

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO

APLICATIVO PARA RASTREAMENTO E
ACOMPANHAMENTO DA COLETA DE LIXO NOS
MUNICÍPIOS – RACL

HUDSON HENRIQUE LOPES

BLUMENAU
2017

HUDSON HENRIQUE LOPES

**APLICATIVO PARA RASTREAMENTO E
ACOMPANHAMENTO DA COLETA DE LIXO NOS
MUNICÍPIOS – RACL**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Marcel Hugo, Mestre - Orientador

**BLUMENAU
2017**

**APLICATIVO PARA RASTREAMENTO E
ACOMPANHAMENTO DA COLETA DE LIXO NOS
MUNICÍPIOS – RACL**

Por

HUDSON HENRIQUE LOPES

Trabalho de Conclusão de Curso aprovado
para obtenção dos créditos na disciplina de
Trabalho de Conclusão de Curso II pela banca
examinadora formada por:

Presidente: _____
Prof. Marcel Hugo, Mestre – Mestre e Orientador, FURB

Membro: _____
Prof(a). Luciana Pereira de Araújo, Mestre – FURB

Membro: _____
Prof. Alexander Roberto Valdameri, Mestre – FURB

Blumenau, 12 de dezembro de 2017.

Dedico este trabalho a todos que estiveram contribuindo de alguma forma para esta conquista, aos colegas, amigos, professores e em especial à minha família e a Deus.

AGRADECIMENTOS

A Deus pela minha existência, capacitação e por me conduzir ao longo de minha jornada.

À minha família que sempre me deu constante suporte e incentivo nesta graduação.

Aos meus amigos que me estimularam a não desistir e sempre procurar me aperfeiçoar profissionalmente.

Ao meu orientador, Marcel Hugo, pelo conhecimento compartilhado, conselhos e por aceitar de prontidão este desafio.

[...] Que nos salvou e chamou com uma santa vocação; não segundo as nossas obras, mas segundo o seu próprio propósito e graça que nos foi dada em Cristo Jesus, antes dos tempos dos séculos [...].

2 Timóteo 1.9 (N.V.I.)

RESUMO

Este trabalho apresenta o desenvolvimento de um aplicativo cliente para a plataforma Android, com intuito de possibilitar o acompanhamento e rastreamento das coletas de resíduos sólidos (lixo). O aplicativo consome informações como cadastros dos dispositivos de rastreamento e das rotas de coleta através de *web service*, além de receber a geolocalização compartilhada de dispositivos coletores. Sendo assim, é necessário que exista um servidor suportando o gerenciamento e envio destas informações. Os cadastros permitem a definição das rotas que serão apresentadas no mapa gerado para o aplicativo, a partir da API do Google Maps, e quais dispositivos compartilham suas geolocalizações no momento em que ocorrem as coletas. Este compartilhamento ocorre através do envio da posição do dispositivo por meio de *web service* para o servidor auxiliar, que disponibiliza aos demais dispositivos nos quais o aplicativo esteja requisitando tal informação. O aplicativo foi desenvolvido em JAVA através da IDE Android Studio, e consiste tanto no compartilhamento da geolocalização do dispositivo de coleta quanto no recebimento desta localização e visualização das rotas definidas para o mapa. Além disso, um módulo web auxiliar foi desenvolvido para a realização dos cadastros e criados os *web services* que realizam a comunicação deste módulo e do aplicativo para com o servidor de aplicação.

Palavras-chave: Recolhimento de resíduos. Coleta de lixo. Android. Google maps.

ABSTRACT

This work presents the development of a client application for the Android platform, in order to enable tracking and monitoring of garbage collections. The application consumes information such as the logs of the tracking devices and the collection routes through web service, in addition to receiving the shared geolocation of collecting devices. It is necessary to have a server computer supporting management and sending information. The registries allow you to define which routes will appear on the map generated for the application by Google Maps API, and which devices share their geolocations at the time the collections occur. This sharing occurs by sending the position of the device through the web service to the helper server, which makes available to the other devices requesting such information. The application was developed in JAVA through the IDE Android Studio, and consists of both the geolocation sharing of the collection device and the receipt of this location and visualization of the routes defined for the map. In addition, an auxiliary web module was developed for a realization of the registrations and also created the web services that perform communication of this module and the application to the application server.

Key-words: Waste collection. Garbage collection. Android. Google Maps.

LISTA DE FIGURAS

Figura 1 – Google Maps para Android	18
Figura 2 – Representação dos satélites para funcionamento do GPS	20
Figura 3 – Funcionalidade de atendimento de ocorrências, no Coletas RCC Transportador ..	21
Figura 4 – Consulta de informações sobre coletas, no VanCollect	22
Figura 5 – Detalhamento das informações, no Coleta Urbana	23
Figura 6 – Funcionalidade de definição do endereço para pesquisa da coleta e rota para entrega voluntária, no Limpa Rápido	24
Figura 7 – Diagrama de casos de uso do aplicativo.....	29
Figura 8 - Diagrama de atividades do aplicativo	31
Figura 9 - Diagrama de casos de uso do módulo web	32
Figura 10 - Diagrama de atividades do módulo web	33
Figura 11 - Tela inicial do aplicativo RACL.....	42
Figura 12 – Telas durante o processo de autenticação, no aplicativo RACL.....	43
Figura 13 – Telas de filtro no modo Consulta, do aplicativo RACL.....	44
Figura 14 – Tela de detalhamento da rota, no aplicativo RACL.....	45
Figura 15 – Simulando consulta de rotas por endereço	48
Figura 16 - Simulando rastreamento do dispositivo	48
Figura 17 - Simulando acompanhamento do rastreo	48
Figura 18 - Cadastro de dispositivos, módulo web de apoio	64
Figura 19 - Cadastro de rotas, módulo web de apoio	64

LISTA DE QUADROS

Quadro 1 – Comparativo de principais características dos correlatos	25
Quadro 2 - Requisitos funcionais do aplicativo	27
Quadro 3 - Requisitos não funcionais do aplicativo	27
Quadro 4 - Requisitos funcionais do módulo web de apoio	28
Quadro 5 - Método drawLinesRoute para desenho da rota no mapa.....	35
Quadro 6 – Método createCustomMarkerOption para customização do marcador	35
Quadro 7 – Método addMarkerToMap para inclusão do marcador no mapa.....	36
Quadro 8 – Método getUrlGoogleDirectionsAPI para adequação da rota	37
Quadro 9 - Parâmetros para Google Directions API	38
Quadro 10 – Método runGPSPosition verifica e solicita a habilitação do GPS e compartilha a localização no dispositivo no modo Rastreo	39
Quadro 11 - Método getJSONFromAPI para comunicação de web services	40
Quadro 12 - Principais características dos correlatos em comparação ao RACL	46
Quadro 13 - Detalhamento do caso de uso Selecionar modo de uso.....	54
Quadro 14 – Detalhamento do caso de uso Encerrar rastreo.....	54
Quadro 15 – Detalhamento do caso de uso Consultar coletas por cidade	55
Quadro 16 – Detalhamento do caso de uso Consultar coletas por endereço.....	55
Quadro 17 – Detalhamento do caso de uso Visualizar informações das coletas	55
Quadro 18 – Detalhamento do caso de uso Visualizar rotas de coleta em mapa	56
Quadro 19 – Detalhamento do caso de uso Acompanhar execução da coleta em tempo real no mapa.....	56
Quadro 20 – Detalhamento do caso de uso Manter dispositivos de coleta.....	56
Quadro 21 – Detalhamento do caso de uso Manter rotas de coleta.....	56
Quadro 22 – Detalhamento do caso de uso Manter agendamentos das coletas na rota.....	57
Quadro 23 – Detalhamento do caso de uso Manter dispositivos nos agendamentos	57

Quadro 24 - Detalhamento do caso de uso Manter pontos a serem percorridos	57
Quadro 25 - Resultado da Google Directions API	58
Quadro 26 - Resultado esperado cadastro de dispositivo do módulo web.....	61
Quadro 27 - Resultado esperado cadastro de rota do módulo web	61

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface

GPS – Global Positioning System

IDE – Integrated Development Enviroment

JSON – JavaScript Object Notation

RACL – Aplicativo para Rastreamento e Acompanhamento da Coleta de Lixo

RF – Requisito Funcional

RNF – Requisito Não Funcional

UC – Use Case

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVOS	15
1.2 ESTRUTURA.....	16
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 RASTREABILIDADE E GEOLOCALIZAÇÃO	17
2.2 GOOGLE MAPS	18
2.3 GPS EM ANDROID	19
2.4 TRABALHOS CORRELATOS	20
2.4.1 Coletas RGG.....	21
2.4.2 VanCollect.....	21
2.4.3 Coleta Urbana.....	22
2.4.4 Limpa Rápido	23
2.4.5 Comparativo entre os correlatos	24
3 DESENVOLVIMENTO DO APLICATIVO	26
3.1 LEVANTAMENTO DE INFORMAÇÕES	26
3.2 REQUISITOS	27
3.2.1 Requisitos do aplicativo	27
3.2.2 Requisitos do módulo web de apoio	28
3.3 ESPECIFICAÇÃO.....	28
3.3.1 Diagrama de casos de uso do aplicativo	28
3.3.2 Diagrama de atividades do aplicativo	30
3.3.3 Diagramas complementares para o módulo web auxiliar	31
3.4 IMPLEMENTAÇÃO	33
3.4.1 Técnicas e ferramentas utilizadas	33
3.4.2 Operacionalidade da implementação	41
3.5 RESULTADOS E DISCUSSÕES	45
3.5.1 Comparação com os trabalhos correlatos em relação ao RAACL	46
3.5.2 Validação do aplicativo.....	47
4 CONCLUSÕES.....	49
4.1 EXTENSÕES	49
REFERÊNCIAS	51

APÊNDICE A – DETALHAMENTO DOS CASOS DE USO	54
APÊNDICE B – EXEMPLO DE RESULTADO DA GOOGLE DIRECTIONS API	58
APÊNDICE C – EXEMPLOS DE RESULTADOS DO MÓDULO WEB DE APOIO...	61
APÊNDICE D – TELAS DE CADASTRO DO MÓDULO WEB DE APOIO.....	64

1 INTRODUÇÃO

Nos últimos anos tem ocorrido um grande aumento no consumo de bens materiais. Em 2013 para os brasileiros, segundo a Fiesp (2014, p. 1), “pelo menos R\$ 50 bilhões foram destinados ao consumo de móveis e artigos do lar”. A ascensão social dos últimos anos, por exemplo, impulsionou tanto o aumento do consumo, quanto a geração de lixo e descarte incorreto do mesmo (TRIGUEIRO, 2013, p. 1). O consumo exagerado de bens não reutilizáveis e/ou, até mesmo, o descarte incorreto dos recicláveis é um grande problema da geração atual e das futuras.

As consequências geradas pelo crescimento exponencial da poluição são cada vez mais visíveis. Em locais em que a coleta não é realizada regularmente, o lixo é depositado em locais impróprios (encostas, rios e córregos) gerando dispersão de insetos e pequenos animais hospedeiros de certas doenças. Além disso, ocorre a produção do chorume que contamina solo, ser humano e a biodiversidade (FREITAS, 2008, p. 1). O lixo produzido pelo ser humano pode também liberar gases para a atmosfera, abarrotar bueiros, ocasionar enchentes, poluir rios destruindo a flora e fauna, bem como permitir que animais espalhem os restos – em busca de comida – pelas ruas da cidade.

Ao longo dos anos houve uma evolução no serviço de coleta de resíduos, porém ainda persistem práticas de descarte inadequadas e prejudiciais ao meio ambiente, sendo essas a queima, enterramento ou simplesmente jogar fora os resíduos (PAZ; MORAIS, 2014, p. 91). O descarte dos resíduos sólidos precisa ser realizado nos devidos locais e nos períodos corretos pelos cidadãos. O depósito do lixo ao céu aberto, sem ser encaminhado para o devido tratamento, libera o gás metano gerado a partir da decomposição do material orgânico, além do chorume, que é um líquido de cor preta gerado também pela decomposição e acúmulo de água (ALVES; MOURA, 2014, p. 53).

Conforme afirma Miranda e Steuer (2014, p. 59), “a sociedade necessita se conscientizar de que o modelo vigente de crescimento e desenvolvimento afeta todos do planeta, sendo necessário reduzir o impacto ambiental”, de forma que tanto os cidadãos quanto a própria sociedade visualizem a importância do meio ambiente e o impacto causado a este pelo desenvolvimento. Desta forma, pequenas iniciativas podem gerar uma reeducação no futuro do planeta e a tecnologia pode auxiliar neste processo.

Procurando facilitar a participação dos cidadãos no processo de melhoria do recolhimento de resíduos, certas cidades no Brasil optaram por disponibilizar a consulta de algumas informações sobre as coletas, como locais e frequência em que estas ocorrem,

utilizando-se de sistemas e aplicativos. A prefeitura de Juiz de Fora, por exemplo, lançou um aplicativo visando melhorar a coleta de lixo, permitindo a localização e previsão dos caminhões de coleta próximos ao aplicativo do usuário, através de um mapa com o trajeto e apresentação dos horários (PESSÔA, 2016, p. 1). Contudo, de acordo com a Google Play (2016, p. 1), “o aplicativo está [ainda] na versão Beta e será melhorado dia após dia a fim de enriquecer a experiência do usuário além de entregar informações cada vez mais completas e precisas a respeito da coleta de lixo em Juiz de Fora”. Outra iniciativa que tem se destacado é o lançamento do aplicativo Limpa Rápido, em São Paulo (PREFEITURA DE SÃO PAULO, 2017, p.1). O aplicativo, segundo a prefeitura, realiza toda a integração dos serviços de limpeza da cidade, permitindo desde a visualização dos veículos da coleta de resíduos no momento real de seu percurso até mesmo a denúncia de descarte irregular do lixo.

O interesse pelo uso da tecnologia para a evolução no processo de coleta de resíduos e a corrente conscientização da responsabilidade dos cidadãos para com o descarte correto do lixo, indica a necessidade de acessibilidade a aplicativos para acompanhamento de coletas. Não estritamente a alguns municípios, mas, de forma mais simples, ao maior número de regiões do país. Um aplicativo que pudesse receber informações de cadastros externos, mantendo uma mesma interface e apresentando tanto horários quanto localização – dos veículos de coleta no momento em que estas coletas estão sendo realizadas – poderia aumentar a aderência de cidadãos ao uso de aplicativos para esse fim. O descarte realizado próximo ao horário de coleta poderia diminuir o risco de decomposição dos resíduos devido às variadas condições climáticas ou, até mesmo, extravio dos recipientes.

1.1 OBJETIVOS

Este trabalho tem como objetivo principal o desenvolvimento de um aplicativo móvel para a plataforma Android que permita a consulta da programação das coletas de lixo e a localização em que estas estão sendo realizadas nas cidades.

Os objetivos específicos deste trabalho são:

- a) disponibilizar uma interface para o usuário consultar os horários de coletas;
- b) visualizar as rotas das coletas de resíduos através de mapas e acompanhar a localização dos coletores;
- c) compartilhar a localização dos dispositivos de coletores;
- d) desenvolver *web services* para integrar o rastreamento com o acompanhamento;
- e) criar uma aplicação web para apoiar o cadastramento de rotas e seus horários.

1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos, sendo estes: introdução, fundamentação teórica, desenvolvimento e conclusão.

O primeiro capítulo introduz a temática, elencando os objetivos e explanando a forma como o trabalho está estruturado.

O segundo capítulo apresenta a fundamentação teórica, conceituando os elementos necessários para o desenvolvimento deste trabalho. As tecnologias utilizadas e os trabalhos correlatos também se encontram neste capítulo.

O desenvolvimento da aplicação é demonstrado no terceiro capítulo. Além disso, são descritos os requisitos, sua especificação através de diagramas e o detalhamento das funcionalidades. Por fim, apresenta-se os resultados obtidos.

O quarto e último capítulo apresenta as conclusões obtidas com o trabalho e possíveis extensões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo está dividido em 4 seções. A primeira seção, 2.1, apresenta a intensificação do uso da rastreabilidade nas empresas e o objetivo ao qual tem sido aplicada. Contrasta o rastreo direcionado ao envio de produtos com o rastreo de coleta (recebimento), conceituando e contextualizando a importância da geolocalização nestes processos. Na seção 2.2 é descrita a ferramenta de mapas da Google. Em sequência, apresenta-se o sistema GPS na seção 2.3, e sua utilização para desenvolvimento na plataforma Android. Por fim, na seção 2.4 são apresentados os trabalhos correlatos e sua comparação.

