

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E TECNOLOGIA
DO AMAZONAS**

CAMPUS MANAUS CENTRO

Ezequiel Sidney de Aguiar Brandão

**ESTIMULANDO A COLETA SELETIVA ATRAVÉS DA
GAMIFICAÇÃO E TECNOLOGIA**

Manaus, Amazonas – Brasil

2021

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E TECNOLOGIA
DO AMAZONAS
CAMPUS MANAUS CENTRO**

Ezequiel Sidney de Aguiar Brandão

**ESTIMULANDO A COLETA SELETIVA ATRAVÉS DA
GAMIFICAÇÃO E TECNOLOGIA**

**“Trabalho de Conclusão de Curso
apresentado à banca examinadora do
Curso Superior de Tecnologia de
Desenvolvimento e Análise de
Software do Instituto Federal de
Educação, Ciências e Tecnologia do
Amazonas – IFAM Campus Manaus -
Centro, como requisito para o
cumprimento da disciplina TCC II –
Projeto de Software”**

Professor M.Sc. Jorge Abílio Abinader Neto

12 /2021

Manaus, Am

Biblioteca do IFAM - Campus Manaus Centro

B816e Brandão,Ezequiel Sidney de Aguiar.

Estimulando a coleta seletiva através da gamificação e tecnologia /
Ezequiel Sidney de Aguiar Brandão. – Manaus, 2021.
53 p.: il. color.

Monografia (Graduação em Análise e Desenvolvimento de Sistemas).
– Instituto Federal de Educação, Ciência e Tecnologia do Amazonas,
Campus Manaus Centro,2021.

Orientador: Prof. MSc. Jorge Abílio Abinader Neto.

1. Informática. 2. Análise de Sistemas. 3. Desenvolvimento de
Sistemas. 4. Reciclagem I. Abinader Neto, Jorge Abílio. (Orient.) II.
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas. III.
Título.

CDD 004.21

Elaborada por Odimar Porto CRB 11/496

AGRADECIMENTOS

Agradeço a Deus e a espiritualidade amiga por inspirar as pessoas que estavam ao meu redor com boas ideias, pensamentos e conselhos. Pelo aprendizado concedido em cada momento da minha vida, tudo foi necessário, importante e contribuiu para a minha formação. Pelo conforto dos braços do consolador nos momentos mais difíceis, e por me dar olhos para enxergar algumas respostas que estavam dentro de mim.

Ao meu orientador Jorge Abílio Abnader, que foi um grande companheiro nessa jornada. Por sua resiliência, paciência, dedicação, colaboração, ideias e palavras.

Aos amigos José Silas Barros Brandão Júnior e Mayana Fernandes Veras, que contribuíram diretamente na concepção da ideia deste trabalho.

À associação dos catadores de Manaus da zona sul, a qual hoje em dia entrego os resíduos recicláveis que separo. Obrigado por sempre me receberem de forma acolhedora, sendo pacientes em ouvir e responder minhas perguntas.

À coordenação do curso, corpo docente, pedagogos, psicólogos e todos os servidores do IFAM, em especial aos servidores do DAIC e a pedagoga Adriana.

À minhas terapeutas Larissa Baptista e Zandra Valdez, não teria conseguido sem seu os cuidados delas.

Ao professor Francisco Januário da UFAM;

Aos colegas de trabalho;

Àos produtores de conteúdo confiáveis para internet. Em especial a Ana Lesnovski e Álvaro Borba cujos conteúdos na internet e seu livro ajudaram-me a sanar algumas dúvidas que eu ainda tinha sobre a confecção de trabalhos como este.

A todos os amigos, em especial a Dulcilene Colares, Andressa Silvestre e Leone dos Anjos. Sou grato pelos seus conselhos, pela paciência infinita, pelas experiências, por dedicarem um tempinho de suas vidas para ouvir minhas angústias ajudando assim a aliviá-las, pelas conversas profundas, pelas rotineiras, pelas fofocas, pelas risadas, pelos memes mais absurdos, e por todos os momentos ao longo desta caminhada.

Aos trigêmeos da casa, Doki Sarutobi, Perla Princesa Disney e Jojis Júnior Lula da Silva, meus pets que sempre estão dispostos a ajudar por serem fontes infinita de carinho.

