Depósito de materiais . . . . .	19
Figura 9 – Tela de Cadastro - Coleta Especializada . . . . .	34
Figura 10 – Tela de Consulta - Coleta Especializada . . . . .	40
Figura 11 – Tela de consulta por tipo de Coleta . . . . .	45
Figura 12 – Lista de Locais . . . . .	48
Figura 13 – Local selecionado . . . . .	48
Figura 14 – Diagrama de Casos de Uso . . . . .	52
Figura 15 – Diagrama de Classes . . . . .	53
Figura 16 – Mapa do aplicativo . . . . .	54
Figura 17 – Desenvolvimento em Camadas . . . . .	55
Figura 18 – Estrutura - Spring Boot . . . . .	56
Figura 19 – Estrutura - Ionic . . . . .	57
Figura 20 – Tela principal do ambiente administrativo. . . . .	57
Figura 21 – Tela de cadastro. . . . .	58
Figura 22 – Tela de consulta e edição. . . . .	58
Figura 23 – Tela principal e de cadastro de usuários. . . . .	60
Figura 24 – Telas de agenda, pontos de coleta e educação ambiental . . . . .	60

## LISTA DE QUADROS

Quadro 1 – Arquivo application.properties comentado. . . . .	27
Quadro 2 – Anotação da camada de persistência. . . . .	30
Quadro 3 – Anotação da camada de serviço. . . . .	32



## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
DAO	Data Access Object
POM	Project Object Model
UML	Unified Modeling Language
STS	Spring Tool Suite
IDE	Integrated Development Environment
SGBD	Sistema de Gerenciamento de Banco de Dados
JDBC	Java Database Connectivity
JPA	Java Persistence API
ORM	Object-relational mapping
CONAMA	Conselho Nacional de Meio Ambiente

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
1.1	Objetivo Geral	13
1.2	Objetivos Específicos	14
1.3	Justificativa	14
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
2.1	Cooperativa de Reciclagem	17
2.2	Tecnologias utilizadas no desenvolvimento do sistema	20
2.2.1	Modelagem do Sistema e diagramas	20
2.2.2	Ambientes de Desenvolvimento	20
2.2.3	Frameworks e tecnologias	21
2.3	Apresentação da metodologia de implementação	22
2.3.1	Documentação de Referência do Spring Framework	22
2.3.2	Classe Abstrata	23
2.3.3	Classe Coleta Especializada	24
2.3.4	Camada de Persistência	26
2.3.5	Camada de Serviço	30
2.3.6	Camada de Controle	32
2.3.7	Camada de visualização	34
2.3.8	Arquitetura REST	43
2.3.9	Desenvolvimento do aplicativo móvel com Ionic	43
2.4	Geolocalização utilizando Google Maps	51
2.5	Sistema de armazenamento de imagens	51
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>52</b>
3.1	Estrutura e organização do Sistema	52
3.1.1	Diagrama de Casos de Uso	52
3.1.2	Diagrama de Classes	53
3.2	Estrutura do Aplicativo	54
3.3	Arquitetura do Software	55
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>56</b>
4.1	Sobre a Construção do protótipo	56
4.1.1	Estrutura e organização do código	56
4.1.2	Ambiente Administrativo	57
4.1.3	Uso do Aplicativo	59

4.1.4	Principais funções do Aplicativo . . . . .	59
4.1.5	Implantação e utilização do Aplicativo . . . . .	60
4.1.6	Publicações . . . . .	61
4.1.7	Registro de Software . . . . .	61
5	CONCLUSÃO . . . . .	62
	REFERÊNCIAS . . . . .	63

## 1 INTRODUÇÃO

Com a promulgação da Constituição Federal de 1988, os municípios passaram a ter autonomia e foram reconhecidos como entes da federação, com isso obtiveram poderes e independência para criar e organizar sua estrutura administrativa e o seu sistema legislativo, como está disposto no artigo 30 da Constituição Federal (BRASIL, 1988). E foi interpretando o inciso V deste mesmo artigo que os gestores municipais assumiram também a titularidade da limpeza pública e a gestão dos resíduos sólidos.

Mesmo com as diretrizes sobre reciclagem, sustentabilidade e educação ambiental, apresentadas pelo Conselho Nacional de Meio Ambiente - CONAMA, ainda não havia uma legislação federal específica sobre gerenciamento de resíduos sólidos no Brasil. Iniciou-se então um profundo debate sobre o assunto em nível nacional que resultou na Lei 12.305 que institui a Política Nacional de Resíduos Sólidos e “que dispõe sobre os princípios, objetivos e instrumentos, bem como sobre as diretrizes relativas à gestão integrada e ao gerenciamento de resíduos sólidos, incluídos os perigosos” (BRASIL, 2010).

Art. 1º, § 1º Estão sujeitas à observância desta Lei as pessoas físicas ou jurídicas, de direito público ou privado, responsáveis, direta ou indiretamente, pela geração de resíduos sólidos e as que desenvolvam ações relacionadas à gestão integrada ou ao gerenciamento de resíduos sólidos (BRASIL, 2010).

No mesmo ano de 2010, foi apresentado o Decreto nº 7.404, que regulamenta a Lei nº 12.305, “Art. 1º - Este Decreto estabelece normas para execução da Política Nacional de Resíduos Sólidos” (BRASIL, 2010), que institui a Política Nacional de Resíduos Sólidos, cria o Comitê Interministerial da Política Nacional de Resíduos Sólidos e o Comitê Orientador para a Implantação dos Sistemas de Logística Reversa, e dá outras providências.

O Decreto apresenta em seu texto, de forma taxativa, as responsabilidades dos geradores de resíduos sólidos e do poder público, bem como as diretrizes para a aplicação da coleta seletiva em um município. O legislador também documenta como deve ser estabelecida a logística reversa, que para as cooperativas de catadores é a principal ferramenta que movimenta os materiais em estoque e gera renda.

Art. 13. A logística reversa é o instrumento de desenvolvimento econômico e social caracterizado pelo conjunto de ações, procedimentos e meios destinados a viabilizar a coleta e a restituição dos resíduos sólidos ao setor empresarial, para reaproveitamento, em seu ciclo ou em outros ciclos produtivos, ou outra destinação final ambientalmente adequada. (BRASIL, 2010)

A Lei 12.305 e o Decreto 7.404 apresentam a forma como os gestores organizarão suas políticas públicas quanto à aplicação da legislação sobre resíduos sólidos. Trata-se dos “Planos de Resíduos Sólidos”. O primeiro apresentado foi o Plano Nacional de Resíduos Sólidos, elaborado pelo Ministério do Meio Ambiente. A Lei também obriga os estados e municípios a criarem seus Planos de Resíduos Sólidos. Para municípios menores, é autorizada a criação de um plano integrado micro regional em que os entes participantes podem compartilhar as soluções desenvolvidas. Os planos devem orientar a gestão no processo de implantação de

técnicas e tecnologias relacionadas sustentabilidade e coleta seletiva. É de responsabilidade dos municípios elaborar o Plano de Gerenciamento de Resíduos Sólidos, que deverá ser aplicado junto as empresas públicas e privadas.

Com os planos estabelecidos nos estados e municípios inicia o processo de criação das Políticas de Resíduos Sólidos, que no município será nomeada por "Política Municipal de Gestão Integrada de Resíduos Sólidos", que regulamenta e detalha mais especificamente a aplicação do Plano de Resíduo Sólidos, incluindo sanções aos que desobedecerem às normas impostas.

Uma vez organizada as ações de sustentabilidade em um município, todos os geradores de resíduos sólidos, sejam pessoas físicas ou jurídicas, passam a ter responsabilidade na participação dos programas implantados, sujeitos a sanções previstas em lei. E nesse processo a Lei nº 9.795/1999 Política Nacional de Educação Ambiental ganha protagonismo, "Como parte do processo educativo mais amplo, todos têm direito à educação ambiental", que auxiliará no processo de educação ambiental para promover a sustentabilidade e o engajamento de todos nos programas e ações estabelecidos nos planos municipais de resíduos sólidos.

Art. 1º Entendem-se por educação ambiental os processos por meio dos quais o indivíduo e a coletividade constroem valores sociais, conhecimentos, habilidades, atitudes e competências voltadas para a conservação do meio ambiente, bem de uso comum do povo, essencial à sadia qualidade de vida e sua sustentabilidade. Art. 2º A educação ambiental é um componente essencial e permanente da educação nacional, devendo estar presente, de forma articulada, em todos os níveis e modalidades do processo educativo, em caráter formal e não-formal (BRASIL, 1999).

Na atual conjuntura, a sustentabilidade não se restringe somente mais a uma palavra bonita ou a uma bandeira que se levanta apenas por modismo casual. É um modo de vida que deve ser adotado por todos, pois é inaceitável, por exemplo, que o lixo urbano seja ainda negligenciado por cidadãos. Que a falta de educação ambiental seja responsável por pequenos atos que poderiam, em muito, auxiliar na sustentabilidade, como a própria separação do lixo doméstico e a participação ativa da coleta seletiva de resíduos sólidos. É preciso que o poder público auxilie as cooperativas de catadores que têm um papel fundamental na reciclagem, pois além de movimentar a economia do setor e aliviar a atividade de coleta realizada pela prefeitura, encaminham materiais recicláveis para que possam retornar e ser reutilizados.

Não pode restringir somente em realizar eventos esporádicos sobre sustentabilidade, coleta seletiva e proteção ao meio ambiente, é preciso criar uma cultura, um estilo de vida mais sustentável. Quando o cidadão coloca seu lixo ele deve saber se a coleta será de lixo úmido ou seco, e também se é o dia correto de coleta. Lixo depositado para a coleta em dias errados podem ser extraviados por animais soltos como cães e gatos. "Os cães, pelo fato de muitos destes serem abandonados e buscarem no lixo o único alimento disponível, espalham resíduos na rua"(VIEIRA, 2012, P. 85). E com isso o lixo pode ser espalhado pela cidade causando outros problemas que afetarão ainda mais a população, a manutenção e a limpeza.

O lixo espalhado nas ruas pode entupir esgotos e causar inundações em tempos chuvosos, gerando mais serviço para as secretarias municipais de infraestrutura. E quando levado pelo vento, como as sacolas plásticas, podem acumular água de chuva e contribuir para

a proliferação de mosquitos, dentre os quais o *aedes aegypti*.

A coleta de lixo urbana é uma das principais medidas sanitárias que ajuda a prevenir a proliferação de doenças em zonas urbanas e rurais. A presença humana em uma região é capaz de produzir um alto sortimento e volume de resíduos, que precisam ser coletados e devidamente tratados (SOBRAL, 2019, P. 1076).

A falta de comunicação entre prefeituras e municípios sobre a importância da coleta seletiva, separação de resíduos sólidos e educação ambiental é evidente quando se tem um número elevado de materiais recicláveis sendo encaminhados para os aterros sanitários, forçando os catadores a trabalharem em condições insalubres e perigosas. Ter acesso a agenda de coleta de lixo pode afetar significativamente a manutenção em várias áreas de uma cidade.

Na busca por ferramentas que auxiliem as prefeituras e cooperativas de catadores no processo educativo, divulgação e integração nas ações sustentáveis, encontramos nas tecnologias de aplicativos para dispositivos móveis uma oportunidade de inovar também neste segmento.

Dos aplicativos pesquisados e disponíveis nas plataformas para *download* encontramos três principais soluções, a primeira com foco na comunidade de catadores e cooperativas, o que é a base para a coleta seletiva hoje no Brasil. Ela conecta os catadores aos municípios que podem cadastrar no aplicativo seus recicláveis, desta forma é gerado um mapeamento e o catador poderá realizar a retirada junto a residência.

Outros modelos que encontramos foram os aplicativos de mapeamento dos Pontos de Coleta Voluntária (PEV). É possível cadastrar todos pontos espalhados na cidade e o usuário do aplicativo poderá conhecer e escolher o ponto mais próximo de sua residência ou por tipo de material coletado. E também os aplicativos dedicados as matérias educacionais, com foco exclusivo em educação ambiental.

Porém a responsabilidade legal de implantar a política nacional de resíduos sólidos, foi atribuída aos municípios e por esse motivo o foco deste projeto serão às Leis e Decretos que regulamentam o tema, bem como o Plano Municipal de Resíduos Sólidos. Para criar um aplicativo que possa realmente auxiliar no planejamento estratégico, criando condições para que o município participe ativamente nas ações implementadas e cumpra exatamente o que foi previsto na legislação vigente. Durante o desenvolvimento do projeto, não encontramos um aplicativo com o mesmo domínio de problema.

## 1.1 Objetivo Geral

A criação de um aplicativo para dispositivos móveis que promova a sustentabilidade, coleta seletiva e educação ambiental, é o objetivo geral deste trabalho. Apresentar à população os programas e ações desenvolvidas pelo município e cooperativa e promover a participação de todos os cidadãos no processo de separação de recicláveis bem como orientar para um estilo de vida mais sustentável.