2.1 RASTREABILIDADE E GEOLOCALIZAÇÃO

Os empresários têm diversas dificuldades na distribuição de seus produtos, pois precisam se adequar às leis e regras de transporte, começando pelo recebimento da matéria-prima e seguindo até a entrega final do produto. Essa dificuldade se intensifica mais ainda para pequenas e médias empresas que não possuem um sistema de rastreabilidade bem definido e atrelado à sua logística. Desta forma, a rastreabilidade assume um papel importante nas empresas, proporcionando um diferencial e a facilidade na localização dos produtos ofertados (SANSON, 2014, p. 1).

Em logística, a rastreabilidade refere-se à capacidade para localizar produtos ao longo da cadeia de distribuição. Já em outras atividades, a rastreabilidade inclui a cadeia de produção, juntamente com a identificação única de todos os produtos e itens de materiais, necessários para criar ou construir o bem final. Na indústria automotiva, torna possível um recall, e, na indústria alimentar, contribui para a segurança alimentar (LOVIS, 2008, p. 105, tradução nossa).

Durante a construção de sua dissertação acerca da rastreabilidade na indústria de rolhas, Pereira (2014, p. 26) conclui que “A implementação de sistemas de rastreabilidade é preponderante para o sucesso das organizações”. Tendo esse entendimento, as empresas têm cada vez mais aderido ao uso da rastreabilidade proporcionada pelas novas tecnologias que, fazendo uso de sistemas adequados, as posicionam à frente do mercado gerando confiabilidade aos consumidores (FOCO RURAL, 2016, p.1).

O presente trabalho utiliza um foco diferente para a rastreabilidade: não para produtos, mas sim para serviços. Ao invés da entrega, sua necessidade é a coleta; coleta de resíduos sólidos. Mas para que possa atingir seu potencial, é necessário o compartilhamento da localização dos executores das coletas. Para isso, serão explorados os benefícios da geolocalização.

Geolocalização refere-se à identificação da localização geográfica do usuário utilizador ou de um dispositivo, através de uma variedade de mecanismos de recolhimento de dados. Normalmente, a maioria dos serviços de geolocalização

usam endereços de roteamento da rede ou dispositivos internos de GPS para determinar este local (GOOGLE MAPS APIS, 2017a, p.1).

A integração da rastreabilidade e o compartilhamento da localização dos dispositivos com o aplicativo desenvolvido, tornam possível o acompanhamento da realização de coletas dos resíduos nos municípios, quando estes estiverem ocorrendo.

2.2 GOOGLE MAPS

O Google Maps é uma ferramenta desenvolvida pela Google, que permite procurar, explorar e encontrar praticamente qualquer local desejado no mundo, através de um mapa. Quanto mais utilizado, mais ele se ajusta às preferências (GOOGLE MAPS HELP, 2016, p. 1). Os mapas da ferramenta estão disponíveis para dispositivos web e móvel, podendo variar quanto aos seus modelos de apresentação, conforme pode-se verificar na Figura 1.

Figura 1 – Google Maps para Android



Fonte: Google Maps APIs (2017b, p. 1).

Dentre as funcionalidades disponíveis, pode-se citar a construção de rotas de trajeto, orientação de navegação GPS, informações sobre o trânsito, modos diferenciados de visualização, além de outras características apresentadas durante sua utilização nos aplicativos

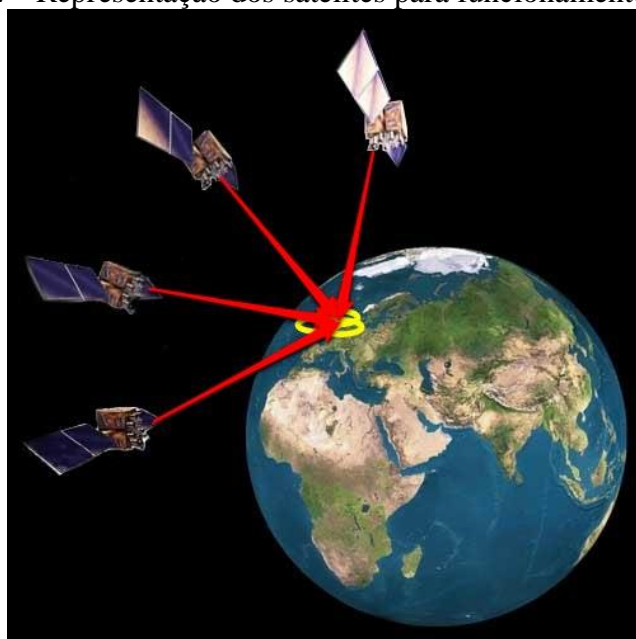
(GOOGLE, 2017, p. 1). O uso dessa ferramenta tem expandido as possibilidades para construção de aplicativos e aumentado a interação dos usuários.

2.3 GPS EM ANDROID

Visando possuir um sistema de alta exatidão e tecnologia que pudesse orientar o exército durante suas jornadas, o Departamento de Defesa dos Estados Unidos desenvolveu um sistema de rádio navegação popularmente conhecido como GPS. Contudo, devido a suas características, a sua abrangência não ficou restrita apenas ao uso militar, despertando seu uso em aplicações para o uso civil (ICPD, 2015, p. 54).

De acordo com o ICPD (2015, p. 54) “a concepção do sistema [GPS] permite que um usuário, em qualquer local da superfície terrestre, tenha à sua disposição, no mínimo, quatro satélites que podem ser rastreados. Este número de satélites permite o posicionamento em tempo real [...]. Além disto, o GPS pode ser usado sob quaisquer condições climáticas”. Essa concepção pode ser visualizada na Figura 2. O princípio da navegação desse sistema procura verificar a distância entre a antena do dispositivo do usuário em relação a três dos quatro satélites – um destes, é necessário apenas para a sincronização dos relógios. Há dois tipos de serviços existentes: o Standard Positioning Service (SPS) que é de acesso liberado e gratuito a todos usuários do planeta, tendo sua exatidão horizontal e vertical em torno de 100 e 140 m (respectivamente), e 340 nano segundos nas medidas de tempo; enquanto o Precise Positioning Service (PPS) é de acesso restrito, podendo alcançar níveis extremos de exatidão, mas se limitando 10 a 20m por motivos de segurança já que é um sistema global (ICPD, 2015, p.14-15).

Figura 2 – Representação dos satélites para funcionamento do GPS



Fonte: Griffin (2011, p.1).

Com o avanço da tecnologia, os smartphones já têm sido oferecidos com o sistema GPS integrado. A plataforma Android permite o desenvolvimento de aplicações móveis fazendo uso desse sistema, no qual o suporte aos serviços de localização ocorre através das classes disponíveis no pacote `android.location`. Para isso, as APIs que determinam a localização são centralizadas no sistema de serviço `LocationManager`. Ele não pode ser instanciado diretamente – assim como outros serviços da Google, é necessário que seja solicitada uma instância através do método `getSystemService(String name)`. Este método recebe o nome do serviço desejado e, para localização, utiliza-se o parâmetro `Context.LOCATION_SERVICE` (ANDROID DEVELOPERS, 2017, p.1).

2.4 TRABALHOS CORRELATOS

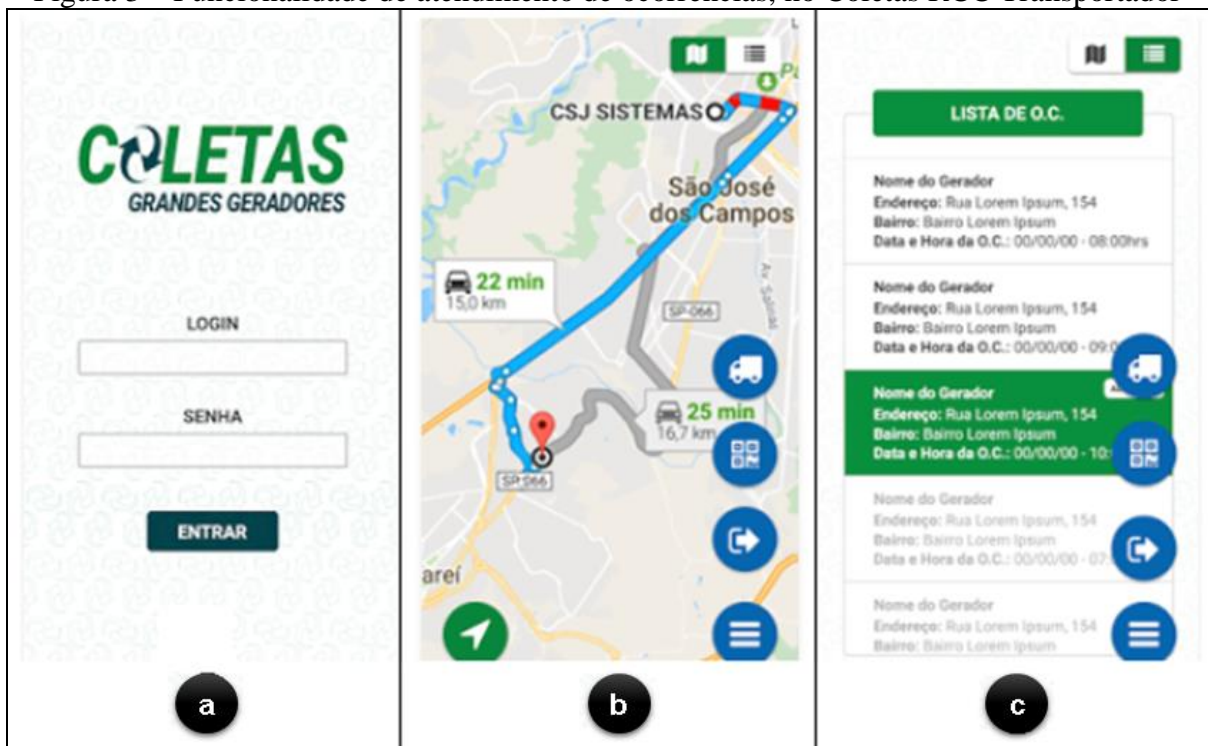
Para a plataforma Android existem diversos aplicativos que tem como funcionalidade o acompanhamento e rastreamento de dispositivos. A seguir são apresentados os trabalhos que possuem características e funcionalidades similares ao trabalho desenvolvido. O aplicativo Coletas RGG Transportador (GOOGLE PLAY, 2017a, p.1) utilizou o Google Maps para construção do mapa e desenho das rotas para transportadoras. O aplicativo VanCollect (GOOGLE PLAY, 2015, p.1) permite a digitação do local a pesquisar coletas e o detalhamento das respectivas informações. O aplicativo Coleta Urbana (GOOGLE PLAY, 2016, p. 1) pesquisa e apresenta as informações de coletas como horários, mapas e posição do caminhão coletor. Por fim, o aplicativo Limpa Rápido (GOOGLE PLAY, 2017b, p. 1) integra

visualização de informações sobre coletas com o acompanhamento de suas execuções em tempo real.

2.4.1 Coletas RGG

O Coletas RGG Transportador foi disponibilizado pelo Coletas Online. É um aplicativo direcionado a transportadoras de resíduos de grandes geradores, com intuito de facilitar o gerenciamento de ocorrências e acompanhar o atendimento destas. Apesar de ser um aplicativo gratuito, seu acesso é restrito por *login* (Figura 3a). Ele apresenta uma lista ao usuário contendo informações sobre os grandes geradores de resíduos com data e localização em que foi registrada a ocorrência vigente (Figura 3c). Selecionando uma destas ocorrências é possível visualizar o melhor percurso para execução da coleta e o devido transporte dos resíduos, calculado a partir do tempo e distância necessários (Figura 3b). Esta funcionalidade pode ser visualizada na Figura 3 (GOOGLE PLAY, 2017a, p.1).

Figura 3 – Funcionalidade de atendimento de ocorrências, no Coletas RCC Transportador



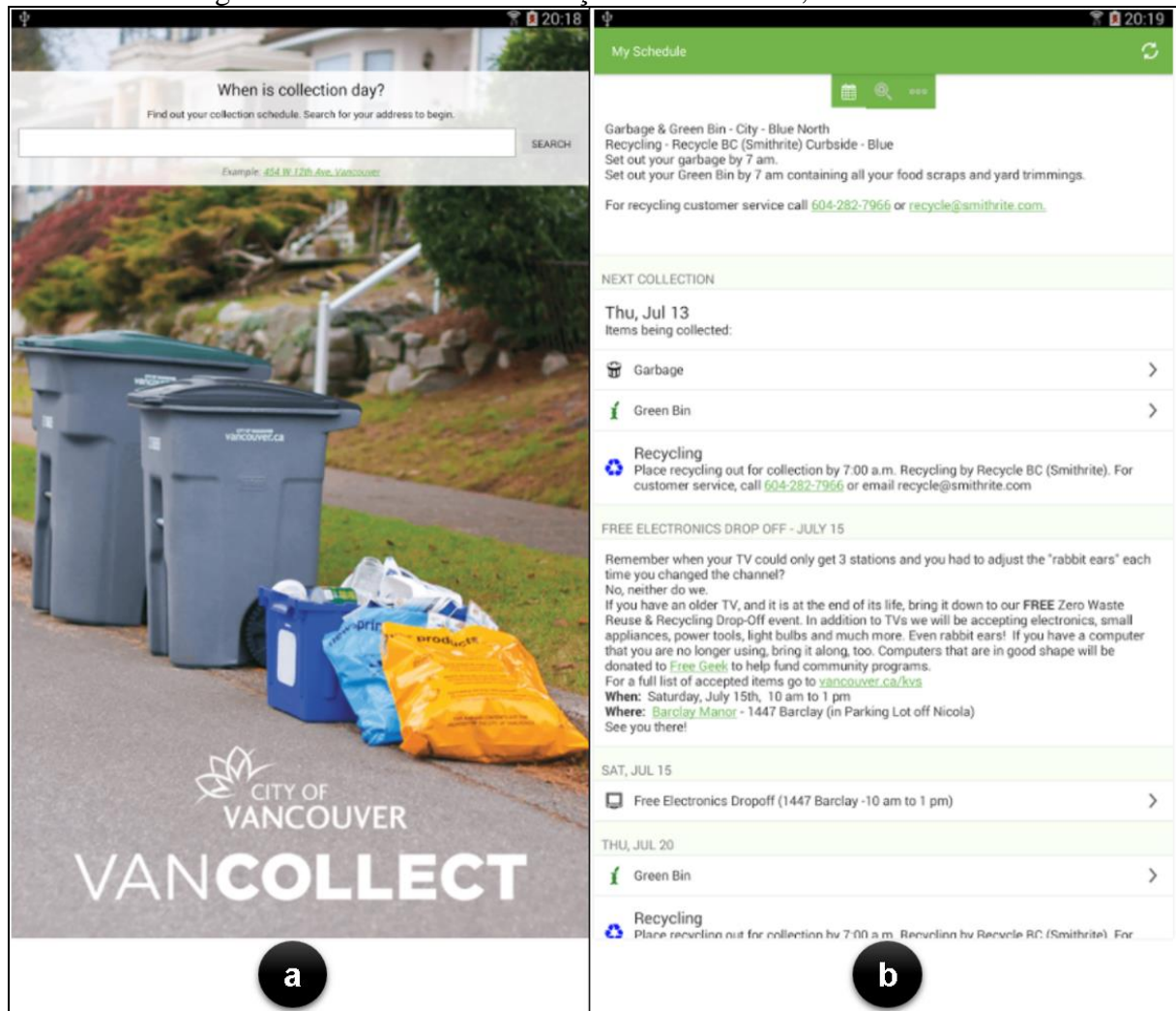
Fonte: Google Play (2017a, p. 1).

2.4.2 VanCollect

O aplicativo VanCollect foi desenvolvido especificamente para a cidade de Vancouver. Ele visa disponibilizar informações sobre as coletas de lixo e seus respectivos horários. Permite a criação de lembretes semanais dentro da aplicação e compartilha informações sobre o correto descarte de resíduos (GOOGLE PLAY, 2015, p.1). Ao informar o endereço

desejado (Figura 4a) é apresentada uma tela com informações sobre as coletas agendadas, desde informações como local e horário, além do planejamento para descarte de elementos específicos, como eletrônicos (Figura 4b).

Figura 4 – Consulta de informações sobre coletas, no VanCollect



Fonte: Google Play (2015, p. 1).

2.4.3 Coleta Urbana

O aplicativo Coleta Urbana foi disponibilizado pela Descarte.me e Demlurb para oferecer a possibilidade de acompanhar a coleta de resíduos domiciliares na cidade de Juiz de Fora. Ele permite que seja definido um endereço para pesquisa das coletas (Figura 5b), seus horários (Figura 5c) e o acompanhamento do percurso a ser executado (Figura 5d), assim como sua efetiva execução em tempo real (GOOGLE PLAY, 2016, p.1).

Figura 5 – Detalhamento das informações, no Coleta Urbana



Fonte: Google Play (2016, p. 1).