À minha família, que me deu suporte necessário para que eu não precisasse largar os estudos. Aos nossos momentos de felicidades genuína e por estarem presentes nas minhas melhores lembranças da infância. Aos meus sobrinhos e primos mais novos, aos meus tios, tias e demais parentes e em especial aos meus irmãos José Silas Barros Brandão Júnior e Augusto de Aguiar Brandão.

E por fim à minha mãe, Edilamar Fernandes de Aguiar, que seu esforço e dedicação possibilitaram a mim e a meus irmãos a chance de estudar. Por ter me incentivado durante todo o curso, por sempre ter acreditado na conclusão deste trabalho e principalmente por nunca ter desistido de mim.

“É fato que no mundo ninja aqueles que quebram as regras são considerados lixo. Mas aqueles que abandonam seus companheiros são piores do que lixo”

Masashi Kishimoto

Resumo

Atualmente, por conta da cultura do consumo de produtos industrializados, o ser humano acaba produzindo, em média, pouco mais de 1 kg de resíduos sólidos por dia. Grande parte desses resíduos são compostos por materiais que, através do descarte adequado, poderiam ser colocados de volta ao ciclo produtivo. Entretanto, a quantidade de resíduo descartado de maneira responsável está muito longe da ideal, gerando problemas de saneamento nas cidades. Este trabalho apresenta um sistema digital que, utiliza como estratégia o conceito da gamificação, visando atrair a atenção e estimular diversos setores da sociedade à participação ativa no processo de descarte e coleta seletiva de resíduos na cidade de Manaus. Apresentaremos aqui a implementação desse sistema, seus módulos, suas versões em ambiente Web e Móvel, além da comunicação entre tais ambientes realizada através de um Web Service REST.

Palavras-Chave: Reciclagem, Gamificação; Desenvolvimento de software web; Aplicações móveis.

Abstract

Currently, due to the culture of industrialized products consumption, human beings end up producing, on average, just over 1 kg of solid waste per day. A large part of this waste is made up of materials that, through proper disposal, could be put back into the production cycle. However, the amount of residuum waste disposed of responsibly is far from ideal, generating sanitation problems in cities. This work proposes to develop a digital system that, using the gamification strategy, attracts attention, raises awareness and encourages many sectors of the society to actively participate in the process of disposal and selective waste collection in the city of Manaus. We will present here this system implementation, its modules, its versions in a Web and Mobile environment, in addition to the communication between such environments carried out through a Web Service REST.

Palavras-Chave: Recycling, Gamification; Web software development; Mobile applications.

Sumário

Capítulo 1 - Introdução	12
1.1 Problematização.....	13
1.2 Justificativa	14
1.3 Objetivos.....	16
1.3.1 Objetivo geral	16
1.3.2 Objetivos específicos	16
1.4 Demais Capítulos.....	16
Capítulo 2 - Fundamentação teórica.....	18
2.1 Reciclagem	18
2.2 Gamificação	19
2.3 Desenvolvimento de sistemas	20
2.3.1 Modelo Visão Controlador – MVC.....	20
2.3.2 AngularJS.....	20
2.3.3 Ionic	21
2.3.4 Node.js e Express.....	21
2.3.5 MongoDB e Mongoose	22
2.4 Sistemas correlatos	22
Capítulo 3 - Desenvolvimento	24
3.1 Introdução ao contexto do sistema	24
3.2 Principais papéis do sistema	26
3.2.1 Administrador	27
3.2.2 Coletor	27
3.2.3 Cidadão	27
3.2.4 Parceiro lojista.....	28
3.3 Principais componentes do sistema.....	28
3.3.1 Componente “Coleta”.....	30
3.3.2 Componente “Loja virtual”	37
3.3.3 Componente “Resgate de produto”	41
Capítulo 4 – Considerações finais	52
4.1 Trabalhos futuros	52
4.2 Lições aprendidas.....	52
4.3 Conclusão.....	53
Anexo I - Diagrama de classes geral do sistema e descrição	54

Anexo II – Diagrama de controle do componente “Resgate de produto” “função criar intenção de resgate”	57
Referências bibliográficas	58

Capítulo 1 - Introdução

“Saneamento básico” é uma expressão utilizada para designar um conjunto de medidas referentes à “captação, tratamento, e distribuição de água potável; a coleta, o tratamento e a disposição final de esgotos cloacais e do lixo, bem como a drenagem urbana” (Krieger et. al., 2008).