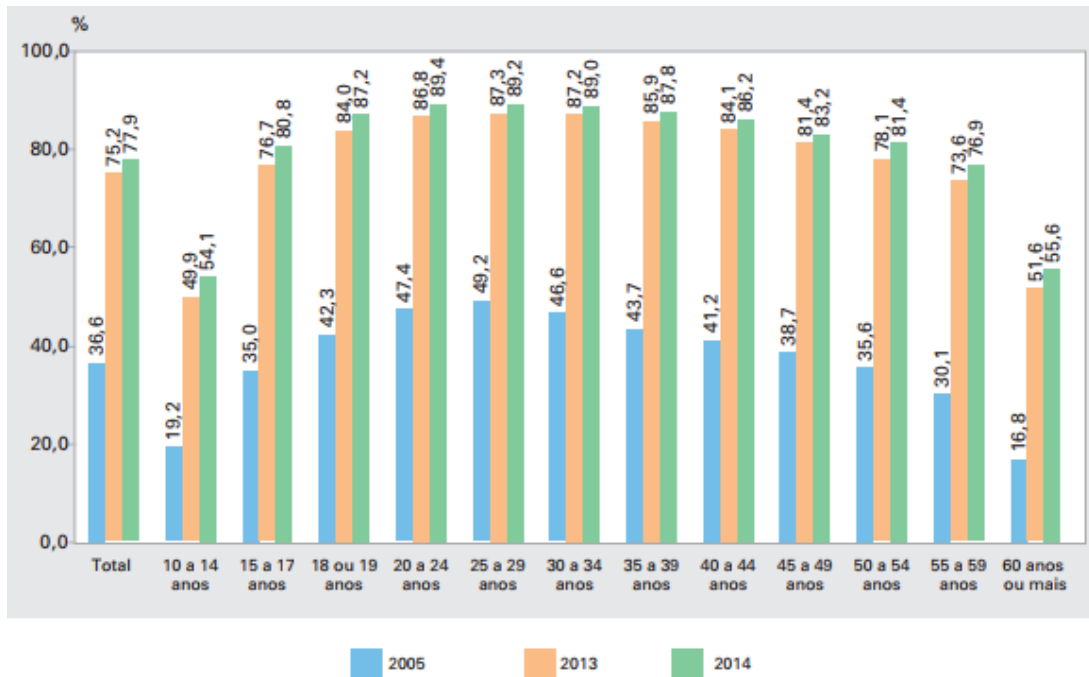
## 1.2 Objetivos Específicos

- O aplicativo manterá uma agenda de horários de coleta de lixo seco, úmido e entulho por bairro.
- O município deverá realizar seu cadastro para receber os horários atualizados ou mudanças nos horários por falta de caminhão ou pessoal, por exemplo, evitando assim que o lixo fique por muito tempo exposto na rua e que sejam violados por animais.
- Será apresentado periodicamente matérias sobre educação ambiental, atividades desenvolvidas com foco em sustentabilidade, técnicas para o desenvolvimento e cultivo de hortas urbanas dentre outros assuntos pertinentes.
- Estarão presentes no aplicativo os programas e ações promovidos pela prefeitura, para que o alcance seja o mais amplo possível, atividades como coleta de entulhos na campanha contra a dengue, bem como agenda de vacinações, dentre outras ações propostas.
- É importante que a população saiba onde depositar materiais que não serão coletados pela cooperativa e que não podem ser encaminhados para o aterro, materiais como pilhas, baterias, lâmpadas fluorescentes e óleo de cozinha. Portanto o aplicativo manterá um mapa com os locais de coleta especializada para esse fim.

## 1.3 Justificativa

Sem o apoio da população toda estratégia relacionada à coleta seletiva e sustentabilidade será fadada ao fracasso. Deste modo, esta dissertação tem por objetivo apresentar uma proposta tecnológica que possa ser implantada pelas prefeituras para se aproximar dos municípios e tratar diretamente sobre os assuntos relacionados às questões ambientais, pois as famílias estão consumindo mais tecnologias como Smartphones e estão mais conectadas à internet (IBGE, 2016), conforme pode ser constatado na Figure 1.

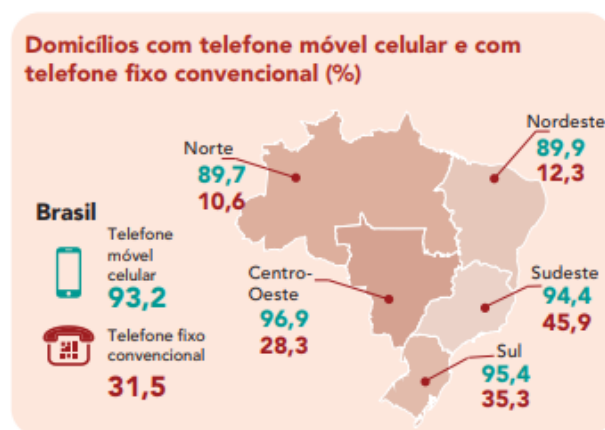
Figura 1 – Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal.



Fonte: IBGE (2016)

Em 2016, já havia telefone móvel celular em 92,6% dos domicílios do país e, em 2017, o percentual aumentou para 93,2%, como mostra a Figure 2. Com relação à parcela dos domicílios em que existia somente telefone fixo convencional, esta passou de 2,0% em 2016 para 1,7% em 2017 (IBGE, 2018).

Figura 2 – Existência de telefones nos domicílios.



Fonte: IBGE (2018)

A criação de um aplicativo para celular se apresenta como uma ferramenta poderosa para educar a população para uma vida mais sustentável, o que é uma obrigatoriedade como está disposto em lei, ou seja, é necessário que se tenha “[...] programas e ações de educação



ambiental que promovam a não geração, a redução, a reutilização, a coleta seletiva e a reciclagem de resíduos sólidos [...]” (BRASIL, 2010).

A proposta deste aplicativo também seguirá o Programa Reciclo, desenvolvido pelo Instituto das Águas do Paraná, que faz um convênio com as prefeituras para a reestruturação e aquisição de infraestrutura destinada à implantação de coleta seletiva e reciclagem como também a implantação de cooperativas de catadores.

O Sistema Integrado de Coleta Seletiva, desenvolvido pelo Instituto das Águas do Paraná, além de promover a preservação do meio ambiente e se preocupar com a saúde pública, é inovador no que tange a sustentabilidade dos recicladores e conscientização da população através da educação ambiental (AGUASPARANA, 2018).

Esta medida poderia impactar significativamente na economia daqueles que trabalham com reciclagem gerando uma maior renda, pois muito material reciclável deixaria de ir para o aterro sanitário, o que aumentaria o número de empregados nesse segmento e a vida útil do aterro.

## 2 FUNDAMENTAÇÃO TEÓRICA

No planejamento e desenvolvimento de software uma das etapas mais importante é o detalhamento do problema que será abordado, e nesse processo compreender as reais necessidades a serem implementadas para que as soluções desenvolvidas correspondam exatamente as funcionalidades desejadas. Nesta seção apresentaremos a cooperativa de reciclagem que foi o ambiente estudado para a construção do sistema e onde coletamos as principais informações e dificuldades em realizar a coleta seletiva e como um aplicativo pode auxiliar. Como o objetivo principal desta dissertação é disponibilizar um projeto de software que possa ser continuado por outras equipes de desenvolvimento, registramos também todas as tecnologias utilizadas na construção do sistema, para facilitar o processo de criar um ambiente de desenvolvimento para dar continuidade ao projeto.

O desenvolvimento do software foi separado por módulos, para facilitar a adição de novas funcionalidades sem prejudicar as que já existem, portanto apresentamos a funcionalidade do requisito, Pontos de Coleta Especializada, documentando a implementação do início ao fim, para que o desenvolvedor compreenda os passos seguidos e possa implementar outros módulos partindo da metodologia apresentada.

### 2.1 Cooperativa de Reciclagem

Para que um município possa desenvolver estratégias relacionadas a coleta seletiva, deverá criar ou organizar, se existir na cidade, uma cooperativa de reciclagem, que concentrará todas as ações e planejamento da coleta, separação e estoque. Bem como manter contratos com empresas que farão a retirada dos materiais, desenvolvendo assim a logística reversa e desenvolvimento da economia local. A lei nº 12.305 orienta a criação de instrumento para a implantação da coleta seletiva e aplicação da logística reversa.

Art. 8º São instrumentos da Política Nacional de Resíduos Sólidos, entre outros:

III - a coleta seletiva, os sistemas de logística reversa e outras ferramentas relacionadas à implementação da responsabilidade compartilhada pelo ciclo de vida dos produtos;

IV - o incentivo à criação e ao desenvolvimento de cooperativas ou de outras formas de associação de catadores de materiais reutilizáveis e recicláveis;

VII - a pesquisa científica e tecnológica;

VIII - a educação ambiental; (BRASIL, 2010)

A indústria que comercializa seus produtos em embalagens recicláveis deve viabilizar a recuperação deste material de maneira proporcional à quantidade colocada anualmente no mercado, por meio da estruturação de sistemas de logística reversa de embalagens. A lei federal nº 12.305 que obriga a aplicação da logística reversa e garante que as ações realizadas

pelas cooperativa terão êxito, pois com a obrigatoriedade, a cooperativa passa a realizar um papel facilitador, onde a indústria poderá em um único local, coletar os materiais e manter os trabalhos da cooperativa.

Art. 3º XII - logística reversa: instrumento de desenvolvimento econômico e social caracterizado por um conjunto de ações, procedimentos e meios destinados a viabilizar a coleta e a restituição dos resíduos sólidos ao setor empresarial, para reaproveitamento, em seu ciclo ou em outros ciclos produtivos, ou outra destinação final ambientalmente adequada; (BRASIL, 2010)

Para compreender as reais necessidades da cooperativa, e levantar os requisitos do aplicativo que auxiliará no processo, visitamos a cooperativa "Recicla Naviraí", uma cooperativa sediada em Naviraí, Mato Grosso do Sul e que está se tornando um modelo no estado, tanto pelas máquinas modernas adquiridas, como o amplo espaço de trabalho como mostra a Figure 3 e a logística de atendimento na coleta seletiva municipal, bem como gerar fonte de renda e fomentar a economia entre trabalhadores que antes viviam na informalidade.

Figura 3 – Ambiente de Trabalho da cooperativa Recicla Naviraí



Fonte: Recicla Naviraí (2019)

Nas imagens a seguir é possível ver algumas áreas em operação da cooperativa de Reciclagem, com os setores de separação de resíduos (Figure 4) o que é reciclável e o que lixo comum e que deverá ser descartado no aterro. Na (Figure 5) temos a esteira de recicláveis que transporta todo material coletado e que depositará nas esteiras de separação (Figure 6), onde os trabalhadores irão separar os materiais por tipo, depositando em *bags* (Tipo de contêiner não rígido) que será pesado e registrado por cooperado.

Figura 4 – Descarga de recicláveis  
 Figura 5 – Esteira de transporte  
 Figura 6 – Separação de materiais



Fonte: Autoria própria (2020)    Fonte: Autoria própria (2020)    Fonte: Autoria própria (2020)

Os materiais depois de pesados e contabilizados no registro de trabalho do cooperado será transportado para o setor de prensa (Figure 7). Na sequência o lote prensado vai para o depósito (Figure 8) que será transportado para a indústria responsável pela reciclagem daquele tipo de material.

Figura 7 – Prensa de materiais



Fonte: Autoria própria (2020)

Figura 8 – Depósito de materiais



Fonte: Autoria própria (2020)

Mesmo com local adequado, máquinas modernas, trabalhadores treinados e contratos que garantem o fluxo dos materiais, a cooperativa precisa do apoio da população para garantir uma coleta seletiva de qualidade, é importante que os cidadãos separem os recicláveis e coloquem nos dias corretos para a coleta. Para a prefeitura é importante que o mínimo ou nenhum material reciclável seja encaminhado para o aterro, portanto garantir uma comunicação com os munícipes que são os principais geradores e promover uma política de separação de

materiais e educação ambiental será de fundamental importância para garantir o sucesso da cooperativa na implantação da coleta seletiva municipal.

## 2.2 Tecnologias utilizadas no desenvolvimento do sistema

Para este trabalho foram utilizadas várias tecnologias de desenvolvimento de software, houve um cuidado nas escolhas para garantir que outra equipe de programadores possam dar continuidade ao projeto sem ter custo de licenças e uma curva de aprendizado muito alta. Para tanto o aplicativo foi desenvolvido utilizando apenas ferramentas gratuitas e ou *open source*.

### 2.2.1 Modelagem do Sistema e diagramas

Neste projeto utilizamos a plataforma *online* do visual paradigm, que nos permite modelar o sistema, criar os diagramas, dentre outros artefatos de software que nos proporcionará melhor entendimento das funcionalidades e como o aplicativo será construído, portanto todos diagramas apresentados neste projeto foi desenvolvido utilizando esta ferramenta. Esta ferramenta está disponível na web bastando para tanto acessar o seguinte endereço <https://online.visual-paradigm.com/>.

### 2.2.2 Ambientes de Desenvolvimento

#### 2.2.2.1 STS – Spring Tool Suite

É um ambiente de desenvolvimento baseado no Eclipse e que foi customizado para o desenvolvimento de aplicativos utilizando o Spring. Ele fornece todas as ferramentas necessárias para implementar, depurar, executar e implantar projetos utilizando o Framework Spring, incluindo integrações com GitHub para controle de versionamento e o uso do Maven responsável por gerenciar as dependências do projeto, ou seja, basta registrar no *Project Object Model* (POM) as bibliotecas, frameworks dentre outros recursos que deseja utilizar no projeto. Está IDE está disponível para download no sitio <https://spring.io/tools>, que mantém o String.

#### 2.2.2.2 Microsoft Visual Studio

O aplicativo foi desenvolvido utilizando o Visual Studio Code, IDE criada e mantida pela Microsoft, que integra todas as ferramentas necessárias para o desenvolvimento de sistemas baseado em Web. Para o desenvolvimento do aplicativo utilizando o Ionic Framework, obtivemos uma ótima integração, facilidade na implementação, depuração e testes. Além das possibilidades de associar plugins e outras ferramentas à IDE, o desenvolvedor poderá obter o software no sitio <https://visualstudio.microsoft.com/pt-br/>.

## 2.2.3 Frameworks e tecnologias

### 2.2.3.1 Spring Boot

O Spring Boot facilita a criação de aplicativos autônomos baseados em Spring e com grau de produção que você pode "simplesmente executar"(SPRING, 2020). Utilizando o Maven com gerenciador de dependências basta adicionar os módulos que deseja implementar no projeto, tais como Web, Templates, Segurança dentre outros. A facilidade de controlar a injeção de dependência e inversão de controle torna este framework tão acessível e utilizado no mercado atualmente. Disponível em <https://spring.io/projects/spring-boot>.