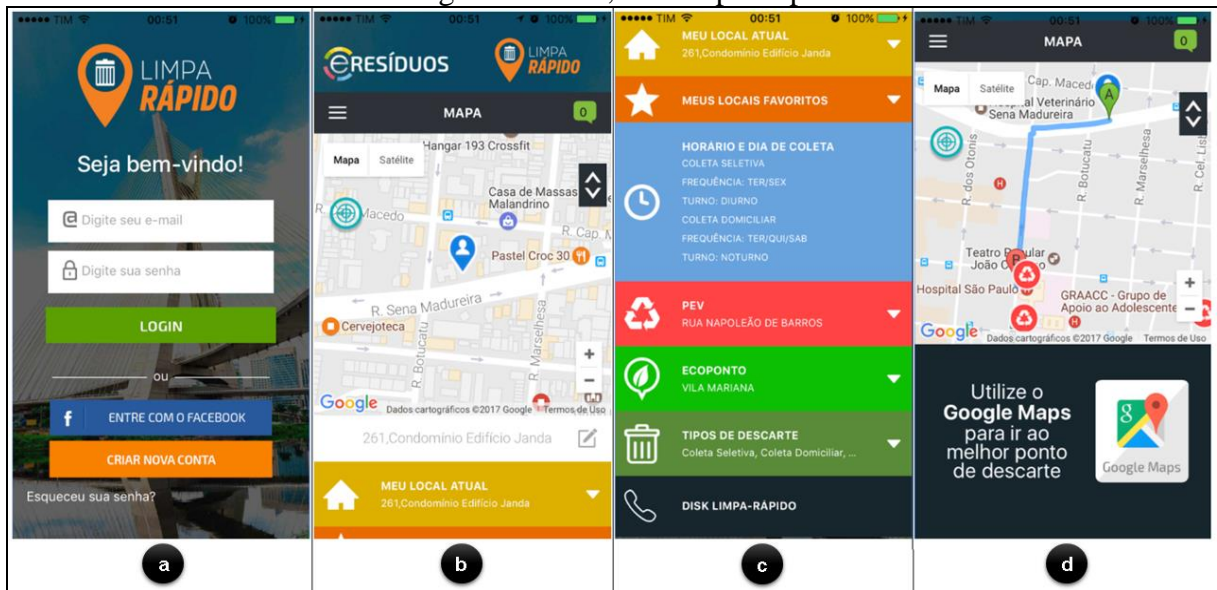
Segundo G1 Zona da Mata (2016, p.1), o aplicativo possibilita “que os resíduos fiquem menos tempo nas calçadas, evitando obstrução da passagem, mau cheiro e que animais espalhem os resíduos”. A coleta é realizada pelo menos três vezes por semana nas regiões da cidade, tanto no período diurno quanto noturno (G1 ZONA DA MATA, 2016, p. 1).

2.4.4 Limpa Rápido

O aplicativo Limpa Rápido foi desenvolvido direcionado para a cidade de São Paulo. É um aplicativo para dispositivos móveis que centraliza informações acerca dos serviços urbanos de limpeza da cidade. Possui diversas funcionalidades que auxiliam os moradores a participarem de forma mais direta e efetiva na manutenção e entendimento dos serviços prestados. Seu acesso é por meio de *login* criado pelo próprio usuário (Figura 6a).

O acompanhamento da localização de caminhões de coleta em tempo real, tanto domiciliar quanto seletiva, é disponibilizado aos moradores para que possam confirmar a execução da coleta na sua região. Inicialmente é apresentada a localização do dispositivo do usuário (Figura 6b) e pode-se salvar os endereços mais acessados no aplicativo (Figura 6c). Um mapa permite que esse acompanhamento seja realizado e a identificação de pontos específicos, bem como a rota, para entrega voluntária. Além disso, têm disponíveis horários e dias em que cada tipo de coleta é executada (Figura 6d). O aplicativo ainda notifica mutirões de limpeza e ações ambientais realizada no bairro. Estatísticas, boas práticas sustentáveis e notícias sobre reciclagem também são compartilhadas no aplicativo (GOOGLE PLAY, 2017b, p.1).

Figura 6 – Funcionalidade de definição do endereço para pesquisa da coleta e rota para entrega voluntária, no Limpa Rápido



Fonte: Google Play (2017b, p. 1).

2.4.5 Comparativo entre os correlatos

O Quadro 1 apresenta um comparativo entre as principais características dos trabalhos correlatos apresentados.

Os aplicativos utilizam a mesma tecnologia de localização, sendo ela o serviço de GPS. O armazenamento de suas informações ocorre através de carregamento USB e também do acesso ao que já havia sido armazenado no dispositivo. Dentre os correlatos apresentados, apenas dois deles utilizam autenticação por *login*, sendo que somente um deles permite que o cadastro seja realizado por qualquer tipo de usuário. As funcionalidades variam entre lista de informações, tela de detalhamento e apresentação de mapa para desenho de rotas e visualização de rastreamento. Todos necessitam de acesso à internet.

Quadro 1 – Comparativo de principais características dos correlatos

Trabalhos relacionados/ Características	Coletas RGG Transportador (GOOGLE PLAY, 2017a, p.1)	VanCollect (GOOGLE PLAY, 2015, p.1)	Coleta Urbana (GOOGLE PLAY, 2016, p.1)	Limpa Rápido (GOOGLE PLAY, 2017b, p.1)
Tecnologia de localização	GPS e acesso à internet	GPS e acesso à internet	GPS e acesso à internet	GPS e acesso à internet
Armazenamento	USB e acessa o que foi armazenado	USB e acessa o que foi armazenado	Não informado	USB e acessa o que foi armazenado
Autenticação	Login	Não informado	Não informado	Login
Apresentação de informações	Mapa e lista de ocorrências	Tela de detalhamento	Tela de detalhamento, mapa e localização em tempo real	Tela de detalhamento, mapa e localização em tempo real
Adicionais	Coleta informações e tem acesso à internet, interpreta a configuração do servidor Google, não permite o dispositivo entrar em Stand-by	Coleta informações e tem acesso à internet, interpreta a configuração do servidor Google, não permite o dispositivo entrar em Stand-by	Coleta informações e tem acesso à internet, interpreta a configuração do servidor Google	Coleta informações e tem acesso à internet, interpreta a configuração do servidor Google, controla vibração do dispositivo, não permite o dispositivo entrar em Stand-by

Fonte: elaborado pelo autor.

3 DESENVOLVIMENTO DO APLICATIVO

Neste capítulo são descritas as funcionalidades do aplicativo, listando seus requisitos, apresentando diagramas e capturas de tela, além de trechos da implementação. São citadas também as informações necessárias para integração do aplicativo a um módulo web auxiliar.

3.1 LEVANTAMENTO DE INFORMAÇÕES

Este trabalho apresenta o desenvolvimento de um aplicativo para sistema Android que permita o acompanhamento e rastreabilidade das coletas de lixo programadas, bem como as que estão sendo realizadas em tempo real. O aplicativo deve localizar geograficamente o prestador de serviços de coleta e, ao mesmo tempo, auxiliar o usuário que utiliza o serviço para que possa encaminhar os lixos e resíduos em um horário próximo à coleta de seu domínio.

Para cadastro e gerenciamento das informações sobre coletas há um módulo web desenvolvido apenas para suporte do aplicativo Android. Neste é possível que a empresa coletora de resíduos realize o cadastro dos municípios, turnos e trajetos das coletas, além do cadastro dos dispositivos de usuários-coletores (caminhão do lixo, catadores, outros) que possuem o aplicativo, compartilhando assim suas respectivas localizações. Para cada dispositivo cadastrado há um código para que a funcionalidade de rastreio seja habilitada.

O aplicativo não depende de autenticação por *login* (usuário e senha). Para consultas o acesso é livre e para ativação do modo de rastreio no dispositivo (prestador de serviços de coleta) é necessária a informação do código de autenticação do dispositivo gerado a partir do cadastro realizado no módulo web.

No modo de consulta, o usuário pode definir filtros para visualização das coletas programadas. Esses filtros podem ser por município, trajeto (vide rua) e turno (vide horários). Estas informações selecionadas para a consulta por filtros são provenientes dos cadastros realizados no módulo web.

Para o modo de rastreio, o usuário-coletor deve informar o código de autenticação do dispositivo, disponibilizado no cadastro do dispositivo no módulo web, gerado pela empresa responsável pela coleta. Após isso, é possível a ativação (e posteriormente, desativação) do rastreio do dispositivo. O dispositivo é atrelado a município, trajetos e turnos. Quando o rastreio do dispositivo estiver ativo, fornecerá as coordenadas (via *web service*) nas consultas de coletas, permitindo o acompanhamento em tempo real do serviço, por meio do Google Maps no Android. O mapa será gerado conforme o filtro de consulta e, pelas informações contidas no módulo web. Para dispositivos que não estão cadastrados no módulo web e o

usuário escolher utilizar o modo de rastreo, é notificada a necessidade de realização do cadastro deste dispositivo para tal funcionalidade.

3.2 REQUISITOS

A apresentação dos Requisitos Funcionais (RF) e Não Funcionais (RNF) do aplicativo ocorre nesta seção. Também são apresentados os Requisitos Funcionais do módulo Web auxiliar (RFW) responsável pelo compartilhamento dos cadastros com o aplicativo.

3.2.1 Requisitos do aplicativo

As características requeridas para funcionamento do aplicativo são apresentadas no Quadro 2. As funcionalidades descritas abrangem tanto o usuário que consulta as informações de coletas quanto o usuário que efetivamente executa a coleta de lixo e podem ser visualizadas no diagrama de caso de uso da Figura 7.

Quadro 2 - Requisitos funcionais do aplicativo

Requisitos Funcionais	Caso de Uso
RF01: O aplicativo deve permitir selecionar um modo de uso: “Consulta” ou “Rastreo”.	UC01
RF02: O aplicativo deve permitir encerrar a rastreabilidade para usuários-coletores autenticados, no modo “Rastreo”.	UC02
RF03: O aplicativo deve permitir a consulta das coletas programadas por cidade, no modo “Consulta”.	UC03
RF04: O aplicativo deve permitir a consulta das coletas programadas por endereço, no modo “Consulta”.	UC04
RF05: O aplicativo deve permitir a visualização das informações de coletas, no modo “Consulta”.	UC05
RF06: O aplicativo deve permitir a visualização das rotas de coletas através de um mapa, no modo “Consulta”.	UC06
RF07: O aplicativo deve permitir o acompanhamento da coleta sendo executada em tempo real, no modo “Consulta”.	UC07

Fonte: elaborado pelo autor.

O Quadro 3 lista os requisitos não funcionais previstos para o aplicativo.

Quadro 3 - Requisitos não funcionais do aplicativo

Requisitos Não Funcionais
RNF01: O aplicativo deve ser desenvolvido em linguagem Java.
RNF02: O aplicativo deve ser desenvolvido utilizando a ferramenta Android Studio.
RNF03: O aplicativo deve ser compatível com o sistema operacional Android.
RNF04: O aplicativo deve apresentar as rotas definidas através do Google Maps API v2 para Android.
RNF05: O aplicativo deve receber as informações cadastrais através de <i>web service</i> .
RNF06: O aplicativo deve atualizar os posicionamentos dos coletores através de <i>web service</i> .

Fonte: elaborado pelo autor.

3.2.2 Requisitos do módulo web de apoio

Para o funcionamento do aplicativo é necessário que informações da gestão da coleta estejam cadastrados, como rotas e horários. Isto foi atendido pelo desenvolvimento de um módulo web de apoio, que disponibiliza o acesso a cadastros de dispositivos de usuários coletores e rotas percorridas pela cidade na execução da coleta de lixo. Os requisitos deste módulo são apresentados no Quadro 4.

Quadro 4 - Requisitos funcionais do módulo web de apoio

Requisitos Funcionais	Caso de Uso
RFW01: O módulo web deve manter dispositivos de coleta.	UCW01
RFW02: O módulo web deve manter rotas.	UCW02
RFW03: O módulo web deve manter agendamentos de coletas na rota.	UCW03
RFW04: O módulo web deve manter dispositivos nos agendamentos de coleta da rota.	UCW04
RFW05: O módulo web deve manter pontos a serem percorridos na rota.	UCW05

Fonte: elaborado pelo autor.

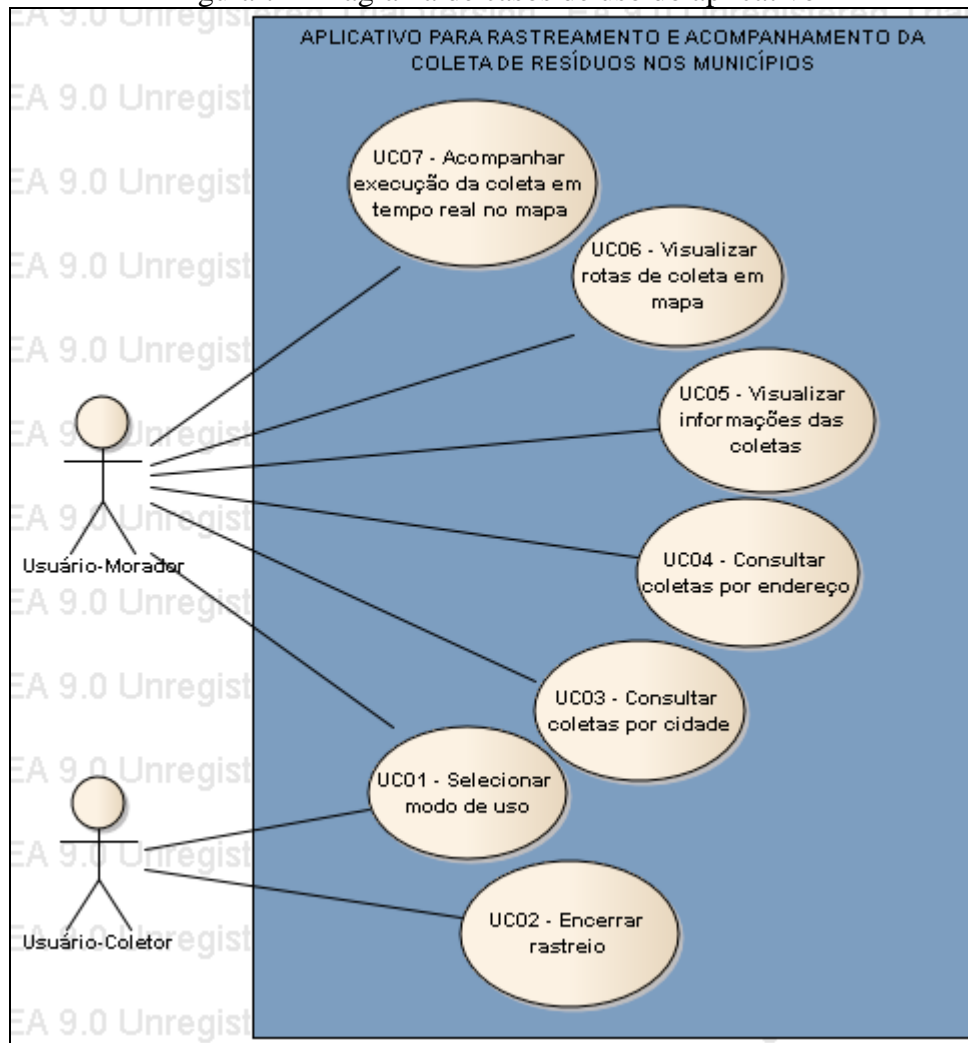
3.3 ESPECIFICAÇÃO

Esta seção apresenta o conjunto de elementos que compõem a especificação total do aplicativo. São apresentados os diagramas de casos de uso descrevendo as funcionalidades disponibilizadas aos usuários morador e coletor, assim como o fluxo do módulo web auxiliar. Como o aplicativo opera através do recebimento dos dados já filtrados pelo servidor, não é apresentada a modelagem de dados. Contudo, no Apêndice C são exemplificados os JSONs esperados para utilização no aplicativo.

3.3.1 Diagrama de casos de uso do aplicativo

O diagrama de casos de uso do aplicativo está representado na Figura 7 e, em sequência, seu detalhamento. Este é composto de 7 casos de uso, contendo dois atores: Usuário-Morador e Usuário-Coletor.

Figura 7 – Diagrama de casos de uso do aplicativo



Fonte: elaborado pelo autor.

A ação inicial do usuário deve ser realizada conforme descrito no caso de uso Selecionar modo de uso, apresentado no Quadro 13 do Apêndice A. Após isso, o usuário receberá uma classificação.

Para a opção Rastreo, o usuário será classificado como Usuário-Coletor e destina-se somente a um comportamento: Encerrar rastreo. No Quadro 14 do Apêndice A é apresentado este comportamento. O aplicativo deve enviar a geolocalização do dispositivo para usuários-coletores, após autenticação do cadastro, no momento em que seleciona o uso de modo “Rastreo”. A partir deste momento, o usuário pode encerrar o processo de rastreo, cancelando o envio de sua geolocalização.