Dentre essas ações, este trabalho aborda aspectos relacionados ao tratamento de lixo utilizando principalmente dados referentes a cidade de Manaus AM. Neste momento é importante ressaltar que se seguiu a recomendação dos especialistas no assunto Zanin e Mancini (2015), optando por utilizar a palavra resíduo ao invés de lixo atenuando a ótica negativa que a segunda palavra tende a representar, para uma perspectiva que valoriza o material descartado.

A produção de resíduos sólidos tornou-se um agravante comum nos aglomerados urbanos de todos os tipos e tamanhos. Segundo Nani (2007) pesquisas apontam que o ser humano produz diariamente pouco mais de um quilo de resíduos sólidos por dia. O autor destaca ainda que em 2007 a produção anual de resíduos sólidos no planeta chegou a 400 milhões de toneladas (Nani, 2007).

De acordo com Zanin e Mancini (2015) o modelo linear de desenvolvimento, que consiste em extrair recursos naturais, transforma-los em produtos, consumi-los e descartá-los, é um modelo antigo e insustentável que tendia a ser substituído por um modelo com base na reutilização e gestão dos resíduos.

Reutilizar resíduos sólidos promove benefícios ambientais uma vez que, além da redução de detritos descartados irregularmente, ocorre também a redução da exploração de recursos naturais, bem como, menor consumo de energia utilizada no processo (Zanin e Mancini, 2015).

No Brasil a Lei Nº 12.305, de agosto de 2010 instituiu a política nacional de resíduos sólidos. O Art. 3º inciso VII define que a “destinação final ambientalmente adequada” dos resíduos deve inclui a reutilização, reciclagem, compostagem, aproveitamento energético entre outras destinações de modo a evitar danos à saúde pública e a minimizar impactos ambientais adversos (Brasil, 2010).

Os produtos destinados diretamente para a reutilização passam apenas por uma triagem e limpeza para serem postos de volta ao uso. Já na reciclagem ocorre um processo industrial para converter o resíduo ou sucata em matéria prima secundária (Nani, 2007).

Ambas as destinações são antecedidas pelo procedimento de separação de resíduo como: orgânico, inorgânico, molhado, seco e em seguida por tipo de material (Nani, 2007). Este processo de separação de materiais foi denominado “coleta seletiva” segundo a lei Nº 12.305 de agosto de 2010 (Brasil, 2010).

Sendo assim, este trabalho descreve a ideia e construção de um sistema tecnológico que emprega conceitos da Gamificação digital, e foi desenvolvido principalmente para ser uma ferramenta que estimule o uso da coleta seletiva.

1.1 Problematização

A coleta seletiva é uma ação que possibilita a reciclagem e reutilização de resíduos sólidos, a sua não realização é um fator de complicação quanto a recolocação desses materiais de volta ao ciclo produtivo e conseqüentemente impacta o meio ambiente aumentando a exploração de recursos naturais, energéticos e a quantidade final de resíduos sólidos produzidos (Zanin e Mancini, 2015).

Essa ação está presente dentre as regulamentações contidas na lei Nº 12.305 de agosto de 2010, que instituiu a política nacional de resíduos sólidos no Brasil. Nela também foi determinado em seu Art.1º § 1º que a grande maioria dos agentes da sociedade participantes da produção, descarte ou gestão de resíduos são responsáveis por cumprir suas normas (Brasil, 2010).

Segundo dados do relatório de gestão da Secretaria Municipal de Limpeza Urbana – SEMULSP (2020), no ano de 2019 a cidade de Manaus AM destinou para reciclagem aproximadamente 12,5 mil toneladas de resíduos, atingindo o pico percentual da sua série histórica iniciada no ano de 2013 (SEMULSP, 2020).

No entanto, em 2019, mais de 576 mil toneladas de resíduos sólidos domiciliares foram descartados via coleta de resíduos comum, e acabaram sendo enviados para o aterro sanitário da cidade (SEMULSP, 2020).