### 2.2.3.2 Hibernate

O framework Hibernate permite aos desenvolvedores criar software utilizando o paradigma Orientado a Objetos e integrar com banco de dados relacionais. Com o hibernate é possível realizar o ORM, mapeamento objeto relacional, utilizando o *Java Database Connectivity* (JDBC) e o *Java Persistence API* (JPA) para anotar as classes que serão mapeadas para o paradigma relacional. "mapeamento objeto relacional"(HIBERNATE, 2020).

O Hibernate é um framework que se relaciona com o banco de dados, onde esse relacionamento é conhecido como mapeamento objeto/relacional para Java, deixando o desenvolvedor livre para se concentrar em problemas da lógica do negócio (GONÇALVES, 2007).

O projeto na verdade utiliza JPA, Java Persistence API, que é uma especificação para mapeamento objeto relacional e utiliza a implementação do Hibernate para isso, porém como todas as implementações seguem a especificação JPA e possível alterar o framework de persistencia de dados sem prejuízo ao projeto. O hibernate pode ser encontrado em seu sitio oficial <https://hibernate.org/orm/>.

### 2.2.3.3 Banco de Dados

O banco de dados escolhido foi o PostgreSQL, mesmo sendo um Sistema de Gerenciamento de Banco de Dados (SGBD) gratuito, possui maturidade sendo aceito pelo mercado como um banco de dados robusto e seguro. Sabendo que este projeto pode ser escalável, a escolha desta tecnologia, nos garante que o projeto pode ser manutenível sem prejuízo com a persistência de dados. O projeto pode ser localizado em <https://www.postgresql.org/>.

### 2.2.3.4 Thymeleaf - Sistema Administrativo

O Thymeleaf não foi desenvolvido pelo Spring, porém é uma biblioteca que facilita a criação das páginas web do projeto atuando na camada de visualização. É um template utilizado em para aplicações em JavaWeb onde as requisições e instruções são executadas todas do lado do servidor, combinando com outras bibliotecas como Bootstrap ele torna fácil a tarefa de prototipar páginas web e realizar testes e entregas em pouco tempo, aumentando a produtividade no desenvolvimento. Disponível em <http://www.thymeleaf.org/>.

### 2.2.3.5 Ionic - Sistema Mobile

O Ionic é um framework open source, que tem por objetivo a criação de aplicações híbridas para os sistema mobile. Com o Ionic é possível criar aplicativos multiplataforma, ou seja podemos criar um app que possa ser executado no Android, iPhone e Windows Mobile, sem necessidade de recompilação da SDK.

O Ionic facilita o processo de criação inicial do desenvolvimento, passando pelos testes em browser, sem a necessidade de instalar o aplicativo em um dispositivo. Isso exceto em casos extremos, como a utilização de APIs para os recursos da plataforma em questão (GOIS, 2017).

Para acesso aos recursos nativos, o Ionic utiliza o Cordova que através de um conjunto de plugins faz a ponte entre o aplicativo e os recursos do celular, "Esses plugins principais fornecem ao seu aplicativo o acesso a recursos do dispositivo, como bateria, câmera, contatos, etc."(CORDOVA, 2015).

O Ionic Framework é construído com a simplicidade em mente, para que a criação de aplicativos Ionic seja agradável, fácil de aprender e acessível a praticamente qualquer pessoa com habilidades de desenvolvimento da Web (IONIC, 2020).

O Ionic framework possui uma documentação completa, o que facilita o trabalho do desenvolvedor no momento de criar ou manter recursos desenvolvidos no aplicativo. A documentação e novas versões do framework pode ser acessados no sitio <https://ionicframework.com/>.

## 2.3 Apresentação da metodologia de implementação

Para exemplificar melhor a estrutura do sistema apresentaremos a resolução de um requisito, Pontos de Coletas Especializadas, como foi modelado no projeto, seu acesso pelo ambiente administrativo e sua implementação no aplicativo. Como todo o sistema foi definido com a mesma metodologia, os outros requisitos seguirão o mesmo modelo.

### 2.3.1 Documentação de Referência do Spring Framework

Podemos acessar o site <https://spring.io/>, escolher o menu "Project", clicar na categoria "Spring Framework". Na página que se abre, clique em "Learn" e escolha "Reference Doc" vai abrir a documentação do Spring Framework. Para compreender melhor o assunto pesquise sobre "Using Spring Boot" no item "1. Build Systems" neste trabalho utilizamos o subitem "1.5. Starters".

#### 2.3.1.1 Dependência utilizada no POM

```

1 <dependency>
2     <groupId>org.springframework.boot</groupId>
3     <artifactId>spring-boot-starter-data-jpa</artifactId>
4 </dependency>
```

## 2.3.2 Classe Abstrata

Utilizando o Spring-Boot e Hibernate normalmente padronizamos a construção das classes de domínio utilizando os atributos "ID" que serão utilizados como identificadores, e também métodos específicos como "equals" e "hashCode" que auxiliarão no processo de gerenciamento das instâncias das classes. Então uma maneira de reutilizar esses métodos é declarar uma super classe disponibilizará essa implementação para todas as outras classes do domínio.

### 2.3.2.1 Anotações utilizadas na classe

- **ID extends Serializable** - Passa a responsabilidade para as classes filhas a definição do tipo do ID
- **@Id** - Mapear o atributo chave que será a PK na tabela, este atributo é obrigatório.
- **@GeneratedValue** - Informa como a chave (ID) será gerada no banco de dados implements Serializable Indica que o tipo (Classe) é serializável;
- **@MappedSuperclass** - Essa anotação indica que durante o mapeamento não teremos a tabela gerada no banco de dados, somente suas filhas
- **@SuppressWarnings("serial")** - Retira o Warning de criação de ID, pois passamos essa responsabilidade para a JRE.

### 2.3.2.2 Classe abstrata implementada

```

1 package com.recycle.web.domain;
2
3 //Imports - Omitidos
4
5 @MappedSuperclass
6 public abstract class AbstractEntity <ID extends Serializable>
7     implements Serializable {
8
9     @Id @GeneratedValue(strategy = GenerationType.IDENTITY)
10    private ID id;
11
12    public ID getId() {
13        return id;
14    }
15
16    public void setId(ID id) {
17        this.id = id;
18    }
19
20    @Override

```



```

20     public int hashCode() {
21         final int prime = 31;
22         int result = 1;
23         result = prime * result + ((id == null) ? 0 : id.hashCode());
24         return result;
25     }
26
27     @Override
28     public boolean equals(Object obj) {
29         if (this == obj)
30             return true;
31         if (obj == null)
32             return false;
33         if (getClass() != obj.getClass())
34             return false;
35         AbstractEntity<?> other = (AbstractEntity<?>) obj;
36         if (id == null) {
37             if (other.id != null)
38                 return false;
39         } else if (!id.equals(other.id))
40             return false;
41         return true;
42     }
43
44     @Override
45     public String toString() {
46         return "ID = " + id;
47     }
48 }

```

### 2.3.3 Classe Coleta Especializada

Implementação da classe de domínio para a Coleta Especializada, **extends AbstractEntity<Long>** indica que a classe herda da super classe **abstract** e define o tipo do ID como Long. Os itens herdados serão ID, hashCode, equals, toString e Serializable e reutilizados.

#### 2.3.3.1 Anotações da JPA utilizadas

- **@Entity** - A classe será mapeada para uma tabela no banco de dados.
- **@Table(name = "coleta-especializada")** - Se a tabela tiver um nome diferente da classe, poderá ser definida.
- **@Column** - Cada atributo da classe será mapeado como atributo na tabela do banco de dados, então esse atributo é facultativo, porém se houver uma configuração específica

poderá ser declarado utilizando-o.

- **name = "tipo-coleta"**, nome da coluna no banco de dados;
  - **nullable = false**, o registro não pode ser nulo;
  - **unique = true**, o registro deve ser único;
  - **length = 60**, define a quantidade de caracteres do registro.
- **@ManyToOne** - Muitos para um, representa por exemplo que pode haver muitos LOCAIS de coleta para cada BAIRRO;
  - **@JoinColumn(name = "id-bairro-fk")** - Define a coluna responsável pelo relacionamento que no modelo relacional é representado pela FK (Foreign Key);
  - Em um relacionamento Bidirecional o outro lado do relacionamento precisa também ser mapeado.
    - **@OneToMany** - Um para muitos, representa por exemplo que pode haver um BAIRRO para vários LOCAIS de coleta;
    - **mappedBy = "bairro"** - Definimos aqui um relacionamento Bidirecional e quando isso acontece precisamos indicar qual o lado forte da relação, e o lado forte sempre é aquele que contém a FK;

### 2.3.3.2 Classe de Coleta implementada

```

1 package com.recycle.web.domain;
2
3 // Imports omitidos
4
5 @Entity
6 @Table(name = "coleta_especializada")
7 public class ColetaEspecializada extends AbstractEntity<Long> {
8     private String local;
9     @Column(name = "tipo_coleta")
10    private String tipoColeta;
11    @Column(name = "horario_atendimento")
12    private String horarioAtendimento;
13    private String contato;
14    private String logradouro;
15    private Integer numero;
16    @ManyToOne
17    @JoinColumn(name = "id_bairro_fk")
18    private Bairro bairro;
19    private String CEP;
20    private String cidade;
21    private String uf;
22

```

```

23 // Metodos Getters and Setters omitidos
24 }

```

## 2.3.4 Camada de Persistência

### 2.3.4.1 Documentação de Referência do Spring Framework

Podemos acessar o site <https://spring.io/>, escolher o menu "Project", clicar na categoria "Spring Framework". Na página que se abre, clique em "Learn" e escolha "Reference Doc" vai abrir a documentação do Spring Framework. Para compreender melhor o assunto pesquise sobre "Data Access". Como vamos utilizar o Spring Boot, clique novamente em "Project" e clique na categoria "Spring Boot", abra a documentação, selecione "Application Properties" escolha especificamente o item "5. Data Properties".

### 2.3.4.2 Configuração do arquivo application.properties

No arquivo "application" podemos configurar vários recursos que o software desempenhará, neste projeto apenas configuramos o acesso ao banco de dados e a criação do esquema baseado nas classes mapeadas com o JPA.

```

1 #DATASOURCE
2 spring.datasource.driver-class-name = org.postgresql.Driver
3 spring.datasource.url = jdbc:mysql://localhost:3306/ppgit-recicle
4 spring.datasource.username = postgres
5 spring.datasource.password = *****
6
7 #JPA
8 spring.jpa.hibernate.ddl-auto = update
9 spring.jpa.show-sql = true
10 spring.jpa.open-in-view = true
11 spring.jpa.hibernate.use-new-id-generator-mappings = false

```

### 2.3.4.3 Dependência utilizada no POM

No desenvolvimento de software utilizando tecnologias Java, para acessar um banco de dados relacional não precisamos nos preocupar com as políticas de acesso, pois cada SGBD implementa seu próprio Driver de conexão facilitando o acesso e a manipulação dos dados. Portanto precisamos registrá-lo no POM para que a aplicação faça o download e configure esta dependência em nosso projeto.

```

1 <dependency>
2     <groupId>org.postgresql</groupId>
3     <artifactId>postgresql</artifactId>
4 </dependency>

```

Quadro 1 – Arquivo application.properties comentado.

<b>Configurações do Banco de Dados</b>	
spring.datasource.url	Caminho do banco de dados
spring.datasource.username	Nome do usuário do banco
spring.datasource.password	Senha do usuário
<b>Configurações gerais da JPA</b>	
spring.jpa.hibernate.ddl-auto	Quando executar vai criar ou recriar o banco de dados baseado no mapeamento da aplicação.
spring.jpa.show-sql	Apresenta o SQL gerado
spring.jpa.open-in-view	Habilita o filtro do Spring para manter aberta uma transação até que o request esteja encerrado, para evitar exceções do tipo Lazy Exception em mapeamentos com anotações do tipo @OneToMany e @ManyToOne
spring.jpa.hibernate.use-new-id-generator-mappings	Garante que a geração de IDs seja de responsabilidade do banco como foi definido em @GeneratedValue e não do Hibernate.

Fonte: Autoria própria (2019)

#### 2.3.4.4 Classe de persistência Genérica

Cada entidade do projeto deverá ter a possibilidade de inserir, atualizar, excluir e consultar os dados armazenados. Utilizando os preceitos da reutilização de código, criamos uma super classe genérica que terá todos os métodos necessários para manipular os registros, bastando assim as entidades herdarem esses comportamentos e a utilizem quando necessitar.

```

1 package com.recycle.web.repository;
2
3 // Imports omitidos
4
5 public abstract class AbstractDao<T, ID extends Serializable> {
6
7     // Recupera a subclasse que foi instanciada
8     @SuppressWarnings("unchecked")
9     private final Class<T> entityClass = (Class<T>) ( (
10         ParameterizedType) getClass().getGenericSuperclass()).
11         getActualTypeArguments()[0];
12
13     // EntityManager responsavel por gerenciar os estados de
14     // transacao do projeto.
15     @PersistenceContext
16     private EntityManager entityManager;
17 }

```

```
15 // Com a visibilidade protegida, apenas as subclasses podem
    acessar o EntityManager.
16 protected EntityManager getEntityManager() {
17     return entityManager;
18 }
19
20 // Salva um registro no banco de dados.
21 public T save(T entity) {
22     return entityManager.merge(entity);
23 }
24
25 // Atualiza um registro no banco de dados.
26 public void update(T entity) {
27     entityManager.merge(entity);
28 }
29
30 // Exclui um registro no banco de dados.
31 public void delete(ID id) {
32     entityManager.remove(entityManager.getReference(entityClass,
33         id));
34 }
35 // Consulta e retorna um registro do banco de dados com base no
    ID passado como parametro.
36 public T findById(ID id) {
37     return entityManager.find(entityClass, id);
38 }
39
40 // Consulta e retorna todos os registros armazenados na banco de
    dados.
41 public List<T> findAll(){
42     return entityManager.createQuery("from "+entityClass.
43         getSimpleName(), entityClass).getResultList();
44 }
45 // Realiza uma consulta personalizada que retorna todos os
    registros armazenados com base nos parametros passados.
46 protected List<T> createQuery(String jpql, Object... params) {
47     TypedQuery<T> query = entityManager.createQuery(jpql,
48         entityClass);
49     for(int i = 0; i < params.length; i++) {
50         query.setParameter(i+1, params[i]);
51     }
52 }
```

```

50     }
51     return query.getResultList();
52 }
53 }