Para a opção Consulta, o usuário será classificado como Usuário-Morador. As funcionalidades Consultar coletas por cidade e Consultar coletas por endereço são responsáveis pela definição das informações que serão utilizadas para filtrar a pesquisa da

coleta, a partir de um certo endereço. Estas são apresentadas, respectivamente, no Quadro 15 e Quadro 16 do Apêndice A.

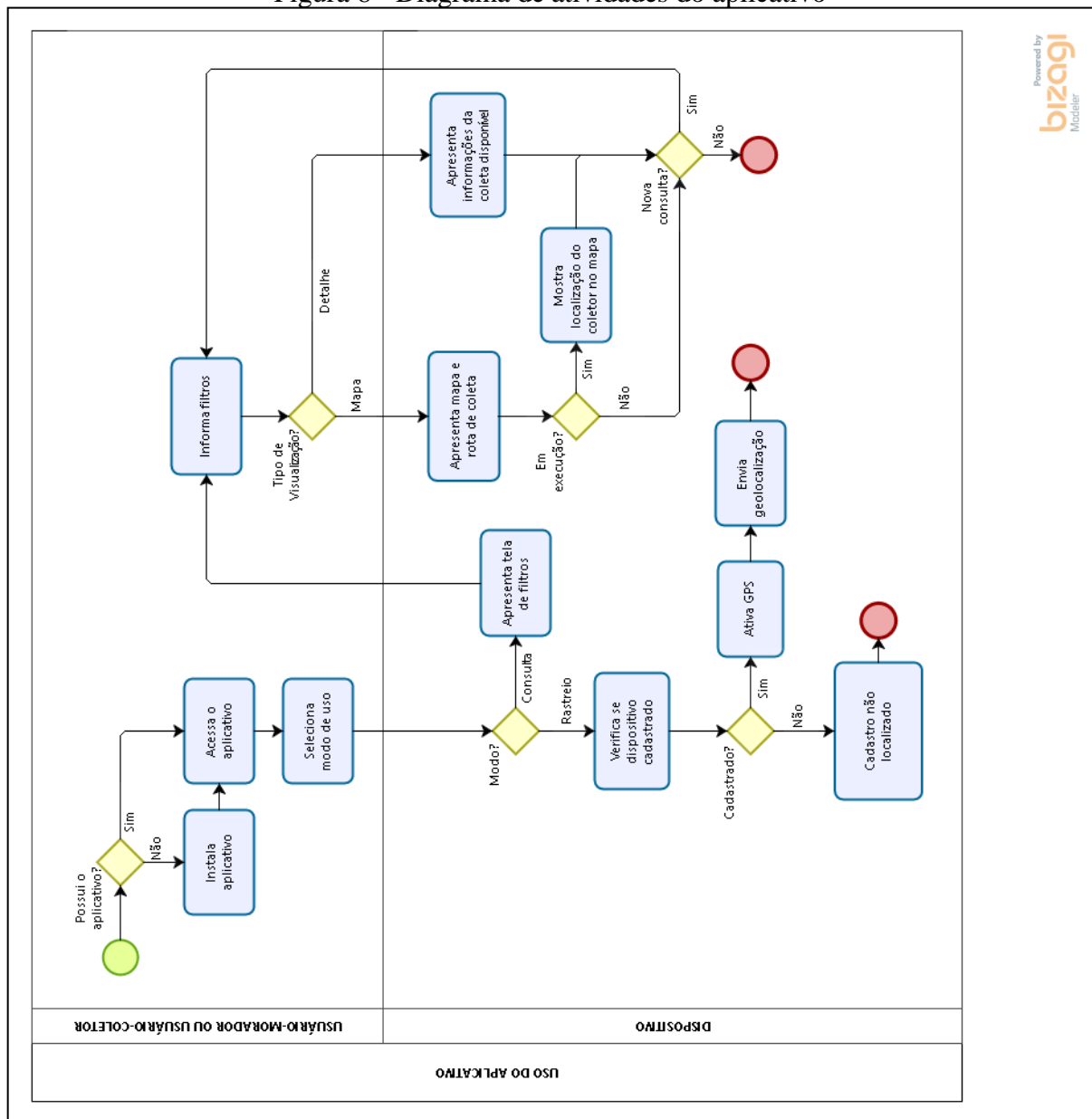
Após a filtragem de informações para pesquisa da rota apresenta-se um menu com dois tipos de visualização: *Detalhe* e *Mapa*. O conteúdo apresentado em cada tipo é demonstrado no Quadro 17 e Quadro 18 do Apêndice A.

E por fim, no Quadro 19 do Apêndice A, é mostrada a localização do dispositivo que está sendo utilizado durante a coleta de lixo e está com o aplicativo no modo *Rastrear*.

3.3.2 Diagrama de atividades do aplicativo

O diagrama de atividade do fluxo de uso do aplicativo Android é apresentado na Figura 8. Nesta são apresentadas as etapas para o uso do aplicativo pelos usuários. Após validações de instalação do aplicativo, serão apresentadas as funcionalidades dispostas pelo mesmo. O aplicativo inicializará apresentando duas opções para uso do aplicativo: *Consulta* e *Rastrear*. A opção *Consulta* tem suas funcionalidades ativadas para todos os usuários. Já a opção de *Rastrear* somente é funcional caso o usuário tenha seu dispositivo cadastrado para realização de coletas, por meio do módulo web.

Figura 8 - Diagrama de atividades do aplicativo



Fonte: elaborado pelo autor.

3.3.3 Diagramas complementares para o módulo web auxiliar

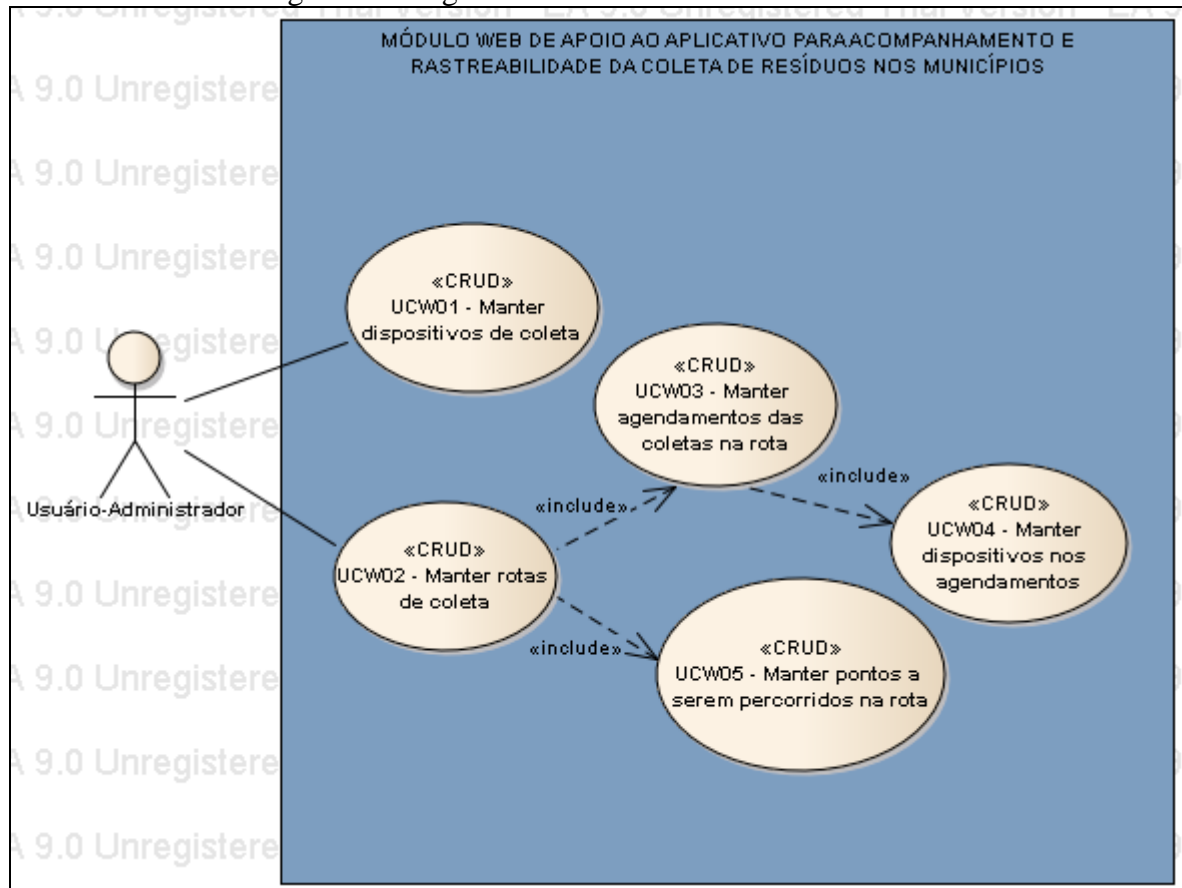
Apesar do módulo web não ser o foco deste trabalho, a compreensão de suas estruturas e entendimento de sua necessidade é importante para o correto funcionamento do aplicativo Android já que ele é o responsável por manter a integridade das informações.

3.3.3.1 Diagrama de casos de uso do módulo web auxiliar

O Diagrama de casos de uso do módulo web é direcionado a um único ator chamado Usuário-Administrador. Apresentado na Figura 9, é composto por 5 casos de uso: Manter dispositivos de coleta, Manter rotas de coleta, Manter agendamentos das

coletas na rota, Manter dispositivos nos agendamentos e Manter pontos a serem percorridos na rota. A autenticação dos dispositivos para utilização do modo Rastreo ocorre mediante o cadastro realizado neste módulo. Assim também todo o detalhamento das rotas e verificação dos dispositivos e agendamentos depende deste módulo.

Figura 9 - Diagrama de casos de uso do módulo web



Fonte: elaborado pelo autor.

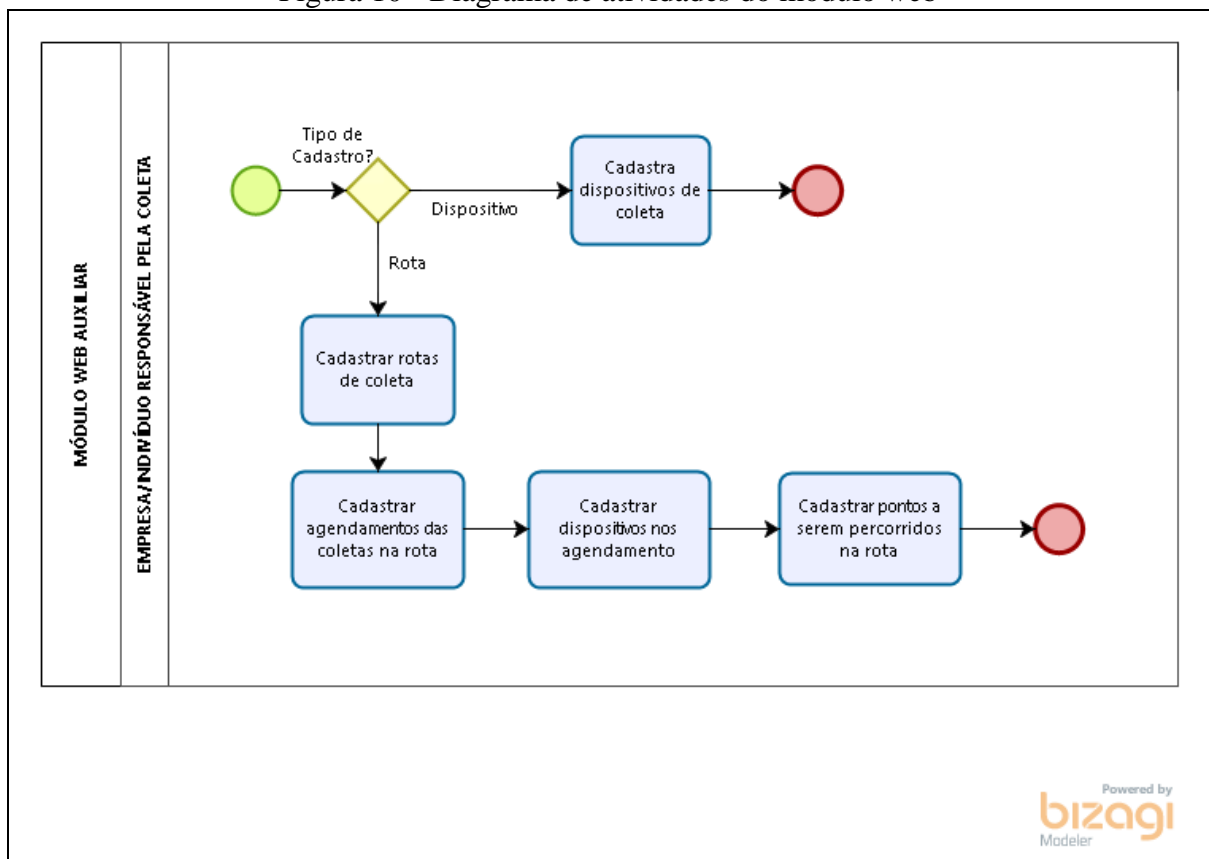
Primeiramente, é necessário que sejam realizados os cadastros dos dispositivos que serão usados nas coletas de lixo, conforme detalhamento da funcionalidade Cadastrar dispositivos de coleta no Quadro 20 do Apêndice A.

Após realizados os cadastros dos dispositivos, torna-se necessário Cadastrar rotas de coleta, que serão utilizadas especificamente no modo Consulta do aplicativo, de acordo com o Quadro 21 do Apêndice A. Todavia, a integridade das rotas depende de suas interações com os cadastros de agendamentos (Quadro 22 do Apêndice A), dispositivos dos agendamentos (Quadro 23 do Apêndice A) e pontos a serem percorridos (Quadro 24 do Apêndice A).

3.3.3.2 Diagrama de atividades do módulo web de apoio

O diagrama de atividade com os fluxos de cadastro e gerenciamento do módulo web, se encontram na Figura 10. Esse é um módulo específico, de suporte ao aplicativo, no qual serão cadastradas informações para as funcionalidades do aplicativo que serão utilizadas de acordo com o modo de uso selecionado. O cadastro fica a encargo do responsável pela realização das coletas no município. Este realiza os cadastros e o planejamento das coletas, além do registro dos dispositivos.

Figura 10 - Diagrama de atividades do módulo web



Fonte: elaborado pelo autor.

3.4 IMPLEMENTAÇÃO

Esta seção descreve as técnicas e ferramentas que foram utilizadas na implementação do aplicativo. São apresentados trechos da codificação para facilitar o entendimento das rotinas. Algumas funcionalidades são brevemente descritas de acordo com a implementação. Por fim é apresentada a operacionalidade do aplicativo.

3.4.1 Técnicas e ferramentas utilizadas

Para a construção do aplicativo foi utilizada a linguagem Java através da IDE Android Studio 3.0. Segundo o IMasters (2013, p. 1), “o Android Studio[...] é uma IDE para

desenvolvimento na plataforma Android baseado no IntelliJ Community Version”. O Google promoveu e recomenda fortemente que desenvolvedores optem por esta opção ao invés do Eclipse, pois tornou-a estável e a tem chamado de “IDE oficial” (INFOQ, 2014, p. 1). Ela disponibiliza um emulador virtual para validação dos aplicativos desenvolvidos e utiliza o Gradle para automatização de *builds* do projeto. Contudo, caso não se utilize um hardware de mais processamento, torna-se necessário algumas customizações nas configurações afim de que possa fazer uso da IDE e do emulador, pois quando utilizados simultaneamente (principalmente em modo debug) podem causar lentidão.

A API Google Maps v2 é utilizada para construção do mapa no aplicativo por meio da biblioteca `com.google.android.gms.maps`. Para que o mapa seja carregado, é necessário informar no arquivo de configuração do projeto uma chave criptografada única (`com.google.android.geo.API_KEY`) gerada a partir do site da Google Maps API (GOOGLE MAPS APIS, 2017c, p.1). As rotas foram desenhadas através de uma lista de pontos geográficos (latitude e longitude). No Quadro 5 pode-se visualizar o método `drawLinesRoute` que recebe como parâmetro uma lista de objetos da classe `LatLng` (linha 01), contendo a latitude e longitude de cada ponto que será adicionado no mapa.