O descarte de forma comum mistura e compacta resíduos recicláveis com não recicláveis, não promove a segregação e condicionamento necessários para a reutilização, inviabilizando assim a possível reciclagem desses resíduos (Zanin e Mancini, 2015).

Para maior alcance no uso da coleta seletiva em Manaus AM, a SEMULSP utiliza estratégias como a coleta porta a porta em 13 bairros da cidade, ou a coleta em Pontos de Entrega Voluntária (PEVs) para reter esses materiais antes da coleta tradicional. Além disso algumas associações de catadores também funcionam como PEVs (SEMULSP, 2020).

Compondo essa estratégia a secretaria também promove, através da Comissão Especial de Divulgação da Política de Limpeza Pública – CEDOLP, atividades de conscientização ambiental com a população manauara. Sendo que somente em 2019 foram realizadas 1.296 ações com alcance de 98.509 pessoas sensibilizadas (SEMULSP, 2020).

Apesar dessas estratégias de conscientização está em prática desde 2013, o percentual de resíduos destinados a reciclagem em 2019 foi de somente 2,16% comparado ao valor total de resíduos vindos da coleta domiciliar (SEMULSP, 2020).

Nani (2007) chama a atenção para a tendência de que os benefícios da reutilização de resíduos, e os prejuízos causados pela falta dela, tendem a ser dificilmente assimilado pela população e reforça a importância de elementos didáticos e midiáticos para impulsionar a questão.

Considerando tal cenário, poderia um sistema tecnológico funcionar como elo entre os agentes da sociedade participantes da produção, descarte ou gestão de resíduos que em parceria cooperariam para preservação do meio ambiente através da coleta seletiva? A gamificação digital pode ser um atrativo às ações de conscientização ambiental? Ela pode cooperar na adesão da coleta seletiva e no engajamento da sociedade em causas ecológicas? Qual seria o valor de um sistema tecnológico que estimule o processo de descarte e coleta seletiva? Nos próximos capítulos deste trabalho é apresentada uma solução tecnologica que objetifica responder essas perguntas.

1.2 Justificativa

Segundo Burke (2015) a Gamificação é o nome utilizado para definir o conjunto de ferramentas contidas em experiências digitais com mecânicas de jogos como pontuação,

progresso, premiação, entre outros mecanismos, que combinadas são utilizadas para motivar e engajar pessoas a atingirem seus objetivos.

Burke (2015) destaca em sua obra alguns casos de sucesso relacionados ao uso do conceito na educação, saúde, negócios e demais setores. O autor informa ainda que se a área em questão utilizar mecanismos digitais em suas atividades, essa torna-se candidata a receber uma solução gamificada em algum momento (Burke, 2015). Dessa maneira este trabalho justifica o uso da Gamificação como ferramenta de engajamento da sociedade em torno da coleta seletiva de resíduos sólidos.

É válido ressaltar que o uso de dispositivos eletrônicos no Brasil apresentam dados satisfatórios que endossam a utilização da tecnologia para soluções de gestão e conscientização ambiental. Segundo dados da Fundação Getúlio Vargas – FGV referentes ao ano de 2020, o Brasil conta com 440 milhões de dispositivos digitais em uso, desses 242 milhões são *smartphones* (FGV, 2020).

Além disso, a utilização da internet no Brasil por pessoas com 10 anos de idade ou mais, em grandes regiões, foi estimada em 74,7% no último trimestre de 2018, segundo revelou dados da Pesquisa Nacional por Amostra de Domicílios Contínua – PNAD (IBGE, 2018). Tal cenário exhibe a aceitação que a tecnologia tem no país e ampara a proposta deste em desenvolver um sistema tecnológico.

Não são apenas benefícios ambientais que a coleta seletiva e o reaproveitamento de resíduos sólidos tendem a promover. Segundo Nani (2007) a indústria da reciclagem produz diversos materiais como papel, folhas de alumínio, lâminas de borracha e ainda gera energia com a combustão de materiais, nesse contexto os resíduos sólidos torna-se fonte renda ao invés de ser um problema sanitário.