```

#### 2.3.4.5 Interface para o repositório

Como boas práticas no desenvolvimento de sistemas baseados em Orientação a Objetos, o acesso as classes sempre serão através de interface, garantindo que se cumpra o comportamento registrado na interface, técnica conhecida como upcasting.

```

1 package com.recycle.web.repository;
2
3 import java.util.List;
4 import com.recycle.web.domain.ColetaEspecializada;
5
6 public interface ColetaEspecializadaDao {
7     ColetaEspecializada save(ColetaEspecializada coleta);
8     void update(ColetaEspecializada coleta);
9     void delete(Long id);
10    ColetaEspecializada findById(Long id);
11    List<ColetaEspecializada> findAll();
12    List<ColetaEspecializada> findByTipoColeta(String tipo);
13 }

```

#### 2.3.4.6 Classe de persistência implementada

A classe de persistência concentra toda lógica de acesso aos dados para a entidade que a instanciar, neste modelo podemos observar que o método `findByTipoColeta(String tipo)` só existe para a entidade `Coleta Especializada`, e não foi adicionada na super classe. Portanto toda lógica de acesso e manipulação dos dados referentes a `Coleta Especializada` devem ser realizadas nesta classe. Não precisamos registrar os métodos da super classe pois já foram herdados.

```

1 package com.recycle.web.repository;
2
3 import java.util.List;
4 import org.springframework.stereotype.Repository;
5 import com.recycle.web.domain.ColetaEspecializada;
6
7 @Repository
8 public class ColetaEspecializadaDaoImpl extends AbstractDao<
    ColetaEspecializada, Long> implements ColetaEspecializadaDao {

```

```

9
10  @Override
11  public List<ColetaEspecializada> findByTipoColeta(String tipo) {
12      return createQuery("select ce from ColetaEspecializada ce
13                          where ce.tipoColeta = ?1", tipo);
14  }
15  }

```

Quadro 2 – Anotação da camada de persistência.

<b>@Repository</b>
Define um bean como sendo do tipo persistente para uso em classes de acesso a banco de dados. A partir desta anotação o Spring pode usar recursos referentes a persistência, como tratar as exceções específicas para este fim.

Fonte: Autoria própria (2019)

## 2.3.5 Camada de Serviço

### 2.3.5.1 Interface para o serviço

Como já mencionado, todos os acessos serão realizado através de interfaces, adicionamos os comportamentos realizados na camada de serviço, portanto quem instância a interface implementa os métodos que já estão registrados. Nota-se que registramos todos os métodos herdados da super classe na camada de persistência e os métodos específicos definidos apenas no repositório de Coleta Especializada.

```

1  package com.recycle.web.service;
2
3  import java.util.List;
4
5  import com.recycle.web.domain.ColetaEspecializada;
6
7  public interface ColetaEspecializadaService {
8      void save(ColetaEspecializada coleta);
9      void update(ColetaEspecializada coleta);
10     void delete(Long id);
11     ColetaEspecializada findById(Long id);
12     List<ColetaEspecializada> findAll();
13     List<ColetaEspecializada> findByTipoColeta(String tipo);
14 }

```

### 2.3.5.2 Classe de serviço que implementa a interface

Nesta camada será implementada todas as regras do negócio, aquilo que acordado com os usuários será implementado nesta classe. Então se houver alguma especificidade na inserção, atualização, exclusão e consulta nos registros de Coleta Especializada, esse comportamento deverá ser implementado nesta classe.

```
1 package com.recycle.web.service;
2
3 // Imports omitidos
4
5 @Service @Transactional(readonly = false)
6 public class ColetaEspecializadaServiceImpl implements
7     ColetaEspecializadaService {
8
9     @Autowired
10    private ColetaEspecializadaDao dao;
11
12    @Override
13    public void save(ColetaEspecializada coleta) {
14        dao.save(coleta);
15    }
16
17    @Override
18    public void update(ColetaEspecializada coleta) {
19        dao.update(coleta);
20    }
21
22    @Override
23    public void delete(Long id) {
24        dao.delete(id);
25    }
26
27    @Override @Transactional(readonly = true)
28    public ColetaEspecializada findById(Long id) {
29        return dao.findById(id);
30    }
31
32    @Override @Transactional(readonly = true)
33    public List<ColetaEspecializada> findAll() {
34        return dao.findAll();
35    }
36 }
```



```

36  @Override @Transactional(readOnly = true)
37  public List<ColetaEspecializada> findByTipoColeta(String tipo) {
38      return dao.findByTipoColeta(tipo);
39  }
40  }

```

Quadro 3 – Anotação da camada de serviço.

<b>@Service</b>
Esta anotação indica ao Spring que esta classe representa a camada de serviço (Service Layer), e que gerencia as regras de negócio da aplicação.
<b>@Autowired</b>
O Spring ao ler esta anotação deverá injetar a dependência à variável anotada na classe em que está declarada.
<b>@Transactional</b>
<b>Adicionada no topo da classe</b> significa que todos os métodos respeitarão as configuração de transação adicionadas pelo framework.
<b>Adicionada no método</b> indica que estamos sobrescrevendo o comportamento do método.
<b>@Transactional(readOnly = false)</b> valor padrão da propriedade, significa que qualquer método vai ser obrigado a abrir uma transação seja de leitura ou escrita.
<b>@Transactional(readOnly = true)</b> quando executado não será aberto uma transação, significa que quando uma transação é aberta a tabela no banco de dados ficará bloqueada até que transação termine, impossibilitando o de outros métodos, sendo assim em consulta não utilizamos transação.

Fonte: Autoria própria (2019)

## 2.3.6 Camada de Controle

### 2.3.6.1 Classe que implementa

```

1  package com.recycle.web.controller;
2
3  // Imports omitidos
4
5  @Controller
6  @RequestMapping("/coleta")
7  public class ColetaController {
8
9      @Autowired
10     private ColetaEspecializadaService coletaService;
11
12     @Autowired
13     private BairroServiceImpl bairroService;

```

```
14
15 @GetMapping("/cadastrar")
16 public String cadastrar(ColetaEspecializada coletaEspecializada)
17     {
18     return "/coleta/cadastro_coleta";
19 }
20
21 @GetMapping("/listar")
22 public String listar(ModelMap model) {
23     model.addAttribute("locais", coletaService.findAll());
24     return "/coleta/lista_coleta";
25 }
26
27 @PostMapping("/salvar")
28 public String salvar(ColetaEspecializada coletaEspecializada,
29     RedirectAttributes attr) {
30     coletaService.save(coletaEspecializada);
31     attr.addFlashAttribute("success", "Coleta cadastrada com
32     sucesso.");
33     return "redirect:/coleta/cadastrar";
34 }
35
36 @GetMapping("/editar/{id}")
37 public String preEditar(@PathVariable("id") Long id, ModelMap
38     model ) {
39     model.addAttribute("coletaEspecializada", coletaService.
40     findById(id));
41     return "/coleta/cadastro_coleta";
42 }
43
44 @PostMapping("/editar")
45 public String editar(ColetaEspecializada coletaEspecializada,
46     RedirectAttributes attr) {
47     coletaService.update(coletaEspecializada);
48     attr.addFlashAttribute("success", "Local editado com sucesso."
49     );
50     return "redirect:/coleta/cadastrar";
51 }
52
53 @GetMapping("/excluir/{id}")
54 public String excluir(@PathVariable("id") Long id, ModelMap
55     model) {
```

```

48     coletaService.delete(id);
49     model.addAttribute("success", "Local excluído com sucesso.");
50     return listar(model);
51 }
52
53 @ModelAttribute("bairros")
54 public List<Bairro> listaDeBairros() {
55     return bairroService.findAll();
56 }
57 }

```

## 2.3.7 Camada de visualização

### 2.3.7.1 Tela de Cadastro - Coleta Especializada

O formulário de cadastro de Pontos de Coleta Especializada é apresentado na Figure 9, os atributos necessários para o cadastro foram repassados pela cooperativa com uma atenção para o endereço completo, pois o mesmo poderá ser utilizado como fonte para renderizar o mapa, caso na implantação haja condições de contratar os serviços de geolocalização.

Figura 9 – Tela de Cadastro - Coleta Especializada

The screenshot shows a web browser window with the URL 'localhost:8080/coleta/cadastrar'. The page title is 'PPGIT - RECICLE'. The main content area is titled 'Cadastrar Coleta Especializada' and contains a form with the following fields:

- Local:** 'Nome do local' (text input) and 'Selecionar o tipo...' (dropdown menu).
- Atendimento:** 'Ex.: Das 08:00 às 18:00 em horário comercial' (text input).
- Contato:** 'Ex.: (00) 00000-0000' (text input).
- Logradouro:** 'Rua' (text input).
- Número:** 'Número' (text input).
- Bairros:** 'Selecionar o estado...' (dropdown menu).
- CEP:** 'CEP' (text input).
- Cidade:** 'Nome da Cidade' (text input).
- UF:** 'Estado' (text input).

A blue 'Salvar' button is located at the bottom left of the form. A 'Listar Coletas' button is visible in the top right corner of the form area.

Fonte: Autoria própria (2019)

### 2.3.7.2 Tela de Cadastro - Implementação

```

1 <!DOCTYPE html>
2 <!--
3     Uso de Template (layout:decorate=~{layout})

```

```

4  =====
5  Indica qual template sera utilizado para esta view no momento da
6  renderizacao
7  da pagina
8  -->
9  <html layout:decorate="~{layout}">
10 <head>
11 <meta charset="UTF-8">
12 </head>
13 <body>
14 <!--
15     Uso de Template (layout:fragment="conteudo")
16     =====
17     Em qual area do template esta pagina sera renderizada
18 -->
19 <section class="layout-content" layout:fragment="conteudo">
20
21 <nav class="navbar navbar-expand-md bg-light">
22   <div class="collapse navbar-collapse" id="
23     navbarsExampleDefault">
24     <ul class="navbar-nav mr-auto">
25       <li class="nav-item active"><i class="oi oi-caret-right"
26         ></i>
27       <span>Cadastrar Coleta Especializada</span>
28     </li>
29   </ul>
30 </div>
31 <!--
32     Solicitacao de recursos (href="")
33     =====
34     /coleta -> Acessa a classe ColetaController da aplicacao
35     /listar -> Acessa o metodo mapeado com este nome dentro do
36     controller
37 -->
38 <a class="btn btn-primary btn-md" href="/coleta/listar"
39   role="button">
40   <span class="oi oi-spreadsheet" title="Cadastro" aria-
41     hidden="true"></span>
42   <span>Listar Coletas</span>
43 </a>
44 </nav>

```

```

41 <div class="container" id="cadastro">
42 <!--
43 Recurso thymeleaf
44 =====
45 th:replace - substitui a tag da pagina de destino pelo
      fragmento.
46 -->
47 <div th:replace="fragments/alert"></div>
48
49 <!--
50 Recurso thymeleaf
51 =====
52 th:action - Qual acao sera executada.
53 th:object - O objeto instanciado na view
54 -->
55 <form th:action="{coletaEspecializada.id == null} ? @{/
      coleta/salvar} : @{/coleta/editar}"
56       th:object="{coletaEspecializada}" method="POST">
57
58 <div class="form-group row">
59 <div class="col-sm-6">
60 <label for="dia" class="col-form-label">Local</label>
61 <!--
62 Recurso thymeleaf
63 =====
64 *{local} - atributo do objeto instanciado
65 -->
66 <input type="text"
67       class="form-control" id="local"
68       placeholder="Nome do local" autofocus="autofocus"
69       th:field="{local}"/>
70 </div>
71
72 <div class="col-sm-6">
73 <label for="tipoColeta" class="col-form-label">Local</
      label>
74 <select id="tipoColeta" class="form-control" th:field=
      "{tipoColeta}">
75 <option value="" >Selecione o tipo...</option>
76 <option value="Lixo Eletronico">Lixo Eletronico</
      option>

```

```

77         <option value="Oleo de Cozinha">Oleo de Cozinha</
           option>
78         <option value="Lampadas Fluorescentes">Lampadas
           Fluorescentes</option>
79         <option value="Pilhas e Baterias">Pilhas e Baterias<
           /option>
80     </select>
81 </div>
82 </div>
83
84 <div class="form-group row">
85     <div class="col-sm-8">
86         <label for="horario" class="col-form-label">
           Atendimento</label>
87         <input type="text"
88             class="form-control" id="horario"
89             placeholder="Ex.: Das 08:00 as 18:00 em horario
           comercial" autofocus="autofocus"
90             th:field="*{horarioAtendimento}"/>
91     </div>
92
93     <div class="col-sm-4">
94         <label for="contato" class="col-form-label">Contato</
           label>
95         <input type="text"
96             class="form-control" id="contato"
97             placeholder="Ex.: (00) 00000-0000" autofocus="
           autofocus"
98             th:field="*{contato}"/>
99     </div>
100 </div>
101
102 <div class="form-group row">
103     <div class="col-sm-10">
104         <label for="logradouro" class="col-form-label">
           Logradouro</label>
105         <input type="text"
106             class="form-control" id="logradouro"
107             placeholder="Rua" autofocus="autofocus"
108             th:field="*{logradouro}"/>
109     </div>
110

```

```

111     <div class="col-sm-2">
112         <label for="numero" class="col-form-label">Numero</
            label>
113         <input type="text"
114             class="form-control" id="numero"
115             placeholder="Numero" autofocus="autofocus"
116             th:field="*{numero}"/>
117     </div>
118 </div>
119
120 <div class="form-group row">
121     <div class="col-sm-10">
122         <label for="bairro" class="col-form-label">Bairros</
            label>
123         <select id="bairro" class="form-control" th:field="*{
            bairro}">
124             <option value="">Selecione o estado...</option>
125             <!--
126                 Recurso thymeleaf
127                 =====
128                 th:each - percorre a lista de bairros e para cada
                    bairro encontrado
129                 cria um option.
130                 th:value - O valor que sera passado.
131                 th:text - O valor que sera apresentado.
132             -->
133             <option th:each="bairro : ${bairros}"
134                 th:value="${bairro.id}"
135                 th:text="${bairro.nome}"></option>
136         </select>
137     </div>
138
139     <div class="col-sm-2">
140         <label for="cep" class="col-form-label">CEP</label>
141         <input type="text"
142             class="form-control" id="cep"
143             placeholder="CEP" autofocus="autofocus"
144             th:field="*{CEP}"/>
145     </div>
146 </div>
147
148 <div class="form-group row">

```

```

149     <div class="col-sm-10">
150         <label for="cidade" class="col-form-label">Cidade</
            label>
151         <input type="text"
152             class="form-control" id="cidade"
153             placeholder="Nome da Cidade" autofocus="autofocus
                "
154             th:field="*{cidade}"/>
155     </div>
156
157     <div class="col-sm-2">
158         <label for="uf" class="col-form-label">UF</label>
159         <input type="text"
160             class="form-control" id="uf"
161             placeholder="Estado" autofocus="autofocus"
162             th:field="*{uf}"/>
163     </div>
164 </div>
165 <!--
166     Recurso thymeleaf
167     =====
168     th:field - Mantem oculto o ID da instancia
169 -->
170 <input type="hidden" id="id" th:field="*{id}" />
171
172 <!-- Realiza a submissao do formulario -->
173 <button type="submit" class="btn btn-primary btn-sm">
174     <span class="oi oi-check" title="Cadastro" aria-hidden="
            true"></span>
175     <span>Salvar</span>
176 </button>
177 </form>
178 </div>
179 </section>
180 </body>
181 </html>

```

### 2.3.7.3 Tela de Consulta - Coleta Especializada

Na página de consulta de Coletas Especializadas Figure 10, é possível verificar todas as informações cadastrados e os tipos de coletas, bem como a possibilidade de excluir ou alterar algum registro.