Quadro 5 - Método `drawLinesRoute` para desenho da rota no mapa

```

01 protected void drawLinesRoute(List<LatLng> decodedPath) {
02     try {
03         GoogleMap map = SingletonMaps.getInstance().getMap();
04         if (map == null || decodedPath.size() == 0) {
05             return;
06         }
07         map.setMapType(GoogleMap.MAP_TYPE_NORMAL);
08
09         // Desenha a rota no mapa
10         map.addPolyline(
11             new PolylineOptions().addAll(decodedPath)
12                 .color(Color.parseColor("#3F51B5"))
13                 .width(30));
14
15         // Cria e adiciona o marcador inicial da rota
16         LatLng initialPosition = decodedPath.get(0);
17         MarkerOptions moi = createCustomMarkerOptions(initialPosition,
18 "Início", R.drawable.route_begin);
19         addMarkerToMap(moi, true);
20
21         // Cria e adiciona o marcador final da rota
22         LatLng finalPosition = decodedPath.get(decodedPath.size() -
23 1);
24         MarkerOptions moe = createCustomMarkerOptions(finalPosition,
25 "Término", R.drawable.route_end);
26         addMarkerToMap(moe, false);
27     } catch (Exception e) {
28         System.out.println("Erro durante a construção e inclusão da
29 rota no mapa: " + e.getMessage());
30     }
31 }

```

Fonte: elaborado pelo autor.

Além da rota, o mapa também recebe marcadores customizados para indicação do início e fim da rota (linhas 15 a 26). A criação e customização do marcador ocorre pelo método `createCustomMarkerOptions` do Quadro 6.

Quadro 6 – Método `createCustomMarkerOption` para customização do marcador

```

01 public MarkerOptions createCustomMarkerOptions(LatLng position,
02 String title, int rDrawableIcon) {
03     if (rDrawableIcon == 0) {
04         rDrawableIcon = R.drawable.garbage_collector;
05     }
06
07     // Criação de marcador para o mapa
08     MarkerOptions marker = new MarkerOptions();
09     marker.position(position).title(title).anchor(0.5f,
10 1).zIndex(1.0f);
11     marker.icon(BitmapDescriptorFactory.fromResource(rDrawableIcon));
12     marker.snippet("(" + position.latitude + "/" + position.longitude
13 + ")");
14     return marker;
15 }

```

Fonte: elaborado pelo autor.

Durante a customização do marcador pode-se destacar a definição da sua posição no mapa (linha 09) e o título a ser apresentado no momento em que este marcador for

selecionado (linha 09). Também se define o ícone do marcador com base nos recursos disponíveis da aplicação Android (linha 11). A partir desse momento, o marcador customizado pode ser adicionado ao mapa pelo método `addMarkerToMap` do Quadro 7.

Quadro 7 – Método `addMarkerToMap` para inclusão do marcador no mapa

```

01 public Marker addMarkerToMap(MarkerOptions markerOptions, boolean
02 locate) {
03     GoogleMap map = SingletonMaps.getInstance().getMap();
04     if (map == null) {
05         return null;
06     }
07
08     // Adicionar marcador customizado ao mapa
09     Marker marker = map.addMarker(markerOptions);
10     if (locate) {
11         CameraPosition p =
12 CameraPosition.builder().target(marker.getPosition())
13             .zoom(20).bearing(8).tilt(45).build();
14         map.moveCamera(CameraUpdateFactory.newCameraPosition(p));
15     }
16     return marker;
17 }

```

Fonte: elaborado pelo autor.

Este método recebe como parâmetros o marcador customizado e a definição se o mapa deve centralizar no local onde este marcador foi adicionado. Então, adiciona-se o marcador no mapa e o armazena na variável `marker` (linha 09). Em seguida, caso deva realizar a centralização, o mapa é ajustado centralizando o marcador na tela (linhas 10 a 15).

Quando obtidos os pontos para desenho da rota no mapa via *web service* com base nos cadastros, existem poucos pontos de referência, o que ocasiona o desenho de retas que não respeitam as delimitações das ruas. Para que isso não aconteça, complementa-se o intervalo das rotas com o serviço da Google Directions API¹, conforme apresentado no método `getUrlGoogleDirectionsAPI` do Quadro 8.

¹ No endereço <https://maps.googleapis.com/maps/api/directions/json>.

Quadro 8 – Método getUrlGoogleDirectionsAPI para adequação da rota

```

01 private String getUrlGoogleDirectionsAPI(List<PointBean> points) {
02     // Tipo de retorno esperado (XML ou JSON)
03     String output = "json";
04
05     // Ponto de partida da rota
06     LatLng origin = points.get(0).getLatLng();
07     String str_origin = "origin=" + origin.latitude + "," +
08 origin.longitude;
09
10     // Ponto de destino da rota
11     LatLng dest = points.get(points.size() - 1).getLatLng();
12     String str_dest = "destination=" + dest.latitude + "," +
13 dest.longitude;
14
15     // Pontos da rota entre a partida e o destino
16     StringBuffer waypoints = new StringBuffer("waypoints=");
17     for (int i = 1; 0 < i && i < points.size() - 1; i++) {
18         PointBean point = points.get(i);
19         waypoints.append(point.getLatitude()).append(",");
20         waypoints.append(point.getLongitude()).append("|");
21     }
22
23     // Concatenação dos parâmetros
24     String parameters = str_origin + "&" + str_dest + "&" +
25 waypoints.substring(0, waypoints.length() - 1);
26
27     // Construção do endereço web do serviço com parâmetros
28     String url = "https://maps.googleapis.com/maps/api/directions/" +
29 output + "?" + parameters;
30     return url;
31 }

```

Fonte: elaborado pelo autor.

Para este serviço deve-se obrigatoriamente informar os parâmetros `origin` e `destination`, com latitude e longitude respectivamente separados por vírgula (linhas 06 a 13). Seu retorno está exemplificado no Apêndice B e permite extrair a lista de pontos necessários para que as retas da rota no mapa respeitem corretamente as delimitações das ruas. O Quadro 9 apresenta os parâmetros aceitos no serviço Google Directions API.

Quadro 9 - Parâmetros para Google Directions API

Parâmetro	Opcional	Descrição
origin	Não	Endereço, latitude/longitude ou ID do local como ponto de partida da rota.
destination	Não	Endereço, latitude/longitude ou ID do local como ponto de destino da rota.
mode	Sim	Especifica o modo de transporte a ser usado ao calcular a rota. Opção padrão: <code>driving</code> .
waypoints	Sim	Especifica uma série de pontos de referência que alteram a rota, entre o ponto de partida e o de destino.
alternatives	Sim	Define se o serviço poderá fornecer mais uma rota como alternativa.
avoid	Sim	Indica os componentes que a rota ao ser calculada deve evitar.
language	Sim	Define o idioma de retorno dos resultados.
units	Sim	Especifica o sistema de unidades de exibição dos resultados.
region	Sim	Especifica o código de região como um valor de “domínio de nível superior” de dois caracteres.
arrival_time	Sim	Especifica a hora desejada de chegada para solicitações de rota em segundos.
departure_time	Sim	Especifica a hora desejada de saída.

Fonte: Google Directions API (2017a, p. 1).

O módulo web para realização dos cadastros utilizou a IDE Eclipse, porém não possui restrições quanto à ferramenta de desenvolvimento. As telas deste módulo encontram-se no Apêndice D.

A integração das informações entre aplicativo Android e o módulo web devem ocorrer por meio de *web services*, para o compartilhamento da localização do dispositivo e para a visualização dos cadastros que serão apresentados na tela de detalhamento das rotas. As informações são integradas ao aplicativo no formato JSON, conforme apresentado no Apêndice C. A obtenção da localização do dispositivo, utilizado na execução da coleta, por meio de GPS ocorre no método `runGPSPosition`, conforme apresentado no Quadro 10.

Quadro 10 – Método runGPSPosition verifica e solicita a habilitação do GPS e compartilha a localização no dispositivo no modo Rastreo

```

01 protected boolean runGPSPosition() {
02     LocationManager locationManager = (LocationManager)
03     getSystemService(LOCATION_SERVICE);
04
05     // Verifica habilitação de permissões necessárias para uso do GPS
06     if (ActivityCompat.checkSelfPermission(FollowActivity.this,
07 Manifest.permission.ACCESS_FINE_LOCATION) !=
08     PackageManager.PERMISSION_GRANTED &&
09 ActivityCompat.checkSelfPermission(FollowActivity.this,
10 Manifest.permission.ACCESS_COARSE_LOCATION) !=
11 PackageManager.PERMISSION_GRANTED) {
12     // Questiona habilitação das permissões
13     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
14         requestPermissions(new
15 String[]{Manifest.permission.ACCESS_COARSE_LOCATION,
16 Manifest.permission.ACCESS_FINE_LOCATION,
17 Manifest.permission.INTERNET}
18         , 10);
19     }
20     return false;
21 }
22
23 locationManager.requestLocationUpdates("gps", 3000, 0, new
24 LocationListener() {
25     @Override
26     public void onLocationChanged(Location location) {
27         DeviceBean deviceBean =
28 SingletonDevice.getInstance().getDeviceBean();
29         deviceBean.setLast_latitude(location.getLatitude());
30         deviceBean.setLast_longitude(location.getLongitude());
31         SingletonDevice.getInstance().setDeviceBean(deviceBean);
32         // Envia as novas coordenadas da posição do dispositivo
33         new DeviceService().execute(ActionEnum.UPDATE,
34 SingletonDevice.getInstance().getDeviceBean());
35     }
36
37     @Override
38     public void onStatusChanged(String provider, int status,
39 Bundle extras) {
40     }
41
42     @Override
43     public void onProviderEnabled(String provider) {
44     }
45
46     @Override
47     public void onProviderDisabled(String s) {
48         Intent i = new
49 Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
50         startActivity(i);
51     }
52 });
53 return true;
54 }

```

Fonte: elaborado pelo autor.

Como o compartilhamento da localização depende da habilitação do GPS do dispositivo, é necessário que seja realizada a verificação de sua situação e, caso negativo,

solicitado ao usuário (linhas 05 a 21). Então, é definido um período em milissegundos ao qual será enviada as coordenadas de localização do dispositivo (linhas 23 a 35).

A consulta de dispositivos e rotas cadastradas bem como a atualização da geolocalização do dispositivo ocorrem por meio de *web services* definidos no servidor de aplicação. Como exemplos, o endereço executado para autenticação do dispositivo (baseado no id do dispositivo Android) é o `http://localhost:8080/device?id=6392be568b3533bd`, enquanto a execução da consulta de rotas que abrangem um certo endereço informado (baseado na latitude e longitude) é realizada por meio do endereço `http://localhost:8080/route?latitude=-26.8888697&longitude=-49.0973044`. O método `getJSONFromAPI` é o responsável por executar esta comunicação, conforme Quadro 11.

Quadro 11 - Método `getJSONFromAPI` para comunicação de *web services*

```

01 public static String getJSONFromAPI(@NonNull String url,
02 @RequestMethod String method, int timeOutRead, int timeOutConnect) {
03     String jsonAsString = "";
04     HttpURLConnection connection = null;
05     InputStream is = null;
06     try {
07         URL apiEnd = new URL(url);
08         connection = (HttpURLConnection) apiEnd.openConnection();
09         connection.setRequestMethod(method);
10         connection.setReadTimeout(timeOutRead);
11         connection.setConnectTimeout(timeOutConnect);
12         //connection.setRequestProperty("Content-Type",
13 "application/json; charset=UTF-8");
14         connection.connect();
15         if (connection.getResponseCode() <
16 HttpURLConnection.HTTP_BAD_REQUEST) {
17             is = connection.getInputStream();
18         } else {
19             is = connection.getErrorStream();
20         }
21         jsonAsString = InputStreamUtils.parseInputStreamToString(is);
22     } catch (MalformedURLException e) {
23         e.printStackTrace();
24     } catch (IOException e) {
25         e.printStackTrace();
26     } finally {
27         if (is != null) {
28             try {
29                 is.close();
30             } catch (IOException e) {
31                 e.printStackTrace();
32             }
33         }
34         if (connection != null) {
35             connection.disconnect();
36         }
37     }
38     return jsonAsString;
39 }
40

```

Fonte: elaborado pelo autor.

Para que seja possível estabelecer esta comunicação são necessários o endereço padrão do *web service* (contendo os parâmetros esperados e seus respectivos valores), o método requisição, os tempos de conexão e de *timeout*. São definidas as configurações para a comunicação (linhas 07 a 13) e, em seguida, estabelecida a conexão que retorna uma resposta do servidor de aplicação (linhas 14 a 20). Essa resposta deve ser convertida para `String` (linha 21) para construção de JSONs que são utilizados nas rotinas do aplicativo e provém do acesso aos cadastros do módulo web por meio do servidor de aplicação. As estruturas dos cadastros são geradas a partir das classes `bean` definidas no servidor de aplicação para um banco de dados MongoDB e a requisição de dados ocorre por meio do objeto `XMLHttpRequest`.

3.4.2 Operacionalidade da implementação

O aplicativo para Rastreamento e Acompanhamento da Coleta de Lixo (RACL) foi desenvolvido objetivando auxiliar os moradores a realizarem o descarte de resíduos nos devidos horários, permitindo que possam se informar e acompanhar a execução das coletas. As imagens apresentadas foram obtidas durante a execução do aplicativo a partir do dispositivo virtual Nexus 5X API 26 x86, emulado a partir da IDE Android Studio.

Ao acessar o dispositivo, é apresentada uma tela para que o usuário possa escolher qual o modo de uso ele deseja utilizar e isso definirá quais funcionalidades estarão disponíveis para o mesmo (Figura 11). O dispositivo virtual possui como identificador o código “6392be568b3533bd”. Para que ele possa ser utilizado como rastreo, foi necessária a criação de um cadastro de dispositivo com este mesmo identificador.

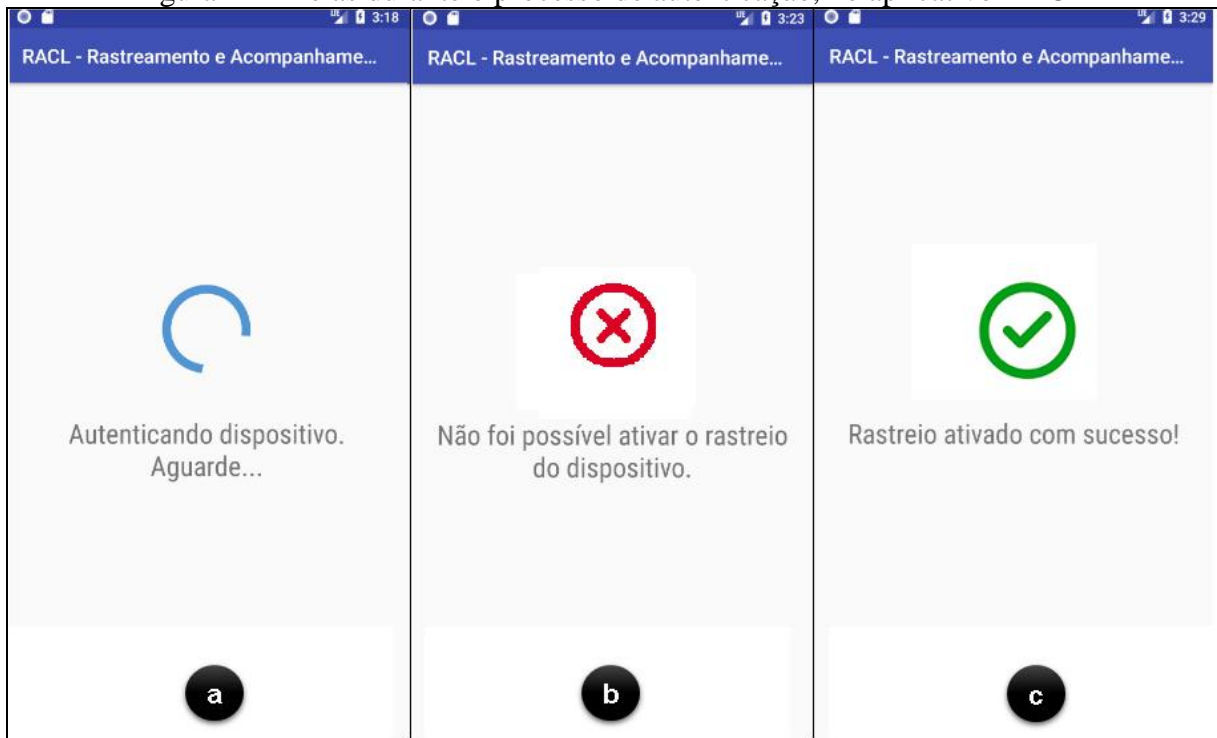
Figura 11 - Tela inicial do aplicativo RACL



Fonte: elaborado pelo autor.

Ao selecionar o modo *Rastreo*, o aplicativo realiza uma consulta por meio de *web service* passando como parâmetro seu código identificador. A informação obtida por meio desta consulta, encontra-se no Apêndice C. Durante a realização da verificação, é apresentada uma tela com a mensagem “Autenticando dispositivo. Aguarde...” (Figura 12a) informando ao usuário a ação que está sendo realizada. Em sequência, caso o dispositivo não tenha sido autenticado, apresenta a mensagem “Não foi possível ativar o rastreo do dispositivo” (Figura 12b). Essa situação pode ocorrer devido à falta de cadastro do dispositivo ou, até mesmo, a indisponibilidade de conexão. Quando autenticado, é apresentada a seguinte mensagem ao usuário “Rastreo ativado com sucesso” (Figura 12c). A partir deste momento, a localização do dispositivo que foi autenticado está sendo compartilhada para outros dispositivos que estejam utilizando o modo *Consulta* do aplicativo. Nota-se que o GPS deve estar ativado para esta funcionalidade. Do contrário, será solicitada permissão ao usuário para fazer ativação deste.