Conforme mencionado no tópico anterior, os dados contidos no relatório de gestão da SEMULSP (2020) sugerem que o setor possui potencial de crescimento em Manaus. Em escala nacional, Nani (2007) destaca que apesar do Brasil ser um dos países que mais se recicla latinhas de alumínio, estima-se que o mesmo desperdice anualmente cerca de R\$ 4,6 bilhões por não reciclar tudo que poderia. Sendo assim, estimular a segregação de resíduos também demonstra capacidade em valorizar os profissionais de limpeza urbana e coleta seletiva, suas cooperativas e indústrias do setor que se beneficiariam com o aumento da oferta de matéria prima reciclável.

Foram utilizados principalmente dados referentes a realidade de Manaus, capital do Amazonas, por ser banhada por rios, igarapés e ainda está situada em meio a floresta amazônica. Entretanto, a gestão de resíduos é regulamentada em todas as unidades da federação do país pela lei Nº 12.305, de agosto de 2010. Dessa maneira, este trabalho desenvolveu seu cenário de forma genérica buscando atender qualquer cidade brasileira, e com poucos ajustes, ser escalonado para gerenciar esta modalidade de preservação do meio ambiente em âmbito nacional.

Com base nos dados, cenários, na legislação vigente, nas necessidades, nos benefícios sociais, ambientais e econômicos neste apresentados, somados às definições da Gamificação e ao dados referentes ao uso de dispositivos tecnológicos. Este trabalho justifica o desenvolvimento de um *software* gamificado como ferramenta alternativa de engajamento de toda a sociedade em torno da coleta seletiva e reutilização de resíduos sólidos.

1.3 Objetivos

1.3.1 Objetivo geral

Criar um sistema de software que empregue tecnologias para o desenvolvimento de aplicações web, web services e aplicações móveis para *smartphones*, que utilize a Gamificação para promover o engajamento da sociedade em torno da participação ativa do processo de coleta seletiva de resíduos sólidos em cidades de todos os tamanhos e qualquer densidade populacional, contribuindo com a conscientização ambiental e preservação da natureza.

1.3.2 Objetivos específicos

- Criar ecossistema que interligue cidadãos, catadores, cooperativas, empresas de reciclagem, indústrias, lojistas e poder público em prol do processo de descarte, coleta seletiva e logística reversa;
- Utilizar técnicas de Gamificação para incentivar a sociedade a participar do processo de descarte e coleta seletivo de resíduos sólidos;
- Facilitar o acesso a dados e a transparência referente a coleta seletiva, uma vez que os mesmos estarão cadastrados digitalmente;

1.4 Demais Capítulos

Nesse momento foram oferecidas as principais problemáticas do tema, justificadas as motivações contidas no trabalho, bem como traçados seus objetivos.

No próximo capítulo será contemplada as principais referências bibliográficas utilizadas como base no desenvolvimento do sistema descrito neste trabalho.

Na sequência, o capítulo 3 aborda a construção do contexto geral, a arquitetura utilizada, e as funcionalidades desenvolvidas com intuito de atender os objetivos propostos neste.

Por fim, no capítulo 4 é apresentado os resultados alcançados, as lições aprendidas, trabalhos futuros e considerações finais.

Capítulo 2 - Fundamentação teórica

2.1 Reciclagem

De acordo com Nani (2007) a palavra reciclagem foi introduzida no vocabulário internacional no fim da década de 80, onde *recycle* significa “re” de repetição e “cycle” de ciclo. O autor ressalta que, segundo estudos, cada ser humano produz em média um quilo de resíduo sólido por dia e que o modo mais simples de iniciar este processo é através da separação do lixo orgânico do inorgânico posteriormente classificando-os por tipo (Nani, 2007).

Nani (2007) afirmar ainda que o processo de reciclagem é diferente do processo de reutilização, sendo que nesse primeiro é necessário um processo de transformação do resíduo em matéria prima secundária, ambos trazem benefícios ao meio ambiente e são antecedidos pelo processo de separação dos resíduos. O autor ressalta ainda que deve-se levar em consideração a análise do ciclo de vida do produto para a decisão de qual descarte aplicar (Nani, 2007).