Figura 10 – Tela de Consulta - Coleta Especializada

#	Local	Tipo de Coleta	Contato		
1	Fogo Atacadista	Pilhas e Baterias	(67) 3461-6000		
2	Supermercado Santos	Pilhas e Baterias	(67) 3461-6000		
5	Platextil	Lixo Eletrônico	(67) 3462-6876		
6	Energisa	Lixo Eletrônico	(67) 3461-4682		
7	Fogo Atacadista	Lâmpadas fluorescentes	(67) 3461-6000		
8	Deposito Santa Catarina	Lâmpadas fluorescentes	(67) 3461-6859		
3	Yolanda Tiemi Sakuraimé	Óleo de cozinha	(67) 3461-3379		
4	Camargo Distribuidora	Óleo de cozinha	(67) 3461-7889		

Fonte: Autoria própria (2019)

### 2.3.7.4 Tela de Consulta - Implementação

```

1 <!DOCTYPE html>
2 <html layout:decorate="~{layout}">
3 <head>
4 <meta charset="UTF-8">
5 </head>
6 <body>
7 <section class="layout-content" layout:fragment="conteudo">
8
9 <nav class="navbar navbar-expand-md bg-light">
10 <div class="collapse navbar-collapse" id="
11 <ul class="navbar-nav mr-auto">
12 <li class="nav-item active"><i class="oi oi-caret-right"
13 <span>Lista de Locais de Coleta</span></li>
14 </ul>
15 </div>
16 <!--
17 <a href="#">Solicitacao de recursos (href="#")
18 =====
19 /coleta -> Acessa a classe ColetaController da aplicacao

```

```

20     /cadastrar -> Acessa o metodo mapeado com este nome dentro
        do controller, abrir formulario
21     -->
22     <a class="btn btn-primary btn-md" href="/coleta/cadastrar"
23         role="button"> <span class="oi oi-plus" title="Cadastro"
24         aria-hidden="true"></span> <span>Novo Local</span>
25     </a>
26 </nav>
27 <div class="container" id="listagem">
28
29     <div th:replace="fragments/alert"></div>
30
31     <div class="table-responsive">
32         <table class="table table-striped table-hover table-sm">
33             <thead>
34                 <tr>
35                     <th>#</th>
36                     <th>Local</th>
37                     <th>Tipo de Coleta</th>
38                     <th>Contato</th>
39                     <th></th>
40                 </tr>
41             </thead>
42             <tbody>
43                 <!--
44                 Recurso thymeleaf
45                 =====
46                 th:each - percorre a lista de locais e para cada
                        local encontrado
47                 uma linha na tabela.
48                 th:text - Representa o valor que sera apresentado.
49                 -->
50                 <tr th:each="local : ${locais}">
51                     <td th:text="${local.id}"></td>
52                     <td th:text="${local.local}"></td>
53                     <td th:text="${local.tipoColeta}"></td>
54                     <td th:text="${local.contato}"></td>
55
56                     <td colspan="2">
57                         <!--
58                         Solicitudacao de recursos (th:href="")

```

```

59      =====
60      /coleta -> Acessa a classe ColetaController da
        aplicacao
61      /editar -> Acessa o metodo mapeado com este nome
        dentro do controller, passando o {id}
62      como parametro
63      -->
64      <a class="btn btn-info btn-sm"
65          th:href="@{/coleta/editar/{id} (id=${local.id})
        }"
66          role="button">
67          <span class="oi oi-brush" title="Editar" aria-
        hidden="true"></span>
68      </a>
69
70      <button th:id="'btn_coleta/excluir/' + local.id}
        "
71          type="button"
72          class="btn btn-danger btn-sm"
73          data-toggle="modal"
74          data-target="#myModal">
75      <span class="oi oi-circle-x"
76          title="Exclusao desabilitada"
77          aria-hidden="true">
78      </span>
79      </button>
80      </td>
81      </tr>
82      </tbody>
83      </table>
84      </div>
85      </div>
86      <!--
87      Solicitacao de recursos
88      =====
89      th:replace - substitui a tag da pagina de destino pelo
        fragmento.
90      -->
91      <div th:replace="fragments/modal :: modal"></div>
92
93      </section>

```

```
94 </body>
```

```
95 </html>
```

## 2.3.8 Arquitetura REST

### 2.3.8.1 Implementação da camada - Coleta Especializada

```
1 package com.recycle.web.resources;
2
3 \\ Imports omitidos
4
5 import com.recycle.web.domain.ColetaEspecializada;
6 import com.recycle.web.service.ColetaEspecializadaService;
7
8 @RestController
9 @RequestMapping(value = "/coletas")
10 public class ColetaResource {
11
12     @Autowired
13     private ColetaEspecializadaService service;
14
15     @CrossOrigin
16     @GetMapping("/{tipo}")
17     public Iterable<ColetaEspecializada> getTipoColeta(@PathVariable
18         ("tipo") String tipo) {
19         return service.findByTipoColeta(tipo);
20     }
21 }
```

## 2.3.9 Desenvolvimento do aplicativo móvel com Ionic

### 2.3.9.1 Interface de domínio

```
1 /**
2  * Interface que representara a coleta especializada
3  * na aplicacao mobile, quando for instanciada a partir
4  * da API.
5  */
6 export interface Coleta {
7     id: number;
8     local: string;
9     tipoColeta: string;
10    horarioAtendimento: string;
11    contato: string;
12    logradouro: string;
```

```

13     numero: string;
14     bairro: string;
15     cep: string;
16     cidade: string;
17     uf: string;
18 }

```

### 2.3.9.2 Provider que conexão com a API

```

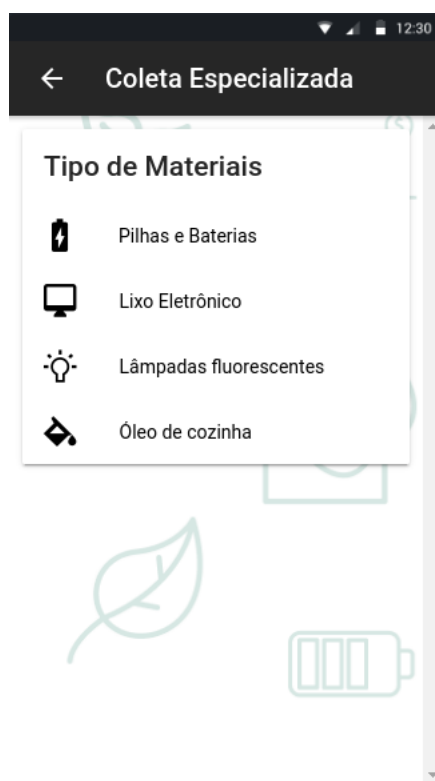
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { Coleta } from '../../domain/coleta';
4
5 /** coletaService : prove o acesso a API */
6 @Injectable()
7 export class ColetaServiceProvider {
8     public coletas: Coleta[];
9
10    public tipoColeta: string;
11
12    constructor(private _http: HttpClient) {
13    }
14
15    lista() {
16        /** Conecta ao servidor onde a API foi disponibilizada e
17         * acessa os dados */
18        return this._http.get<Coleta[]>("http://localhost:8080/coletas/" + this.tipoColeta);
19    }
20 }

```

### 2.3.9.3 Tela de escolha por tipo

Para facilitar para o usuário ao escolher a opção Coleta Especializada na *dashboard*, será apresentado a tela com os tipos de materiais Figure 11 coletados, assim poderá visualizar apenas os pontos de coletas para o tipo específico.

Figura 11 – Tela de consulta por tipo de Coleta



Fonte: Autoria própria (2019)

#### 2.3.9.4 Tela de escolha - Implementação das regras de negócio

```

1 import { Component } from '@angular/core';
2 import { IonicPage, NavController, NavParams } from 'ionic-angular';
3 import { ColetaPage } from '../coleta/coleta';
4
5 @IonicPage()
6 @Component({
7   selector: 'page-tipo-coleta',
8   templateUrl: 'tipo-coleta.html',
9 })
10 export class TipoColetaPage {
11
12   //Vetor que armazenara os tipos de coleta
13   public tipoColetas: {
14     icon: string;
15     tipoColeta: string;
16   }[];

```

```

17
18     constructor(public navCtrl: NavController,
19         public navParams: NavParams) {
20         /**
21          * Lista dos tipos de coletas
22          * */
23         this.tipoColetas = [
24             {
25                 icon: "battery-charging",
26                 tipoColeta: "Pilhas e Baterias"
27             },
28             {
29                 icon: "desktop",
30                 tipoColeta: "Lixo Eletronico"
31             },
32             {
33                 icon: "bulb",
34                 tipoColeta: "Lampadas fluorescentes"
35             },
36             {
37                 icon: "color-fill",
38                 tipoColeta: "Oleo de cozinha"
39             }
40         ];
41     }
42
43     /**
44     * Metodo consultarTipo
45     * Quando acionado executa o metodo push que devera
46     * abrir a tela de ColetaPage levando como parametro o * tipo
47     * selecionado
48     */
49     consultarTipo(tipo: string){
50         this.navCtrl.push(ColetaPage.name, {tipoColeta: tipo});
51     }

```

### 2.3.9.5 Tela de escolha - Implementação da página de visualização

```

1 <ion-header>
2   <ion-navbar color="dark">
3     <ion-title>Coleta Especializada</ion-title>
4   </ion-navbar>

```

```

5 </ion-header>
6 <ion-content class="background">
7   <ion-card>
8     <ion-card-header color="dark" style="font-size: 1.5em; font-
9       weight: 500;">
10      Tipo de Materiais
11    </ion-card-header>
12    <ion-list>
13      <!--
14        Para cada tipo de coleta sera renderizado um <button>
15
16        *ngFor, percorre o vetor e passo o valor para a
17        variavel tipo
18
19        (click), evento que executa o metodo consultarTipo
20        e passa o valor do tipo de coleta selecionado
21      -->
22    <button ion-item class="button"
23      *ngFor="let tipo of tipoColetas"
24      (click)="consultarTipo(tipo.tipoColeta)">
25      <ion-icon name="{{tipo.icon}}" item-start></ion-icon>
26      {{tipo.tipoColeta}}
27    </button>
28  </ion-list>
29 </ion-card>
30 </ion-content>

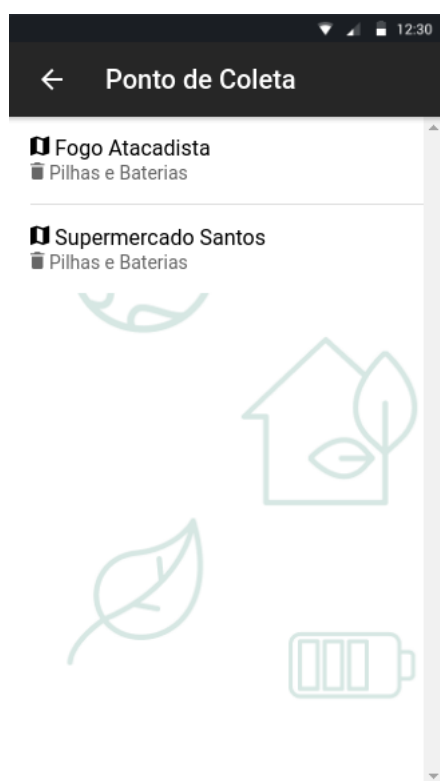
```

### 2.3.9.6 Telas de coleta Especializada

Uma vez selecionado o tipo de material será apresentado uma nova tela com os pontos cadastrados, Figure 12, o usuário escolhe o ponto mais mas próximo de sua residência e terá acesso ao endereço do local conforme a Figure 13.