Figura 12 – Telas durante o processo de autenticação, no aplicativo RACL



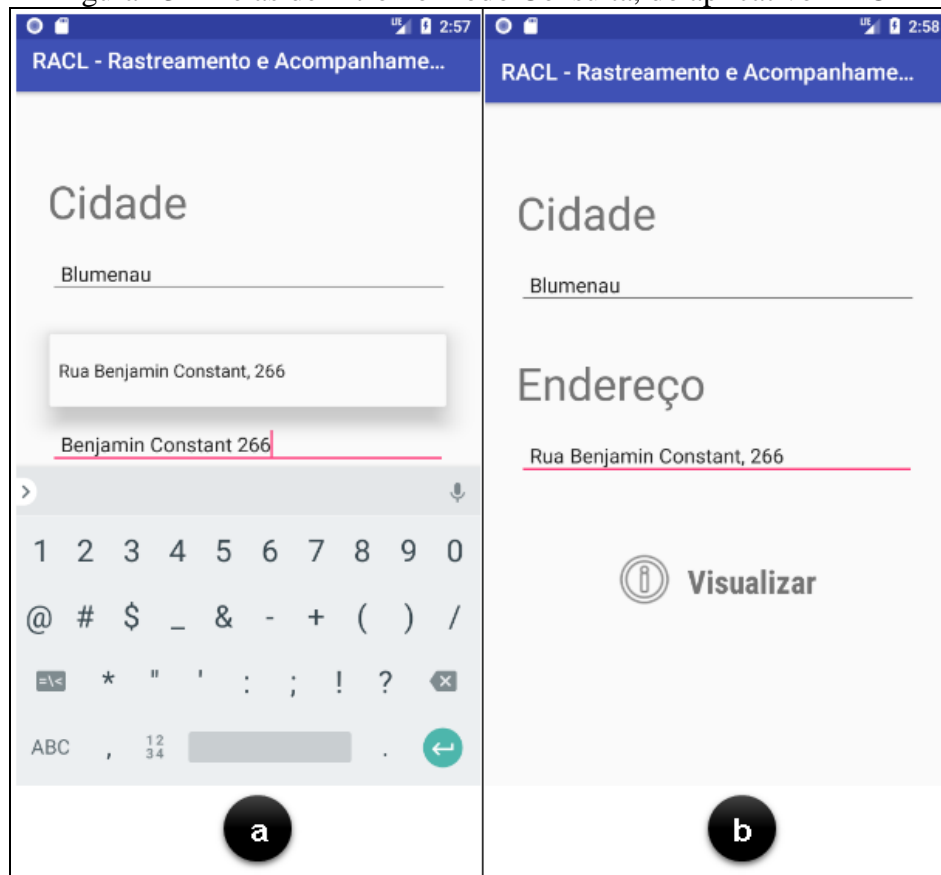
Fonte: elaborado pelo autor.

O rastreo do dispositivo fica ativo somente enquanto o aplicativo não é encerrado pelo usuário ou enquanto este não retorna ao menu principal do aplicativo.

Ao selecionar modo *Consulta* do aplicativo, o aplicativo assume somente como funcionamento o recebimento de informação e não mais compartilhamento. É exibida uma tela para que o usuário informe os filtros da consulta de rotas de coletas. São apresentados dois campos auto sugestivos de filtro: *Cidade* e *Endereço* (Figura 13a). À medida que o usuário realiza a digitação, são sugeridas opções para o usuário selecionar.

O campo *Cidade* necessita ser informado obrigatoriamente e sua abrangência é nacional (somente cidades brasileiras). Caso este campo não seja informado, não será possível informar o campo endereço e, conseqüentemente, não permitirá a consulta das rotas. Preenchido este campo, o endereço pode então ser informado. Por fim, basta utilizar o botão *Visualizar* (Figura 13b).

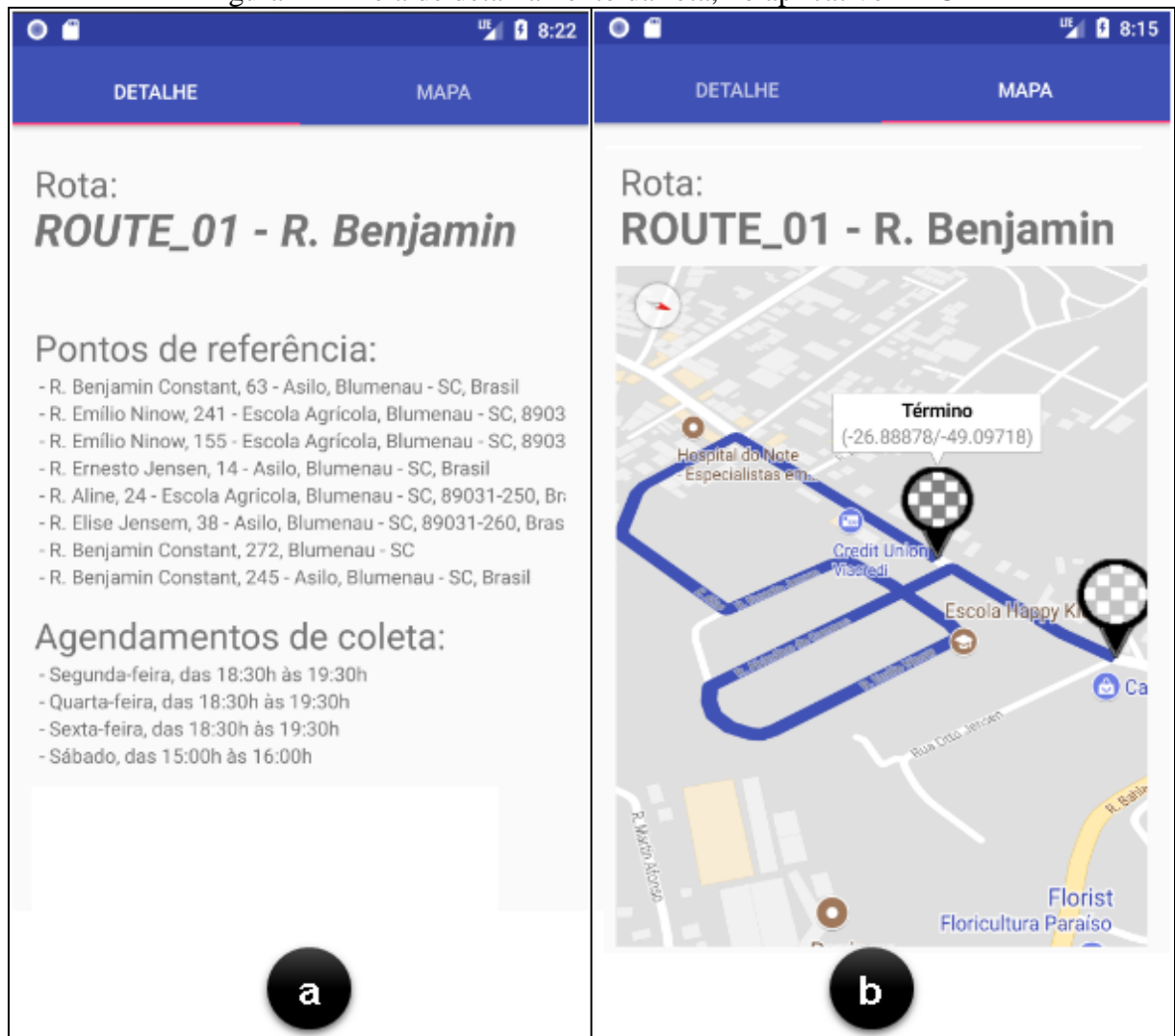
Figura 13 – Telas de filtro no modo Consulta, do aplicativo RACL



Fonte: elaborado pelo autor.

O aplicativo realiza uma verificação nas rotas cadastradas pelo módulo web por meio de *web service*, enviando as coordenadas do endereço solicitado pelo usuário. Não localizando nenhuma rota de coleta para o endereço informado, é apresentada uma mensagem notificando o usuário. Em caso afirmativo, será apresentada uma tela com o detalhamento da rota de coleta realizada no endereço, como pode-se verificar na Figura 14. O retorno desta consulta pode ser verificado no Quadro 26 do Apêndice C.

Figura 14 – Tela de detalhamento da rota, no aplicativo RACL



Fonte: elaborado pelo autor.

A tela de detalhamento possui duas opções de apresentação: Detalhe (Figura 14a) e Mapa (Figura 14b). A opção Detalhe apresenta as informações de forma descritiva, descrevendo o nome dos pontos percorridos pela rota e horário de realização das coletas. Já a opção Mapa, conecta os pontos cadastrados no mapa, desenhando a rota a ser percorrida durante a coleta. Caso a coleta esteja sendo executada no momento da consulta e haja um dispositivo coletor sendo rastreado, a posição desse dispositivo será apresentada no mapa, para acompanhamento por parte do usuário.

3.5 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados os resultados obtidos com este trabalho realizando uma análise comparativa entre os trabalhos correlatos e apresentando a forma de validação do aplicativo.

3.5.1 Comparação com os trabalhos correlatos em relação ao RACL

O Quadro 12 apresenta um comparativo entre as principais características do RACL em relação aos trabalhos correlatos.

Quadro 12 - Principais características dos correlatos em comparação ao RACL

Trabalhos relacionados/ Características	Coletas RGG Transportador (GOOGLE PLAY, 2017a, p.1)	VanCollect (GOOGLE PLAY, 2015, p.1)	Coleta Urbana (GOOGLE PLAY, 2016, p.1)	Limpa Rápido (GOOGLE PLAY, 2017b, p.1)	RACL
Tecnologia de localização	GPS e acesso à internet	GPS e acesso à internet	GPS e acesso à internet	GPS e acesso à internet	GPS e acesso à internet
Armazenamento	USB e acessa o que foi armazenado	USB e acessa o que foi armazenado	Não informado	USB e acessa o que foi armazenado	Através de <i>web service</i> , no módulo web
Autenticação	Login	Não informado	Não informado	Login	Através de <i>web service</i> , no módulo web
Apresentação de informações	Mapa e lista de ocorrências	Tela de detalhamento	Tela de detalhamento, mapa e localização em tempo real	Tela de detalhamento, mapa e localização em tempo real, notificações	Tela de detalhamento, mapa, localização em tempo real intermunicipal, rastreo do dispositivo
Adicionais	Coleta informações e tem acesso à internet, interpreta a configuração do servidor Google, não permite o dispositivo entrar em Stand-by	Coleta informações e tem acesso à internet, interpreta a configuração do servidor Google, não permite o dispositivo entrar em Stand-by	Coleta informações e tem acesso à internet, interpreta a configuração do servidor Google	Coleta informações e tem acesso à internet, interpreta a configuração do servidor Google, controla vibração do dispositivo, não permite o dispositivo entrar em Stand-by	Coleta informações e tem acesso à internet, interpreta a configuração do servidor Google, não permite o dispositivo entrar em Stand-by

Fonte: elaborado pelo autor.

O aplicativo desenvolvido conseguiu reunir algumas das principais funcionalidades apresentadas nos aplicativos Coletas RGG Transportador (GOOGLE PLAY, 2017a, p.1), VanCollect (GOOGLE PLAY, 2015, p.1), Coleta Urbana (GOOGLE PLAY, 2016, p.1) e Limpa Rápido (GOOGLE PLAY, 2017b, p.1).

A construção de uma rota no mapa é resultado da definição de pontos a serem percorridos entre um ponto de origem e outro de destino, permitindo assim uma visualização dinâmica do trajeto total, da mesma maneira como ocorre no Coletas RGG Transportador, Coleta Urbana e Limpa Rápido. Ao permitir a definição do endereço a que se deseja consultar as coletas programadas, o aplicativo atende a funcionalidade presente no VanCollect, Coleta Urbana e Limpa Rápido, sendo que este último permite que a localização desejada também possa ser identificada diretamente no mapa. O detalhamento das informações de coleta é um recurso que também foi implementado e facilita a compreensão do usuário final, contido em todos os correlatos. Contudo, o RACL não se limita a consultas de coletas dentro de uma única cidade e é também o responsável por compartilhar a geolocalização do dispositivo utilizado na execução da coleta. Em contrapartida, não armazena dados comuns no aplicativo como ocorre na maioria dos citados, o que poderia reduzir o consumo de pacote dados durante seu uso.

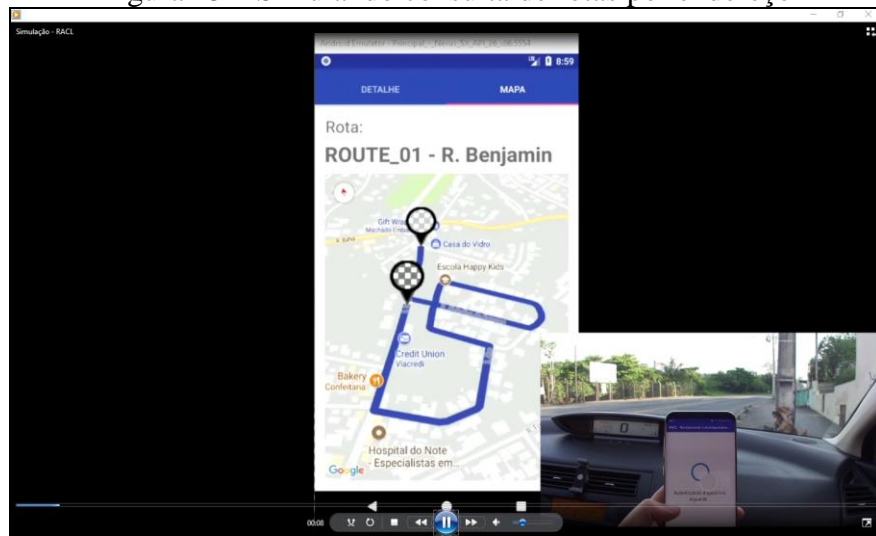
3.5.2 Validação do aplicativo

O aplicativo desenvolvido possui duas funcionalidades principais que precisam ser validadas em paralelo: o acompanhamento em tempo real e o rastreamento da execução da coleta.

O aplicativo foi transferido para um dispositivo Android físico para ser utilizado no modo `Rastreamento` e, ao mesmo tempo, emulado em uma máquina de desenvolvimento através da IDE Android Studio. O dispositivo emulado fez uso do modo `Consulta` para o endereço `Rua Benjamin Constant, 266`, no qual foi apresentada a rota `ROUTE_01 - R. Benjamin` (Figura 15). No momento em que o modo `Rastreamento` foi selecionado, iniciou-se o compartilhamento da localização do dispositivo físico para o mapa do aplicativo no dispositivo emulado (Figura 16). Conforme movimentado, a cada 5 segundos, ocorria uma atualização da posição do dispositivo físico no mapa (Figura 17).

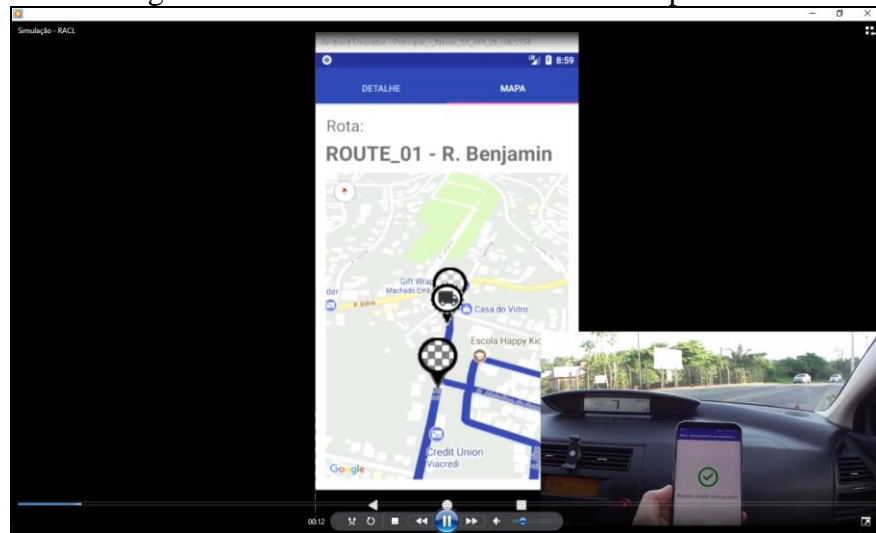
Com esta validação foi possível confirmar a funcionalidade do aplicativo desenvolvido, permitindo que este seja tanto utilizado para consulta de rotas de coleta por usuários comuns quanto por rastreamento da localização de coletores de lixo durante suas atividades em tempo real.