Coleta seletiva foi definida pela lei Nº 12.305, de agosto de 2010 como o processo de recolhimento de resíduos devidamente selecionados e condicionados por suas características. Esta lei instituiu regras, definições, instrumentos, diretrizes, planos e responsabilidades referentes à gestão de resíduos sólidos no país (Brasil, 2010).

O Art.1º § 1º da regulamentação referida, atribuiu a responsabilidade no cumprimento desta lei à “... pessoas físicas ou jurídicas, de direito público ou privado, responsáveis, direta ou indiretamente, pela geração de resíduos sólidos e as que desenvolvam ações relacionadas à gestão integrada ou ao gerenciamento de resíduos sólidos” (Brasil, 2010).

Em seu Art. 33º essa legislação obrigou empresas a realizarem a estruturação e implementação da logística reversa, que seria o retorno dos resíduos gerados após o consumo de alguns produtos que essas produzem, de forma independente do serviço público de limpeza urbana (Brasil, 2010).

A logística reversa foi inicialmente implantada em somente alguns dos resíduos definidos no Art. 33º, o restante foram previstos a serem contemplados futuramente baseados no Art. 56º da mesma lei (Brasil, 2010). Neste último Caput citado está contido o cronograma

de logística reversa que avançou em 2017 e em 2020 mediante aos decretos Nº 9.177 e Nº 10.240 respectivamente e possivelmente entrará em pleno vigor no ano de 2022.

Dentre os principais problemas que são encontrados para a realização do processo de reciclagem, de acordo com Zanin e Mancini (2015) são a dimensão continental do país, falta de preocupação do mercado consumidor próprio, o sistema predominante de coleta e descarte de resíduos pós consumo que ao misturar recicláveis de não recicláveis acaba inutilizando o material que poderia ser reaproveitado, a má reputação do material reciclado, a dificuldade de abastecimento frequente das indústrias, e a presença de muitas indústrias de pequeno porte com pouca tecnologia associada a alta carga tributária.

Apesar disso o mercado da reciclagem também visa lucro, segundo Zanin e Mancini (2015) a indústria da reciclagem adota alternativas a sistemas de reciclagem convencionais, que possam fornecer produto de valor agregado maior, através de pesquisas, análises de custos e desenvolvimento.

2.2 Gamificação

Busarello et al. (2014) entende a Gamificação “como estímulo à ação de se pensar sistematicamente [...], com intuito de resolver problemas, melhorar produtos, processos, objetos e ambientes com foco na motivação e no engajamento”, sendo estes aspectos fundamentais para construção do conhecimento dos sujeitos.

Para Burke (2015) um projeto gamificado possui mecanismos de jogos como placares, distintivos, pontos e outros. Tais elementos representam para os usuários experienciar a sensação de conquista dentro de um ambiente digital. Essas sensações servem para engajar e motivar os usuários de diferentes maneiras “A gamificação vem sendo utilizada no sentido de alterar comportamentos, desenvolver habilidades e impulsionar inovações” (Burke, 2015).

Alguns elementos contidos na Gamificação promovem engajamentos de diferentes maneiras como o engajamento transacional e o engajamento emocional. De acordo com o *Chartered Institute of Personnel and Development* – CIPD (Instituto de Desenvolvimento Pessoal) o engajamento transacional é aquele quando o indivíduo realiza uma atividade para receber algo em troca. Já o engajamento emocional está ligado ao desejo do indivíduo de fazer algo além do que é esperado para em troca receber mais, como um contrato psicológico profundo e compensador (Burker, 2015).

2.3 Desenvolvimento de sistemas

De modo geral a estrutura do sistema deste trabalho é baseada na arquitetura de três camadas, composta pela aplicação cliente, servidor e base de dados, onde a camada cliente realiza as chamadas ao servidor que as processa e consulta a base de dados quando necessário (Fowler, 2018).

As principais ferramentas utilizadas no desenvolvimento da aplicação cliente foram o AngularJS e o IONIC. No servidor destacam-se o Express, Mongoose e a plataforma Node.js. Por fim o MongoDB foi escolhido para o banco de dados. Nos próximos subtópicos serão detalhados os principais conceitos e características dessas tecnologias.