Figura 12 – Lista de Locais



Fonte: Autoria própria (2019)

Figura 13 – Local selecionado



Fonte: Autoria própria (2019)

### 2.3.9.7 Tela de consulta - Implementação das regras de conexão

```

1 import { Component } from '@angular/core';
2 import { IonicPage, NavController, NavParams, LoadingController,
   AlertController } from 'ionic-angular';
3 import { Coleta } from '../domain/coleta';
4 import { ColetaMapaPage } from '../coleta-mapa/coleta-mapa';
5 import { ColetaServiceProvider } from '../providers/coleta-
   service/coleta-service';
6 import { HttpResponse } from '@angular/common/http';
7
8
9 @IonicPage()
10 @Component({
11   selector: 'page-coleta',
12   templateUrl: 'coleta.html',
13 })
14 export class ColetaPage {
15

```

```
16 public coletas: Coleta[];
17 public tipoColeta: any;
18
19 constructor(public navCtrl: NavController,
20 public navParams: NavParams,
21 private _loadingCtrl: LoadingController,
22 private _alertCtrl: AlertController,
23 private _coletaService: ColetaServiceProvider) {
24
25 }
26
27 /** ionViewDidLoad
28  * - vai executar quando terminar o carregamento da pagina */
29 ionViewDidLoad() {
30
31     //Passa para o Provider o tipo de coleta que sera consultada
32     this._coletaService.tipoColeta = this.navParams.get('
33         tipoColeta');
34
35     /** Cria o componente de loading para a lista de agendamentos
36         */
37     let loading = this._loadingCtrl.create({
38         content: "Carregando Pontos de Coleta..."
39     });
40
41     /** present() - Apresenta o componente de loading enquanto a
42         lista e carregada */
43     loading.present();
44
45     /** agendaService: supre a conexao com a API */
46     this._coletaService.lista().subscribe(
47         coleta => {
48             this.coletas = coleta;
49             /** dismiss() - Retira da tela o componente de loading
50                 depois que a lista foi carregada */
51             loading.dismiss();
52         },
53         /** Se houver algum erro na conexao */
54         (error: HttpResponse) => {
55             console.log(error);
56             loading.dismiss();
57             this._alertCtrl
```

```

54         .create({
55             title: "Falha na conexao",
56             subTitle: "Nao foi possivel carregar os pontos de
                    coleta, tente mais tarde",
57             buttons: [{ text: "Ok" }]
58         })
59         .present();
60     }
61 );
62 }
63
64 selecionaColeta(coleta: Coleta) {
65     /** push() - utilizamos dois parametros
66      * - pagina que sera carregada
67      * - envia informacoes para a outra pagina
68      */
69     this.navCtrl.push(ColetaMapaPage.name, {
70         coletaSelecionada: coleta
71     });
72 }
73
74 }

```

### 2.3.9.8 Tela de consulta - Implementação da lista de tipos encontrados

```

1 <ion-header>
2   <ion-navbar color="dark">
3     <ion-title>Ponto de Coleta</ion-title>
4   </ion-navbar>
5 </ion-header>
6 <ion-content class="background">
7   <ion-list>
8     <ion-item *ngFor="let coleta of coletas" (click)="
          selecionaColeta(coleta)">
9       <div>
10        <ion-icon name="map"></ion-icon>
11        <span>{{coleta.local}}</span>
12      </div>
13      <div>
14        <p>
15          <ion-icon name="trash"></ion-icon>
16          {{coleta.tipoColeta}}</p>
17      </div>

```

```
18     </ion-item>
19 </ion-list>
20 </ion-content>
```

## 2.4 Geolocalização utilizando Google Maps

A apresentação dos locais de coleta especializada, bem como rotas da coleta seletiva seriam interessante ser apresentadas por mapas com imagens, as visualizações das postagens no aplicativo seriam mais atraente para o usuário final. Normalmente as prefeituras já possuem contratos com plataformas digitais como Google, para utilizar e-mail institucional, dentre os vários serviços oferecidos pela plataforma. É possível integrar no aplicativo os serviços de geolocalização utilizando o google maps

Na plataforma do Google Maps que pode ser encontrada no seguinte endereço <https://developers.google.com/maps/documentation/javascript/tutorial?hl=pt>, é possível obter todas as informações para implantação, o google possui um manual técnico completo com o código fonte exemplificativo, portanto basta utilizar a documentação oficial e utilizar os mapas no aplicativo.

Para utilizar os mapas no Ionic será necessário implementar recursos nativos, que o framework já traz consigo a documentação e a codificação necessária para implementar o uso de mapas nos smartphones, sua documentação oficial pode ser lida no sitio oficial do Ionic <https://ionicframework.com/docs/native/geolocation/>.

## 2.5 Sistema de armazenamento de imagens

Tanto o aplicativo quanto o ambiente administrativo, poderá utilizar imagens, seja na forma de ícones ou imagens como as adicionadas nas matérias educativas, ver a Figure 24. Já citamos que o projeto poderá utilizar contratos com serviços de terceiros como o Google Maps, para o armazenamento de imagens indicamos também os serviços do Google Cloud, que oferece um serviço de armazenamento, que poderá ser encontrado no seguinte endereço web <https://cloud.google.com/>, outros serviços também estão disponíveis, como o Amazon S3 que também oferece serviços de armazenamento e pode ser encontrado no sitio oficial <https://aws.amazon.com/pt/s3/>. Os serviços apresentados, possuem manual técnico, portanto os desenvolvedores poderão implementar essas funcionalidades com apoio da equipe de suporte.

### 3 MATERIAIS E MÉTODOS

#### 3.1 Estrutura e organização do Sistema

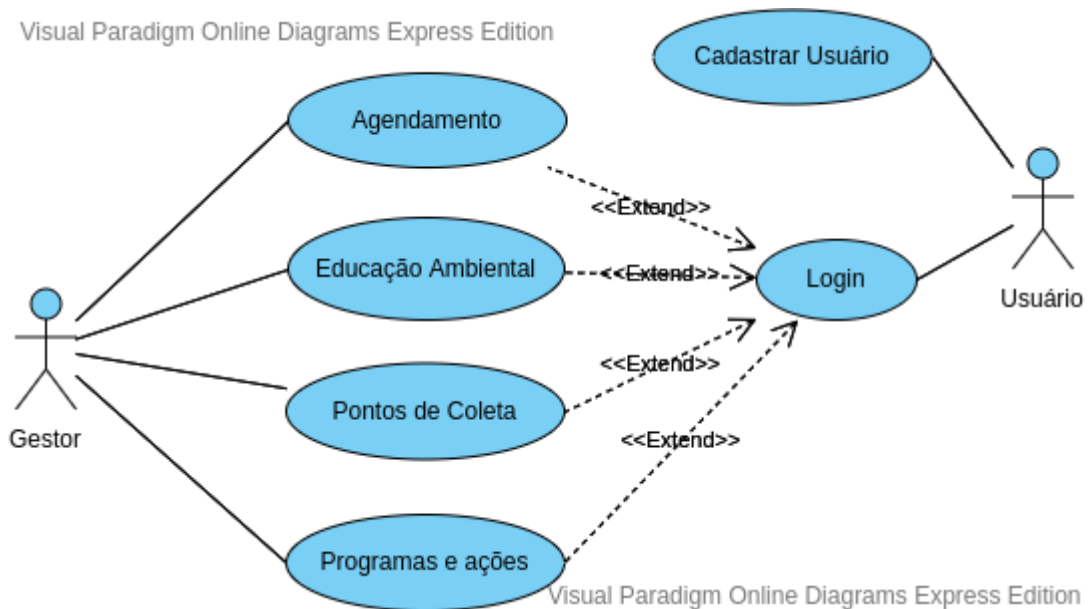
##### 3.1.1 Diagrama de Casos de Uso

O diagrama de Casos de Uso é um meio de comunicar o comportamento do sistema, além de identificar quem ou o que interage com o mesmo. Representa todos os requisitos identificados e que serão implementados, o que o torna um artefato de software indispensável para o planejar e construir a aplicação.

Um caso de uso representa uma determinada funcionalidade de um sistema conforme externamente. Representa também os agentes externos que interagem com o sistema. Um caso de uso, entretanto não revela a estrutura e o comportamento internos do sistema. (BEZERRA, 2007)

Neste exemplo identificamos como atores o Usuário que representa quem interage com aplicativo, quem vai utilizar as informações adicionadas no sistema, e o Gestor que gerenciará os dados através do sistema administrativo. As funcionalidades que listamos para este aplicativo estão apresentadas na forma de Casos de Uso dispostos na Figure 14, nada impede que novas funcionalidades sejam requisitados e implementados, já que o sistema foi modularizado para ser escalável com facilidade.

Figura 14 – Diagrama de Casos de Uso



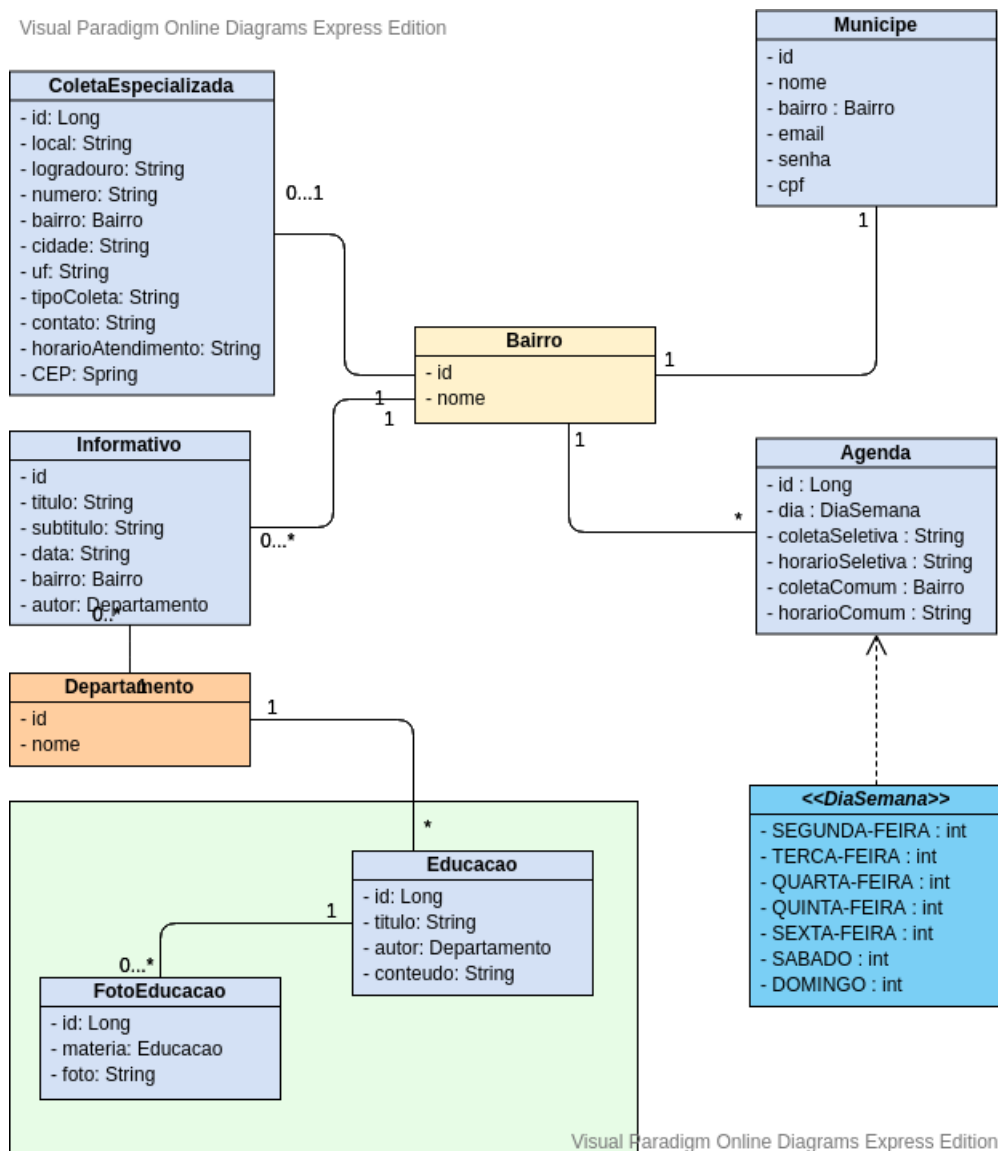
Fonte: Autoria própria (2019)

### 3.1.2 Diagrama de Classes

O diagrama de classes, como mostra a Figure 15, representa a estrutura de um sistema desenvolvido baseado no paradigma Orientado a Objetos, com ele podemos separar as responsabilidades e estabelecer a colaboração entre os objetos, de forma a garantir a manutenibilidade e escalabilidade do projeto após sua implantação.

Após extrairmos dos requisitos os objetos da aplicação, precisaremos separar e classificar suas características, modelando, por conseguinte, as classes do sistema. Entretanto, a essência de um sistema não está apenas em suas classes, mas principalmente nos seus relacionamentos. Já diz o ditado que "uma andorinha não faz verão", portanto não conseguiremos um sistema com classes isoladas (MELO, 2010).