Figura 15 – Simulando consulta de rotas por endereço



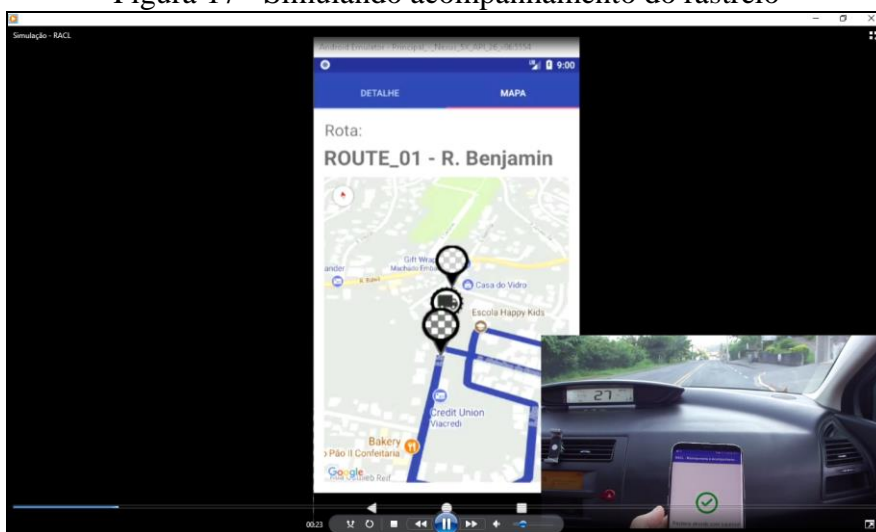
Fonte: elaborado pelo autor.

Figura 16 - Simulando rastreamento do dispositivo



Fonte: elaborado pelo autor.

Figura 17 - Simulando acompanhamento do rastreo



Fonte: elaborado pelo autor.

4 CONCLUSÕES

O projeto teve seu objetivo principal alcançado, o qual era desenvolver um aplicativo que permitisse tanto o compartilhamento da execução das coletas sendo realizadas nas cidades quanto a consulta dos horários programados e seu acompanhamento. Os objetivos específicos que eram permitir aos usuários verificar os horários de coleta de resíduos a partir de uma certa localidade, visualizando a rota por meio de um mapa e a localização do dispositivo utilizado pelo coletor quando a coleta estivesse em execução também foram alcançados.

O objetivo de exibição das rotas de coleta em mapas foi alcançado. Utilizando os pontos (endereços) dos cadastros de rotas, compartilhados pelo módulo web e acessados através de *web service*, a Google Directions API realiza uma complementação entre cada dois pontos, traçando todos os pontos necessários para o desenho da rota no mapa. A construção desse mapa é realizada com a API do Google Maps.

O objetivo de compartilhar a localização atual do dispositivo foi alcançada. O aplicativo envia a localização para o servidor de aplicação através de *web service*, fazendo uso das funcionalidades disponíveis no `LocationManager` – que coleta dados do GPS do dispositivo – e estabelecendo uma conexão com a internet. A partir desse momento, a localização do coletor (Usuário-Coletor) é propagada para acompanhamento de moradores (Usuário-Morador) em outros dispositivos e apresentada no mapa.

A ferramenta Android Studio 3.0, utilizada para o desenvolvimento deste aplicativo, cumpre com seu objetivo ao atender os mesmos requisitos de sua antecessora, a ferramenta Eclipse com o plugin ADT. Contudo, requer um bom hardware e uma otimizada configuração, visto que o Android Studio é muito pesado e causa certa lentidão durante o carregamento de alguns processos. Seu dispositivo virtual é muito prático e funcional na validação de aplicações.

4.1 EXTENSÕES

Algumas extensões possíveis para esse trabalho são:

- a) criação de lembretes com os horários de coletas na agenda do dispositivo;
- b) armazenamento de filtros de consulta preferidos do usuário;
- c) definir modo de uso favorito do aplicativo, para carregar automaticamente;
- d) apresentar o tempo de espera até que o dispositivo coletor chegue no endereço em que a consulta foi realizada;
- e) permitir filtrar as consultas por horário inicial e final;

- f) otimização das requisições realizadas buscando dados e informações das coletas.

REFERÊNCIAS

- ANDROID DEVELOPERS. **Location and maps**. [S.I.], 2017. Disponível em: <<https://developer.android.com/guide/topics/location/index.html>>. Acesso em: 14 novembro 2017.
- ALVES, A.O.; MOURA, F.M. Resíduos domésticos e da agricultura nos assentamentos do litoral norte e zona da mata de Alagoas. In: EL-DEIR, S.G. **Resíduos sólidos: Perspectivas e desafios para a gestão integrada**. 1. ed. Recife, 2014. cap. 2, p. 47-58. Disponível em: <http://www.mpgo.mp.br/portal/arquivos/2015/01/28/16_42_43_117_ebook_residuos_solidos_2014.pdf>. Acesso em: 14 novembro 2017.
- FIESP. **Aumento do consumo Brasileiro impulsiona expectativas da MOVINTER 2014**. [S.I.], 2014. Disponível em: <<http://www.fiesp.com.br/simm/noticias/aumento-do-consumo-brasileiro-impulsiona-expectativas-da-movinter-2014>>. Acesso em: 22 outubro 2017.
- FREITAS, Eduardo de. **Os Problemas Provocados pelo Lixo**. [S.I.], 2008. Disponível em: <<http://www.mundoeducacao.com/geografia/os-problemas-provocados-pelo-lixo.htm>>. Acesso em: 22 outubro 2017.
- FOCO RURAL. **Empresas paranaenses usam rastreabilidade como fator de segurança para frutas, legumes e verduras**. [S.I.], 2016. Disponível em: <http://www.focorural.com/detalhes/n/n/8986/37/Empresas_paranaenses_usam_rastreabilidade_e_como_fator_de_seguranca_para_frutas__legumes_e_verduras.html>. Acesso em: 11 novembro 2017.
- G1 ZONA DA MATA. **Aplicativo que monitora coleta de lixo começa a funcionar em Juiz de Fora**. Juiz de Fora, 2016. Disponível em: <<http://g1.globo.com/mg/zona-da-mata/noticia/2016/07/aplicativo-que-monitora-coleta-de-lixo-comeca-funcionar-em-juiz-de-fora.html>>. Acesso em: 28 novembro 2017.
- GOOGLE. **Google Maps**. [S.I.], 2017. Disponível em: <<https://www.google.com.br/maps/about>>. Acesso em: 11 novembro 2017.
- GOOGLE DIRECTIONS API. **WebServices: Directions API: Guia do desenvolvedor**. [S.I.], 2017a. Disponível em: <<https://developers.google.com/maps/documentation/directions/intro>>. Acesso em: 19 novembro 2017.
- _____. [S.I.], 2017b. Disponível em: <<https://maps.googleapis.com/maps/api/directions/json?origin=-26.8886192,-49.09570150000002&destination=-26.8889187,-49.09678050000002>>. Acesso em: 18 novembro 2017.
- GOOGLE MAPS APIS. **Maps Android API: Geolocalização: Como exibir a posição do usuário ou do dispositivo no Maps**. [S.I.], 2017a. Disponível em: <<https://developers.google.com/maps/documentation/javascript/geolocation?hl=pt-br#o-que--geolocaliza>>. Acesso em: 10 novembro 2017.
- _____. **Maps Android API: Câmera e visualização**. [S.I.], 2017b. Disponível em: <<https://developers.google.com/maps/documentation/android-api/views>>. Acesso em: 11 novembro 2017.
- _____. **Maps Android API: Primeiros passos**. [S.I.], 2017c. Disponível em: <<https://developers.google.com/maps/documentation/android/start#get-key>>. Acesso em: 19 novembro 2017.

GOOGLE MAPS HELP. **How to use Google Maps.** [S.I.], 2016. Disponível em: <<https://support.google.com/maps/answer/144349>>. Acesso em: 11 novembro 2017.

GOOGLE PLAY. **Coletas RGG Transportador.** [S.I.], 2017a. Disponível em: <<https://play.google.com/store/apps/details?id=com.csjsistemas.coletasrggcoletor>>. Acesso em: 14 novembro 2017.

_____. **Coleta Urbana.** [S.I.], 2016. Disponível em: <<https://play.google.com/store/apps/details?id=com.jfmais.limpa&rdid=com.jfmais.limpa>>. Acesso em: 24 outubro 2017.

_____. **Limpa Rápido.** [S.I.], 2017b. Disponível em: <<https://play.google.com/store/apps/details?id=com.pv.eresiduos>>. Acesso em: 28 novembro 2017.

_____. **VanCollect.** [S.I.], 2015. Disponível em: <<https://play.google.com/store/apps/details?id=net.recollect.vancouver>>. Acesso em: 14 novembro 2017.

GRIFFIN, Darren. **How does the Global Positioning System work?.** [S.I.], 2011. Disponível em: <<http://www.pocketgpsworld.com/howgpsworks.php>>. Acesso em: 14 novembro de 2017.

ICPD. **Curso de GPS e cartografia básica.** [S.I.], 2015. Disponível em: <http://www.fernandoquadro.com.br/files/cartografia_e_gps.pdf>. Acesso em: 14 novembro 2017.

IMASTERS. **Android Studio: vantagens e desvantagens com relação ao Eclipse.** [S.I.], 2013. Disponível em: <<https://imasters.com.br/mobile/android/android-studio-vantagens-e-desvantagens-com-relacao-ao-eclipse/?trace=1519021197&source=single>>. Acesso em: 07 agosto 2017.

INFOQ. **Desenvolvedores Android são encorajados a migrar do Eclipse para o Android Studio 1.0.** [S.I.], 2014. Disponível em: <<https://www.infoq.com/br/news/2014/12/android-studio-1>>. Acesso em: 07 agosto 2017.

LOVIS, Christian. **Traceability in Healthcare: Crossing Boundaries.** [S.I.], 2008. Disponível em: <<https://imia.schattauer.de/en/contents/archive/issue/2256/manuscript/9828/download.html>>. Acesso em: 10 novembro 2017.

MIRANDA, M.J.L.; STEUER, I.R.W. Diagnóstico e análise socioambiental do gerenciamento dos resíduos sólidos urbanos do município de Carpina - PE. In: EL-DEIR, S.G. **Resíduos sólidos: Perspectivas e desafios para a gestão integrada.** 1. ed. Recife, 2014. cap. 2, p. 59-68. Disponível em: <http://www.mpggo.mp.br/portal/arquivos/2015/01/28/16_42_43_117_ebook_residuos_solidos_2014.pdf>. Acesso em: 22 outubro 2017.

PAZ, Y.M.; MORAIS, M.M. Resíduos sólidos e percepção ambiental no semiárido pernambucano, uma análise temporal em comunidade rural. In: EL-DEIR, S.G. **Resíduos sólidos: Perspectivas e desafios para a gestão integrada.** 1. ed. Recife, 2014. cap. 2, p. 91-99. Disponível em: <http://www.mpggo.mp.br/portal/arquivos/2015/01/28/16_42_43_117_ebook_residuos_solidos_2014.pdf>. Acesso em: 22 outubro 2017.

PEREIRA, Raquel Catarina Ribeiro Gonçalves Rianço. **Rastreabilidade na Indústria das Rolhas**. set, 2014. 82f. Dissertação para obtenção do Grau de Mestre em Engenharia Química e Bioquímica (Licenciatura em Engenharia Química e Bioquímica), Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Lisboa. Disponível em:

<https://run.unl.pt/bitstream/10362/13935/1/Pereira_2014.pdf>. Acesso em: 11 novembro 2017.

PESSÔA, Júlia. **Prefeitura lança app para otimizar coleta de lixo**. Minas Gerais, 2016. Disponível em: <<http://www.tribunademinas.com.br/noticias/cidade/20-07-2016/prefeitura-lanca-app-para-otimizar-coleta-de-lixo.html>>. Acesso em: 24 outubro 2017.

PREFEITURA DE SÃO PAULO. **Prefeitura lança aplicativo com informações sobre serviços de limpeza e coleta em SP**. São Paulo, 2017. Disponível em: <<http://www.capital.sp.gov.br/noticia/prefeitura-lanca-aplicativo-com-informacoes-sobre-servicos-de-limpeza-e-coleta-em-sp>>. Acesso em: 24 outubro 2017.

SANSON, Marcio. **Rastreabilidade: eficiência na redução de custos logísticos**. [S.I.], 2014. Disponível em: <<http://www.unoerp.com.br/post/rastreabilidade-eficiencia-na-reducao-de-custos-logisticos>>. Acesso em: 10 novembro 2017.

TRIGUEIRO, Thiago. **Volume de lixo cresce em proporção maior que a população brasileira**. [S.I.], 2013. Disponível em: <<http://g1.globo.com/jornal-da-globo/noticia/2013/05/volume-de-lixo-cresce-em-proporcao-maior-que-populacao-brasileira.html>>. Acesso em: 22 outubro 2017.

APÊNDICE A – Detalhamento dos casos de uso

Este apêndice apresenta os detalhes de casos de uso do aplicativo referenciados ao longo do trabalho.

São apresentados os casos de uso do aplicativo Android no Quadro 13, Quadro 14, Quadro 15, Quadro 16, Quadro 17, Quadro 18 e Quadro 19.

Quadro 13 - Detalhamento do caso de uso *Selecionar modo de uso*

UC01 – Selecionar modo de uso	
Pré-condições	O dispositivo deve estar conectado à internet. O dispositivo deve estar com o GPS ativo.
Cenário principal	01) O usuário seleciona o modo de uso do aplicativo.
Cenário alternativo 1	01.1) O usuário seleciona o modo Consulta. 01.2) O aplicativo abre a tela de filtros da Consulta.
Cenário alternativo 2	01.1) O usuário seleciona o modo Rastreo. 01.2) O aplicativo abre a tela de autenticação do dispositivo. 01.3) O aplicativo autentica o dispositivo. 01.4) O aplicativo compartilha a geolocalização do dispositivo a cada 5 segundos.
Exceção 01	No passo 01.2, em modo Rastreo, caso não haja conectividade com a internet, o aplicativo apresenta uma mensagem de erro ao usuário (<i>Não foi possível ativar o rastreo do dispositivo</i>).
Exceção 02	No passo 01.4, em modo Rastreo, caso não tenha permissão para ativação do GPS, apresenta uma mensagem de erro ao usuário (<i>Não foi possível ativar o rastreo do dispositivo</i>).
Pós-condições	O usuário recebe uma classificação e são apresentadas as funcionalidades destinadas a esta classificação.

Fonte: elaborado pelo autor.

Quadro 14 – Detalhamento do caso de uso *Encerrar rastreo*

UC02 – Encerrar rastreo	
Pré-condições	O dispositivo deve estar conectado à internet. O dispositivo deve estar com o GPS ativo. O aplicativo deve estar no modo Rastreo.
Cenário principal	01) O usuário retorna à tela de seleção de modo de uso do aplicativo. 02) O aplicativo desativa a funcionalidade de rastreo.
Cenário alternativo 1	01.1) O usuário utiliza a opção Voltar do dispositivo.
Cenário alternativo 2	01.1) O usuário encerra o aplicativo.
Pós-condições	O aplicativo é reiniciado.

Fonte: elaborado pelo autor.

Quadro 15 – Detalhamento do caso de uso Consultar coletas por cidade

UC03 – Consultar coletas por cidade	
Pré-condições	O dispositivo deve estar conectado à internet. O aplicativo deve estar no modo Consulta.
Cenário principal	01) O usuário digita o nome de uma cidade no campo Cidade. 02) O aplicativo pesquisa cidades no Brasil a partir do 5º caractere digitado pelo usuário no campo Cidade. 03) O aplicativo sugere uma lista de cidades conforme digitação do usuário. 04) O usuário seleciona a cidade desejada dentre as opções listadas. 05) O aplicativo ativa a digitação e pesquisa no campo de endereço.
Exceção 01	No passo 02, caso não haja conectividade com a internet, não realizará a pesquisa de cidades e não apresentará uma lista para o usuário selecionar.
Pós-condições	O campo Endereço ativa a pesquisa de endereços na cidade selecionada.

Fonte: elaborado pelo autor.