Figura 15 – Diagrama de Classes

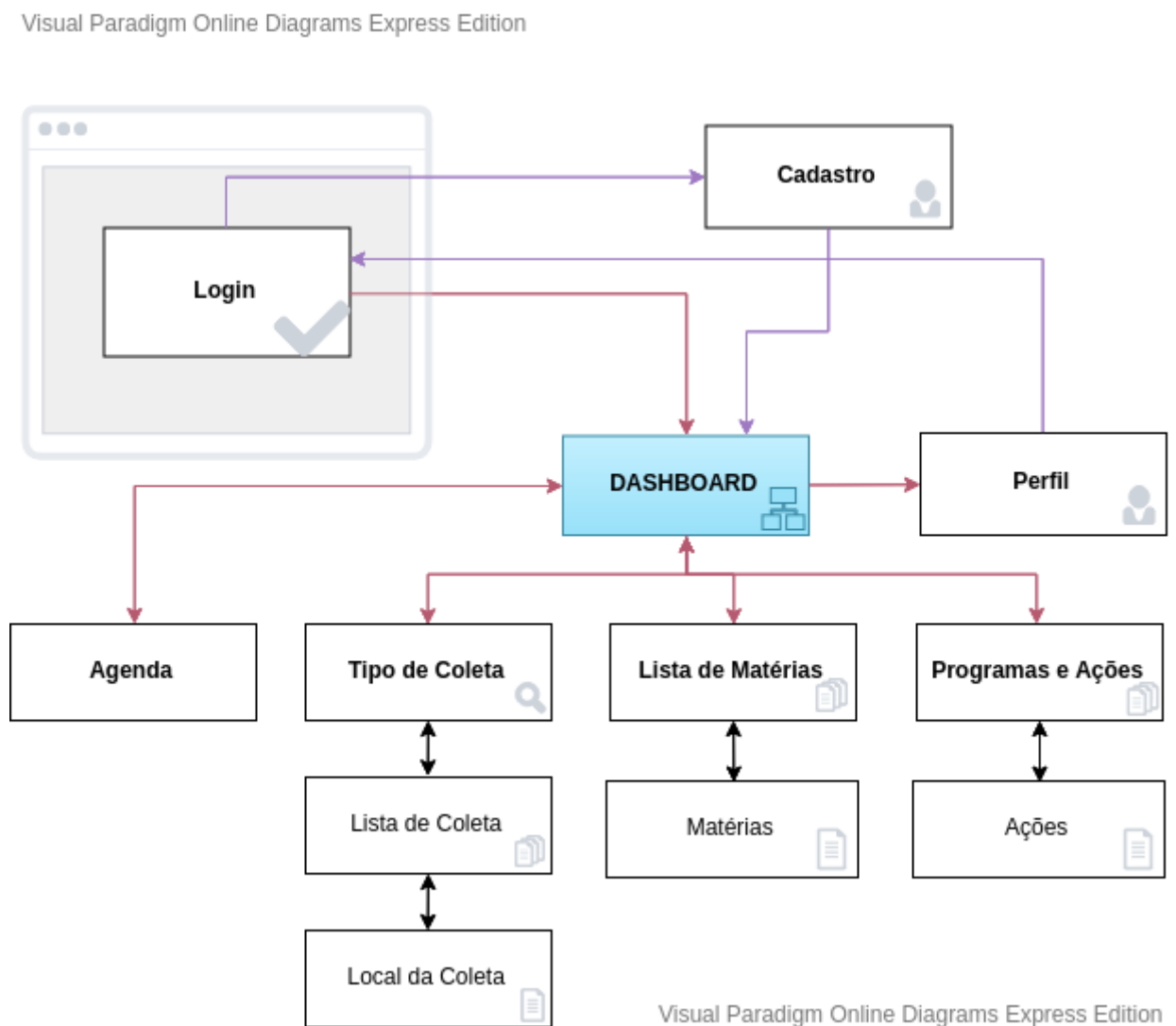


Fonte: Autoria própria (2019)

### 3.2 Estrutura do Aplicativo

Importante para novas implementações no aplicativo é conhecer as ramificações do sistema, para que no momento em que excluirmos um recurso ou adicionar uma nova funcionalidade não quebrar as conexões entre as páginas ou deixar de adicionar acesso ao recurso inserido. Manter uma estrutura confiável garante uma boa experiência com o usuário, erros de links quebrados pode acarretar em má usabilidade e navegabilidade, o sucesso do aplicativo esta ligado diretamente a facilidade e acessibilidade em buscar e manipular as informações. A Figure 16 apresenta o *sitemap* do aplicativo, a forma como o aplicativo foi estruturado, quando for inserido um novo recurso poderá manter a mesma estrutura.

Figura 16 – Mapa do aplicativo

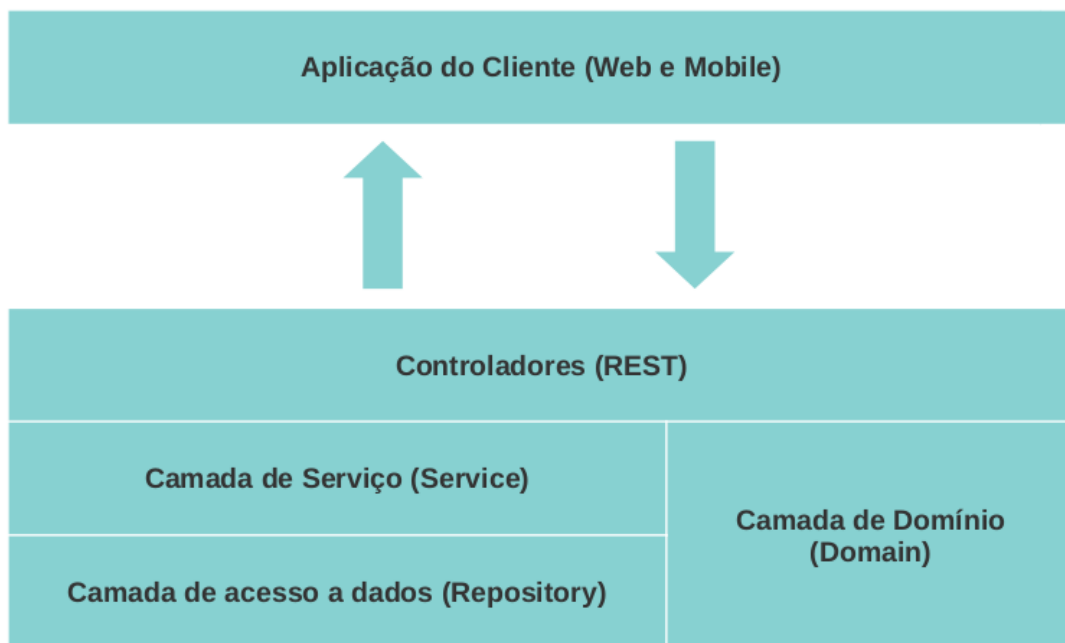


Fonte: Autoria própria (2019)

### 3.3 Arquitetura do Software

No projeto foi utilizado o conceito de separação de responsabilidade e colaboração de serviços, como mostra a Figure 17, separando em camadas o projeto o se torna muito mais flexível à mudanças. Portanto o projeto consiste em três camadas principais, cada uma com responsabilidades específicas, iniciando com a camada de acesso a dados que mantém todas as regras de manipulação das informações contidas no banco de dados e como acessá-las. Acima desta temos a camada de serviço que concentrará as regras de negócios do projeto e como os dados serão tratados e validados, entre as duas, temos a camada de domínio que representa o negócio modelado e como as instâncias colaboram entre si. Entre a interface dos usuários e o sistema está a camada de controle, que tem a responsabilidade de gerenciar as requisições dos usuários e responder as solicitações com as rotas das páginas que serão renderizadas.

Figura 17 – Desenvolvimento em Camadas



Fonte: Autoria própria (2019)



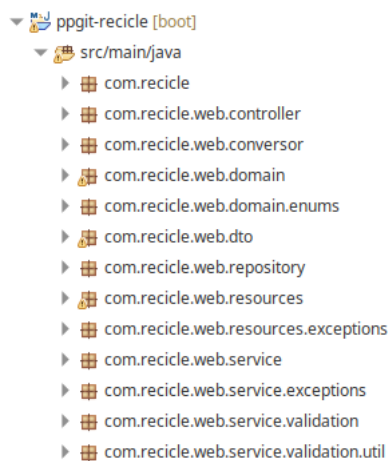
## 4 RESULTADOS E DISCUSSÕES

### 4.1 Sobre a Construção do protótipo

Este estudo se baseia em uma pesquisa exploratória, mediante a análise de dados qualitativos focando no estudo das leis e decretos que fundamentam as principais ações que estabelecem normas e regras relacionadas à sustentabilidade, educação ambiental e coleta seletiva a serem implantadas nos municípios. A pesquisa apresenta um protótipo para a construção de um aplicativo mobile que reúne as principais ações para manter um diálogo entre prefeituras e seus munícipes.

#### 4.1.1 Estrutura e organização do código

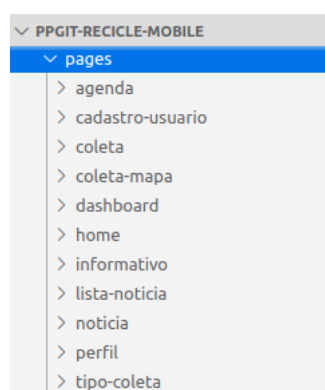
Figura 18 – Estrutura - Spring Boot



Fonte: Autoria própria (2019)

Para a construção do software, utilizamos o STS que é um projeto Spring, que facilita a construção do software apoiando o desenvolvedor no foco do problema e abstraindo a complexidade, tornando o desenvolvimento rápido e seguro. Utilizando o conceito de pacotes, apresentado na Figure 18, separamos em módulos para facilitar as equipes que poderão adicionar novas funcionalidade ao sistema bem como facilitará o estudo da arquitetura do software, e com todas as responsabilidades separadas em escopo distintos é possível planejar mudanças pontuais.

Figura 19 – Estrutura - Ionic



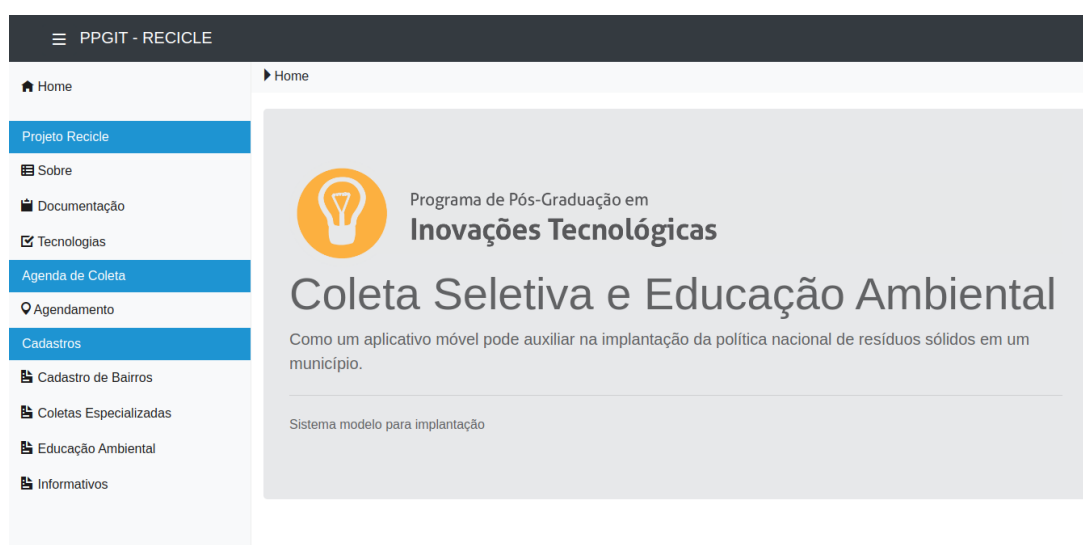
Fonte: Autoria própria (2019)

A construção do aplicativo seguiu-se o modelo do sistema construído no STS, utilizamos o Visual Studio Code que também permite a separação das responsabilidades por pacotes. A Figure 19 apresenta esse modelo, que tem como objetivo a facilidade em desenvolver novas funcionalidades para o aplicativo sem afetar o que já está pronto, portanto o sistema está preparado para ser remodelado baseado no ambiente onde será implantado.

#### 4.1.2 Ambiente Administrativo

Para que aplicativo seja dinâmico com informações atualizadas, foi desenvolvido um ambiente administrativo para o gestor municipal ou membros da cooperativa possam inserir conteúdos que serão visualizados pelos usuários no sistema mobile. Na Figure 20 podemos observar a tela principal do sistema, com acesso aos recursos do sistemas. Uma seção sobre o projeto foi criada ver a aba "Projeto Recicle" onde foi adicionado um ambiente para apresentar o projeto, parte da documentação e as tecnologias que foram utilizadas na construção do software. Logo abaixo temos acesso aos cadastros e gerenciamento dos conteúdos que serão disponibilizados no aplicativo.

Figura 20 – Tela principal do ambiente administrativo.



Fonte: Autoria própria (2019)

As telas de cadastro foram criadas utilizando o mesmo padrão, portanto há uma

facilidade na manutenção do software ou mesmo na criação de novos formulários. Na Figure 21 podemos observar o padrão do formulário de cadastro, que também servirá para editar os registros.

Figura 21 – Tela de cadastro.

PPGIT - RECICLE

▶ Cadastrar Coleta Especializada Listar Coletas

Local  Local

Atendimento  Contato

Logradouro  Número

Bairros  CEP

Cidade  UF

Salvar

Fonte: Autoria própria (2019)

Na tela demonstrada pela Figure 22 podemos consulta a lista de registros mantidos no banco de dados, nesta página é possível editar os registros, que renderizá o formulário de cadastro para executar a tarefa. Para excluir, basta escolher a opção de exclusão e confirmar que o registro será deletado da base de dados.

Figura 22 – Tela de consulta e edição.

PPGIT - RECICLE

▶ Lista de Locais de Coleta + Novo Local

#	Local	Tipo de Coleta	Contato	
1	Fogo Atacadista	Pilhas e Baterias	(67) 3461-6000	
2	Supermercado Santos	Pilhas e Baterias	(67) 3461-6000	
5	Platextil	Lixo Eletrônico	(67) 3462-6876	
6	Energisa	Lixo Eletrônico	(67) 3461-4682	
7	Fogo Atacadista	Lâmpadas fluorescentes	(67) 3461-6000	
8	Deposito Santa Catarina	Lâmpadas fluorescentes	(67) 3461-6859	
3	Yolanda Tiemi Sakuraime	Óleo de cozinha	(67) 3461-3379	
4	Camargo Distribuidora	Óleo de cozinha	(67) 3461-7889	

Fonte: Autoria própria (2019)

Todas as páginas seguem o mesmo padrão de desenvolvimento, podemos realizar manutenção na estrutura do site e atingir todo o sistema, com isso garantimos manutenção rápida e escalabilidade sem grandes mudanças na base do projeto.