Quadro 16 – Detalhamento do caso de uso Consultar coletas por endereço

UC04 – Consultar coletas por endereço	
Pré-condições	O dispositivo deve estar conectado à internet. O aplicativo deve estar no modo Consulta.
Cenário principal	01) O usuário digita um endereço da cidade informada no campo Endereço. 02) O aplicativo pesquisa endereços a partir do 5º caractere digitado pelo usuário no campo Endereço. 03) O aplicativo sugere uma lista de endereços na cidade, conforme digitação do usuário. 04) O usuário seleciona o endereço desejado dentre as opções listadas. 05) O usuário clica no botão Visualizar. 06) O aplicativo apresenta o detalhamento da coleta para a cidade e endereço selecionados.
Exceção 01	No passo 02, caso não haja conectividade com a internet, não realizará a pesquisa de cidades e não apresentará uma lista para o usuário selecionar.
Exceção 02	No passo 05, caso não haja rota para o endereço informado, o aplicativo apresenta uma mensagem de erro ao usuário (<i>Nenhuma rota localizada para este endereço</i>).
Pós-condições	O aplicativo abre uma tela com o detalhamento da rota.

Fonte: elaborado pelo autor.

Quadro 17 – Detalhamento do caso de uso Visualizar informações das coletas

UC05 – Visualizar informações das coletas	
Pré-condições	O dispositivo deve estar conectado à internet. O aplicativo deve estar no modo Consulta.
Cenário principal	01) O usuário seleciona o menu Detalhe. 02) O aplicativo apresenta o nome, pontos da rota, horários e dias em que a coleta é realizada.
Exceção 01	No passo 01, caso não haja conectividade com a internet, não apresentará as informações da rota.
Pós-condições	O aplicativo apresenta o menu de tipos de visualização.

Fonte: elaborado pelo autor.

Quadro 18 – Detalhamento do caso de uso Visualizar rotas de coleta em mapa

UC06 – Visualizar rotas de coleta em mapa	
Pré-condições	O dispositivo deve estar conectado à internet. O aplicativo deve estar no modo Consulta.
Cenário principal	01) O usuário seleciona o menu Mapa. 02) O aplicativo apresenta a rota de coleta de lixo traçada ao longo do mapa.
Exceção 01	No passo 01, caso não haja conectividade com a internet, não apresentará a rota da coleta no mapa.
Pós-condições	O aplicativo apresenta o menu de tipos de visualização.

Fonte: elaborado pelo autor.

Quadro 19 – Detalhamento do caso de uso Acompanhar execução da coleta em tempo real no mapa

UC07 – Acompanhar execução da coleta em tempo real no mapa	
Pré-condições	O dispositivo deve estar conectado à internet. O aplicativo deve estar no modo Consulta. O dispositivo da rota visualizada com o aplicativo no modo Rastreo.
Cenário principal	01) O usuário seleciona o menu Mapa, no horário em que a coleta de lixo está em execução. 02) O aplicativo apresenta um ícone representando o dispositivo do coletor que se movimenta ao longo da rota.
Exceção 01	No passo 02, caso não haja conectividade com a internet, não apresentará o ícone do dispositivo coletor.
Pós-condições	O ícone do dispositivo utilizado na coleta se movimenta até que o rastreo seja desabilitado.

Fonte: elaborado pelo autor.

São apresentados os casos de uso do módulo web no Quadro 20, Quadro 21, Quadro 22, Quadro 23 e Quadro 24.

Quadro 20 – Detalhamento do caso de uso Manter dispositivos de coleta

UCW01 – Manter dispositivos de coleta	
Cenário principal	01) O usuário informa o ID do dispositivo a ser utilizado para a coleta. 02) O usuário informa o nome do dispositivo.
Cenário alternativo 1	02.1) O usuário informa observações acerca do dispositivo (opcional).
Pós-condições	O cadastro é finalizado e fica disponível para consulta pelo aplicativo Android.

Fonte: elaborado pelo autor.

Quadro 21 – Detalhamento do caso de uso Manter rotas de coleta

UCW02 – Manter rotas de coleta	
Cenário principal	01) O usuário informa o nome da rota. 02) O usuário adiciona agendamentos das coletas na rota. 03) O usuário adiciona pontos a serem percorridos na rota.
Pós-condições	O cadastro é finalizado e fica disponível para consulta no modo Consulta do aplicativo Android.

Fonte: elaborado pelo autor.

Quadro 22 – Detalhamento do caso de uso Manter agendamentos das coletas na rota

UCW03 – Manter agendamentos das coletas na rota	
Cenário principal	01) O usuário informa o dia da semana em que ocorre a coleta na rota. 02) O usuário informa o horário inicial e final da coleta. 03) O usuário informa um dispositivo para o agendamento.
Cenário alternativo 1	03.1) O usuário informa observações acerca do dispositivo (opcional).
Pós-condições	O agendamento da coleta é cadastrado na rota. O cadastro da rota atualiza a lista de dias da semana em que a coleta é executada (depende da estruturação dos dados).

Fonte: elaborado pelo autor.

Quadro 23 – Detalhamento do caso de uso Manter dispositivos nos agendamentos

UCW04 – Manter dispositivos nos agendamentos	
Cenário principal	01) O usuário informa o ID do dispositivo a ser utilizado para o agendamento da coleta.
Pós-condições	O dispositivo é vinculado ao agendamento da rota.

Fonte: elaborado pelo autor.

Quadro 24 - Detalhamento do caso de uso Manter pontos a serem percorridos

UCW05 – Manter pontos a serem percorridos	
Cenário principal	01) O usuário informa a latitude do endereço que faz parte da rota. 02) O usuário informa a longitude do endereço que faz parte da rota. 03) O usuário informa a descrição do endereço que faz parte da rota.
Pós-condições	O ponto de percurso é adicionado na rota. O cadastro da rota atualiza a menor e maior latitude dos seus pontos (depende da estruturação dos dados). O cadastro da rota atualiza a menor e maior longitude dos seus pontos (depende da estruturação dos dados).

Fonte: elaborado pelo autor.

APÊNDICE B – Exemplo de resultado da Google Directions API

Este apêndice apresenta o resultado gerado pela Google Directions API, utilizando como endereço de partida R. Emílio Ninow, 155 - Escola Agrícola, Blumenau - SC, 89031-240 e endereço de destino R. Ernesto Jensen, 14 - Asilo, Blumenau - SC, Brasil, conforme Quadro 25.

Quadro 25 - Resultado da Google Directions API

```
{
  "geocoded_waypoints": [
    {
      "geocoder_status": "OK",
      "place_id": "ChIJ03MjSuke35QRmJpq0nJQcAU",
      "types": [
        "street_address"
      ]
    },
    {
      "geocoder_status": "OK",
      "place_id": "ChIJN-K8Veke35QRgvaqWPejUt0",
      "types": [
        "street_address"
      ]
    }
  ],
  "routes": [
    {
      "bounds": {
        "northeast": {
          "lat": -26.8883293,
          "lng": -49.0957337
        },
        "southwest": {
          "lat": -26.8889318,
          "lng": -49.0967446
        }
      },
      "copyrights": "Dados cartográficos ©2017 Google",
      "legs": [
        {
          "distance": {
            "text": "0,2 km",
            "value": 156
          },
          "duration": {
            "text": "1 min",
            "value": 42
          },
          "end_address": "R. Ernesto Jensen, 14 - Asilo, Blumenau - SC,
Brasil",
          "end_location": {
            "lat": -26.8889318,
            "lng": -49.0967446
          },
          "start_address": "R. Emílio Ninow, 155 - Escola Agrícola,
Blumenau - SC, 89031-240, Brasil",
          "start_location": {
            "lat": -26.8886721,
```

```

    "lng": -49.0957337
  },
  "steps": [
    {
      "distance": {
        "text": "0,1 km",
        "value": 145
      },
      "duration": {
        "text": "1 min",
        "value": 36
      },
      "end_location": {
        "lat": -26.8888361,
        "lng": -49.0967007
      },
      "html_instructions": "Siga na direção  

\u003cb\u003enoroeste\u003c/b\u003e na \u003cb\u003eR. Em\u003c  

Ninow\u003c/b\u003e em direção à \u003cb\u003eR. Alcindino de  

Campos\u003c/b\u003e",
      "polyline": {
        "points": "dubcDh_tjH_AfCCD?D@BDDHDLD\\Jf@J"
      },
      "start_location": {
        "lat": -26.8886721,
        "lng": -49.0957337
      },
      "travel_mode": "DRIVING"
    },
    {
      "distance": {
        "text": "11 m",
        "value": 11
      },
      "duration": {
        "text": "1 min",
        "value": 6
      },
      "end_location": {
        "lat": -26.8889318,
        "lng": -49.0967446
      },
      "html_instructions": "Continue para \u003cb\u003eR. Ernesto  

Jensen\u003c/b\u003e\u003cdiv style=\"font-size:0.9em\" \u003eO destino  

estará à direita\u003c/div\u003e",
      "polyline": {
        "points": "fvbcDjetjHPF"
      },
      "start_location": {
        "lat": -26.8888361,
        "lng": -49.0967007
      },
      "travel_mode": "DRIVING"
    }
  ],
  "traffic_speed_entry": [],
  "via_waypoint": []
}
],
"overview_polyline": {
  "points": "dubcDh_tjHcALC@HNJj@Px@R"
}

```

```
    },  
    "summary": "R. Emílio Ninow",  
    "warnings": [],  
    "waypoint_order": []  
  }  
],  
"status": "OK"  
}
```

Fonte: Google Directions API (2017b, p.1).

APÊNDICE C – Exemplos de resultados do módulo web de apoio

No Quadro 26 é apresentado o resultado aguardado pelo aplicativo em formato JSON, ao consultar dispositivos Android cadastrados para autenticação no servidor de aplicação.

Quadro 26 - Resultado esperado cadastro de dispositivo do módulo web

```
[
  {
    "id": "6392be568b3533bd",
    "name": "DISPOSITIVO_01",
    "information": "Dispositivo do emulador",
    "last_latitude": 0,
    "last_longitude": 0
  },
  {
    "id": "6434as56e9b53zo2",
    "name": "DISPOSITIVO_02",
    "information": "",
    "last_latitude": 0,
    "last_longitude": 0
  }
]
```

Fonte: elaborado pelo autor.

No Quadro 27 é apresentado o resultado aguardado pelo aplicativo em formato JSON, ao enviar um endereço que se deseja pesquisar a rota de coleta de lixo no servidor de aplicação.

Quadro 27 - Resultado esperado cadastro de rota do módulo web

```
[
  {
    "id": "0b67c7e4-6bfb-4438-82c5-63046004487d",
    "name": "ROUTE_01 - R. Benjamin",
    "latitude_min": -26.8909308,
    "latitude_max": -26.8873337,
    "longitude_min": -49.09752209999999,
    "longitude_max": -49.094960000000015,
    "week_days": "MONDAY,WEDNESDAY,FRIDAY,SATURDAY,",
    "points": [
      {
        "id": "4192d762-9f2f-41c4-9870-de027bd407ec",
        "name": "R. Benjamin Constant, 63 - Asilo, Blumenau - SC, Brasil",
        "latitude": -26.8873337,
        "longitude": -49.096755199999996
      },
      {
        "id": "4ba61f60-22a8-446a-b42f-3d4a80308058",
        "name": "R. Emilio Ninow, 241 - Escola Agrícola, Blumenau - SC, 89031-240, Brasil",
        "latitude": -26.8890365,
        "longitude": -49.094960000000015
      },
      {
        "id": "80f2000a-3196-488f-85fc-99d7080e50cb",
        "name": "R. Emilio Ninow, 155 - Escola Agrícola, Blumenau - SC, 89031-240, Brasil",
        "latitude": -26.8886192,
        "longitude": -49.095701500000002
      }
    ]
  }
]
```

```

    },
    {
      "id": "690fc696-a181-48bb-944f-fafbd0f80ae1",
      "name": "R. Ernesto Jensen, 14 - Asilo, Blumenau - SC, Brasil",
      "latitude": -26.8889187,
      "longitude": -49.09678050000002
    },
    {
      "id": "8bc7d833-535e-4ad9-9f28-fa367cf7f596",
      "name": "R. Aline, 24 - Escola Agrícola, Blumenau - SC, 89031-250,
Brasil",
      "latitude": -26.8907095,
      "longitude": -49.0961241
    },
    {
      "id": "78f5f1d4-0d3b-4ebc-bf66-64e614107443",
      "name": "R. Elise Jensem, 38 - Asilo, Blumenau - SC, 89031-260,
Brasil",
      "latitude": -26.8909308,
      "longitude": -49.097522099999999
    },
    {
      "id": "7ee187cb-1d3a-47be-8951-49557d8d2125",
      "name": "R. Benjamin Constant, 272, Blumenau - SC",
      "latitude": -26.8909308,
      "longitude": -49.097522099999999
    },
    {
      "id": "780d07dd-8334-4bdb-8020-82a222165be1",
      "name": "R. Benjamin Constant, 245 - Asilo, Blumenau - SC, Brasil",
      "latitude": -26.8888173,
      "longitude": -49.0970413
    }
  ],
  "schedules": [
    {
      "id": "b578a6d5-0bf5-4a2c-9e6c-7df78bf6b793",
      "week_day": "MONDAY",
      "initial_hour": "18:30",
      "final_hour": "19:30",
      "information": "",
      "id_device": "6392be568b3533bd"
    },
    {
      "id": "19882560-da7c-4382-b97a-e9ff812fc830",
      "week_day": "WEDNESDAY",
      "initial_hour": "18:30",
      "final_hour": "19:30",
      "information": "",
      "id_device": "6392be568b3533bd"
    },
    {
      "id": "4f7b7557-d4f1-4707-89ac-2cebec677f8f",
      "week_day": "FRIDAY",
      "initial_hour": "18:30",
      "final_hour": "19:30",
      "information": "",
      "id_device": "6392be568b3533bd"
    },
    {
      "id": "5ac19c81-328e-4c65-a18d-df3cb1065ef6",

```

```
"week_day": "SATURDAY",  
"initial_hour": "15:00",  
"final_hour": "16:00",  
"information": "",  
"id_device": "6434as56e9b53zo2"  
}  
]  
}  
]
```

Fonte: elaborado pelo autor.

APÊNDICE D – Telas de cadastro do módulo web de apoio

Na Figura 18 é apresentado a tela de cadastro de dispositivos do módulo web de apoio utilizado neste trabalho.

Figura 18 - Cadastro de dispositivos, módulo web de apoio

Cadastro de Dispositivos

ID (chave única do dispositivo):

Nome:

Observação:

Sem informações adicionais.

Fonte: elaborado pelo autor.

Na Figura 19 é apresentada a tela de cadastro de rotas do módulo web de apoio utilizado neste trabalho. Nela, são adicionados os agendamentos de coleta (dias da semana, horários, e respectivo dispositivo) e os pontos de referência da rota a ser utilizada.

Figura 19 - Cadastro de rotas, módulo web de apoio

Cadastro de rotas

Descrição:

Agendamentos	Pontos
Dia da semana: <input type="text" value=""/> Período: <input type="text" value="--:--"/> <input type="text" value="--:--"/> Dispositivo: <input type="text" value=""/>	Endereço: <input type="text" value=""/>
<input type="button" value="+ Adicionar"/>	<input type="button" value="+ Adicionar"/>
<ul style="list-style-type: none"> <input checked="" type="checkbox"/> Segunda-feira, 18:30h - 19:30h , DISPOSITIVO_01 (6392be568b3533bd) <input checked="" type="checkbox"/> Quarta-feira, 18:30h - 19:30h , DISPOSITIVO_01 (6392be568b3533bd) <input checked="" type="checkbox"/> Sexta-feira, 18:30h - 19:30h , DISPOSITIVO_01 (6392be568b3533bd) <input checked="" type="checkbox"/> Sábado, 15:00h - 16:00h , DISPOSITIVO_02 (6434as56e9b53zo2) 	<ul style="list-style-type: none"> <input checked="" type="checkbox"/> R. Benjamin Constant, 63 - Asilo, Blumenau - SC, Brasil <input checked="" type="checkbox"/> R. Benjamin Constant, 236 - Asilo, Blumenau - SC, Brasil <input checked="" type="checkbox"/> R. Emílio Ninow, 241 - Escola Agrícola, Blumenau - SC, 89031-240, Brasil <input checked="" type="checkbox"/> R. Emílio Ninow, 155 - Escola Agrícola, Blumenau - SC, 89031-240, Brasil <input checked="" type="checkbox"/> R. Ernesto Jensen, 14 - Asilo, Blumenau - SC, Brasil <input checked="" type="checkbox"/> R. Aline, 24 - Escola Agrícola, Blumenau - SC, 89031-250, Brasil <input checked="" type="checkbox"/> R. Elise Jensem, 38 - Asilo, Blumenau - SC, 89031-260, Brasil <input checked="" type="checkbox"/> R. Benjamin Constant, 272, Blumenau - SC <input checked="" type="checkbox"/> R. Benjamin Constant, 231 - Asilo, Blumenau - SC, Brasil
<input type="button" value="Adicionar"/> <input type="button" value="Atualizar"/> <input type="button" value="Deletar"/>	

Fonte: elaborado pelo autor.