#### 4.1.3 Uso do Aplicativo

O aplicativo poderá ser disponibilizado para download nas plataformas Android e IOS. O usuário poderá utilizar o aplicativo (app) sem se cadastrar para acessar as seções relacionadas à educação ambiental, ao mapa de coleta do município e aos pontos de coletas especializadas. Para ter acesso a todas as funcionalidades, o usuário deverá realizar o cadastro e informar qual seu bairro. Com isso, visualizará a agenda de coleta seletiva de sua localidade bem como a coleta de entulhos, e terá ainda funcionalidades relacionadas à comunicação com a prefeitura ou departamento responsável e à cooperativa de recicláveis.

#### 4.1.4 Principais funções do Aplicativo

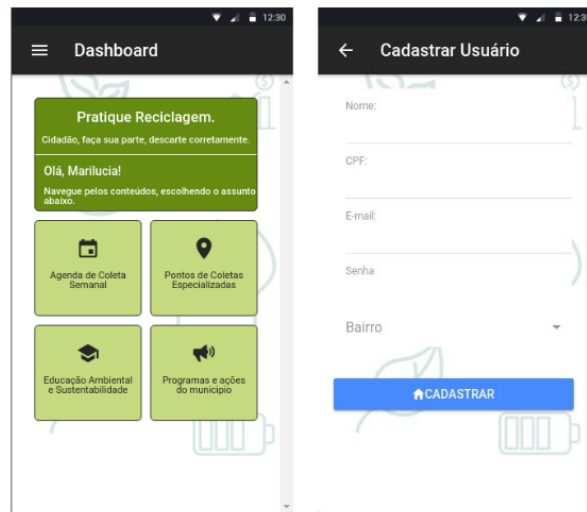
O usuário, depois de instalado o aplicativo em seu smartphone, terá acesso à tela principal, como mostra a Figure 23, onde poderá abrir a seção para visualizar o mapa de coleta seletiva do município, dias e horários e tipos de coleta. Terá acesso também às seções de pontos de coleta e educação ambiental. Nesta, o usuário poderá acessar vários artigos e vídeos instrutivos sobre sustentabilidade, meio ambiente, coleta seletiva, dentre outros temas que a gestão julgar pertinente. Na segunda tela, apresentada pela Figure 24, o usuário poderá visualizar onde há coleta especializada para que possa depositar o lixo que não poderá ir para o aterro e que também não é reciclável, como pilhas e baterias, lâmpadas fluorescentes e lixos eletrônicos.

Quando cadastrado, o usuário receberá informações referentes às coletas do seu bairro, se houver mudanças, por motivos de manutenção de caminhões por exemplo, poderá se orientar para depositar seu lixo no dia correto. Também poderá visualizar a agenda de coleta do município. O usuário poderá cadastrar materiais recicláveis para que os catadores possam coletar diretamente em sua residência.

A prefeitura manterá atualizado no aplicativo o mapa com todos os pontos de coletas específicas, e incentivará a ampliação dos pontos criando outras categorias.

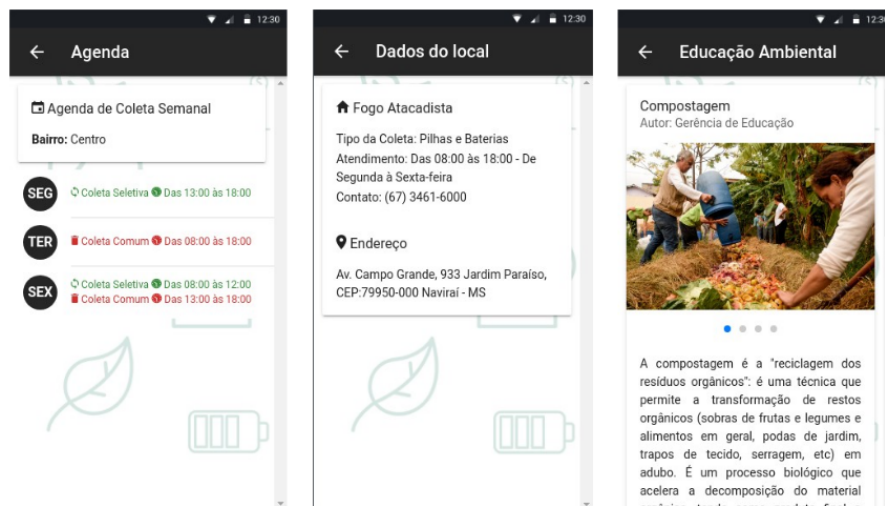
A seção destinada à educação ambiental deverá ser mantida pela prefeitura, que poderá criar mecanismos de cooperação com as escolas e a comunidade para gerar conteúdo destinado à educar a população sobre sustentabilidade e meio ambiente. Os temas poderão ser diversos como compostagem, separação dos recicláveis e hábitos sustentáveis. A prefeitura recebe as postagens e faz uma seleção para disponibilizá-las no aplicativo, bem como poderá criar concursos para geração de conteúdo.

Figura 23 – Tela principal e de cadastro de usuários.



Fonte: Autoria própria (2019)

Figura 24 – Telas de agenda, pontos de coleta e educação ambiental



Fonte: Autoria própria (2019)

#### 4.1.5 Implantação e utilização do Aplicativo

O projeto contendo o aplicativo, o código-fonte e a documentação associada ficará disponível no acervo de Softwares do Governo Federal, mantido pelo “Portal do Software Público Brasileiro do Ministério do Planejamento, Desenvolvimento e Gestão” (SOFTWAREPUBLICO, 2018).

O Software Público Brasileiro é um tipo específico de software livre que atende às necessidades de modernização da administração pública de qualquer dos Poderes da União, dos Estados, do Distrito Federal e dos Municípios e é compartilhado sem ônus no Portal do Software Público Brasileiro, resultando na economia de recursos públicos e constituindo um

recurso benéfico para a administração pública e para a sociedade (SOFTWAREPÚBLICO, 2018).

Parcerias com entidades públicas ou privadas, cujo ramo de atividades se relaciona com sustentabilidade, coleta seletiva e meio ambiente, como o Instituto das Águas e Instituto Ambiental do Paraná, também poderão utilizar o projeto do aplicativo e ou incentivar seu uso por gestores, que buscam novas técnicas ou ferramentas para otimizar a coleta seletiva em seus municípios.

#### 4.1.6 Publicações

Como o tema deste projeto é de interesse social e os gestores municipais tem responsabilidade legal de aplicar a Política Nacional de Resíduos Sólidos, os projetos de pesquisa com foco em educação ambiental, reciclagem, coleta seletiva e sustentabilidade devem, quando possível, ser publicados em eventos e congressos para que esta publicidade auxilie na implantação de novas técnicas e tecnologias aplicadas ao meio ambiente.

O presente trabalho foi submetido e aceito pela Comissão Organizadora do IX Congresso Brasileiro de Engenharia de Produção - Engenharia na Indústria da Bebida, para apresentação e publicação nos Anais do evento, na área temática 10. Engenharia da Sustentabilidade e na modalidade apresentação oral no evento e pode ser consultado nos anais do evento no seguinte endereço [http://aprepro.org.br/conbrepro/2019/anais/arquivos/09292019\\_150914\\_5d90fe4e01a79.pdf](http://aprepro.org.br/conbrepro/2019/anais/arquivos/09292019_150914_5d90fe4e01a79.pdf).

#### 4.1.7 Registro de Software

Foi requerido junto ao Núcleo de Inovação Tecnológica - NIT da UTFPR Câmpus Campo Mourão o registro de software, que garantirá a comunidade o uso do sistema sem custo. Posteriormente o software será incluído em uma licença de software livre, assim outras equipes poderão incluir ou melhorar as funcionalidades do aplicativo.

## 5 CONCLUSÃO

Criar uma cultura de coleta seletiva em um município, vai muito além de alguns trabalhos esporádicos de conscientização. Vale lembrar que o Governo Federal já estabeleceu normas e diretrizes que apresentam, inclusive, os responsáveis à observância dessas leis como está previsto na Lei nº 12.305 em seu Art. 1º parágrafo 1º “Estão sujeitas à observância desta Lei as pessoas físicas ou jurídicas, de direito público ou privado, responsáveis, direta ou indiretamente, pela geração de resíduos sólidos e as que desenvolvam ações relacionadas à gestão integrada ou ao gerenciamento de resíduos sólidos.” (BRASIL, 2010). E também como é apresentado no Decreto nº 7.404 em seu Art. 7º: “O Poder Público, o setor empresarial e a coletividade são responsáveis pela efetividade das ações voltadas para assegurar a observância da Política Nacional de Resíduos Sólidos [...]” (BRASIL, 2010).

No final a responsabilidade de criar uma cultura mais sustentável aplicar coleta seletiva, desenvolver políticas públicas e planejamento urbano ficará a cargo do poder público como está previsto em lei.

Como esta pesquisa apresenta um protótipo que poderá ser implementado por qualquer gestor público que queira utilizar um aplicativo como ferramenta de comunicação entre a prefeitura e seus cidadãos. A ideia é que o aplicativo seja desenvolvido de acordo com a região ou cidade, tratando também das especificidades do local e que possa ser disponibilizado em plataforma pública para que possa ser utilizado por outros gestores. Apresenta-se neste protótipo alguns direcionamentos para sua implementação que também possa suprir as exigências previstas em Lei.

## REFERÊNCIAS

- AGUASPARANA: Programa Reciclo: Sistema Integrado de Coleta Seletiva.. [S. l.], 2018. Disponível em: <http://www.aguasparana.pr.gov.br/modules/conteudo/conteudo.php?conteudo=326>. Acesso em: 14 dez. 2018.
- BEZERRA, E. **Princípios de Análise e Projeto de Sistemas com UML**. 2. ed. rev. e atual. Rio de Janeiro: Elsevier, 2007.
- BRASIL. [Constituição (1988)]. **Constituição da República Federativa do Brasil de 1988**. Brasília: Presidência da República, 1988. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/constituicao/constituicao.htm](http://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm). Acesso em: 14 dez. 2018.
- BRASIL. **Lei nº 9.795, de 27 de abril de 1999**. Dispõe sobre a educação ambiental, institui a Política Nacional de Educação Ambiental e dá outras providências. [S. l.], 27 abr. 1999. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/LEIS/L9795.htm](http://www.planalto.gov.br/ccivil_03/LEIS/L9795.htm). Acesso em: 2 dez. 2018.
- BRASIL. **Decreto nº 7.404, de 23 de dezembro de 2010**. Regulamenta a Lei no 12.305, de 2 de agosto de 2010, que institui a Política Nacional de Resíduos Sólidos, cria o Comitê Interministerial da Política Nacional de Resíduos Sólidos e o Comitê Orientador para a Implantação dos Sistemas de Logística Reversa, e dá outras providências. [S. l.], 23 dez. 2010. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/\\_Ato2007-2010/2010/Decreto/D7404.htm](http://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2010/Decreto/D7404.htm). Acesso em: 14 dez. 2018.
- BRASIL. **Lei nº 12.305, de 2 de agosto de 2010**. Institui a Política Nacional de Resíduos Sólidos; altera a Lei no 9.605, de 12 de fevereiro de 1998; e dá outras providências. [S. l.], 2 ago. 2010. Disponível em: [http://www.planalto.gov.br/ccivil\\_03/\\_Ato2007-2010/2010/Lei/L12305.htm](http://www.planalto.gov.br/ccivil_03/_Ato2007-2010/2010/Lei/L12305.htm). Acesso em: 14 dez. 2018.
- CORDOVA. **Introdução geral ao Cordova**. 2015. Disponível em: <https://cordova.apache.org/docs/en/latest/guide/overview/index.html>. Acesso em: 24 abr. 2020.
- GOIS, A. **Ionic Framework - Construa aplicativos para todas as plataformas mobile**. São Paulo: Casa do Código, 2017.
- GONÇALVES, E. **Desenvolvendo Aplicações Web com JSP, Servlets, JavaServerFaces, Hibernate, EJB 3 Persistence e Ajax**. Rio de Janeiro: Ciência Moderna Editora, 2007.
- HIBERNATE. Idiomatic persistence for Java and relational databases. 2020. Disponível em: <https://hibernate.org/orm/>. Acesso em: 26abr. 2020.



IBGE. **Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal: 2014/IBGE**. 2016. Disponível em:

<https://biblioteca.ibge.gov.br/visualizacao/livros/liv95753.pdf>. Acesso em: 15 dez. 2018.

IBGE. **Acesso à Internet e à televisão e posse de telefone móvel celular para uso pessoal 2017**. 2018. Disponível em:

<https://biblioteca.ibge.gov.br/visualizacao/livros/liv101631informativo.pdf>. Acesso em: 20 dez. 2018.

IONIC. **Visão geral do Ionic**. 2020. Disponível em: <https://ionicframework.com/docs>. Acesso em: 24 abr. 2020.

MELO, A. C. **Desenvolvendo aplicações com UML 2.2: do conceitual à implementação**. Rio de Janeiro: Brasport, 2010.

SOBRAL, M. F. F. Casos de dengue e coleta de lixo urbano: um estudo na cidade do Recife, Brasil. **Ciência e Saúde Coletiva**, scielo, v. 24, p. 1075 – 1082, 03 2019. ISSN 1413-8123.

Disponível em: <http://www.scielo.br/scielo.php?script=sciarttext&pid=S1413-81232019000301075&nrm=iso>. Acesso em: 15 ago. 2019.

SOFTWAREPUBLICO. O que é o Software Público. 2018. Disponível em:

<https://softwarepublico.gov.br/social/>. Acesso em: 02 jan. 2019.

SPRING. **Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can just run**. 2020. Disponível em:

<https://spring.io/projects/spring-boot>. Acesso em: 24 abr. 2020.

VIEIRA, P. C. Percepção e hábitos relacionados ao lixo doméstico entre moradores da comunidade do Coripós, Blumenau, SC. **Revista de APS**, v. 15, p. 82–91, 2012. Disponível em:

<https://periodicos.ufjf.br/index.php/aps/article/view/14779>. Acesso em: 03 ago. 2019.