2.3.1 Modelo Visão Controlador – MVC

O MVC é uma abordagem utilizada no desenvolvimento de softwares que separa os objetos Modelo, Visão e Controlador objetivando aumentar a flexibilidade e reutilização de códigos (Gamma, 2008).

Seu objeto Modelo representa os dados da aplicação, a Visão a tela que reflete os dados do Modelo e permite a interação com o usuário. Já o objeto Controlador é responsável pela interação entre Visão e Modelo (Gamma, 2008).

2.3.2 AngularJS

AngularJS é um *framework* JavaScript mantido pelo Google baseado no modelo MVC utilizado para o desenvolvimento de páginas web dinâmicas e robustas no lado cliente (Pereira, 2014). O *framework* fornece diretivas e módulos com funcionalidades específicas que combinadas facilitam o desenvolvimento de sistemas web do lado do cliente (Almeida, 2018).

O resultado obtido com o AngularJS são *Single Page Application* – SPA, que são aplicações para o navegador que funcionam sem a necessidade de serem recarregadas (Pereira, 2014). Dessa maneira o AngularJS utiliza uma página principal (*index.html*) e páginas parciais (*partials*) que serão carregadas conforme a demanda do usuário (Almeida, 2018).

No framework o módulo `NgRoute` é o que possui a funcionalidade de criação de rotas internas do AngularJS, as rotas são responsáveis pela exibição das *partials* sem que estas recarreguem (Almeida, 2018). Dentro desse módulo o objeto `$routeProvider` permite a configuração da rota interna passando a URL que ela terá, um template que é a *partial* em HTML, e um *controller* em Javascript {explicitar MVC} (Almeida, 2018).

As solicitações para o servidor ficam por conta dos objetos `$http` ou `$resource`, ambos realizam chamadas assíncronas Ajax e utilizam o padrão *promise* do JavaScript para tratar o retorno do serviço consumido (Almeida, 2018).

2.3.3 Ionic

É um framework para o desenvolvimento de aplicações móveis híbridas que utiliza as tecnologias básicas do desenvolvimento web como HTML, CSS e JavaScript para gerar aplicações nativas na plataforma desejada (Gois, 2017). O *Cordova* é seu o *framework* interno que converte os códigos *web* para as plataformas *mobile*, além disso o Ionic também utiliza o AngularJs (Gois, 2017).

2.3.4 Node.js e Express

Node.js é uma plataforma utilizada para o desenvolvimento de *web services* em Javascript criada sob o ambiente de execução do Chrome. Seu gerenciador de pacotes npm (*Node package manager*), permite aos programadores a utilização das bibliotecas desenvolvidas pela comunidade Node.js (Almeida, 2018).

O Express é um *framework* para a plataforma Node.js que utiliza *middlewares* para expandir as funcionalidades de um servidor Node.js comum. Facilitando assim a criação de *web services* RESTfull que utilizam os verbos GET, POST, PUT e DELETE para implementação de rotas que serão consumidas pelas aplicações cliente (Almeida, 2018).

Middlewares são funções que lidam com as requisições. Uma requisição pode ser associada a uma pilha de *middlewares* cada um com sua responsabilidade como segurança, auditoria e etc (Almeida, 2018).

2.3.5 MongoDB e Mongoose

MongoDB é um banco de dados NoSQL que permite consultar e armazenar dados em arquivos utilizando um formato semelhante ao JSON. Por se tratar de um banco de dados fracamente tipado, a validação e integridade dos dados ficam a cargo da implementação da aplicação (Almeida, 2018).

O Mongoose é um framework da plataforma Node.js que auxilia a criação, armazenamento e consulta dos dados NoSQL do MongoDB, fornecendo ainda uma solução baseada em esquemas para modelar os dados da aplicação. Ele uma camada em torno do driver do MongoDB que gerencia relacionamentos e executa validações (Almeida, 2018).

2.4 Sistemas correlatos

Atualmente não é difícil encontrar aplicações que utilizam os conceitos de gamificação, Burke (2015) comenta em seu livro que a técnica é utilizada em sistemas de empresas e softwares que ajudam crianças em tratamento de câncer. Por conta disso alguns sistemas cujo contexto são diferente do abordado neste trabalho são considerados correlatos, dois bons exemplos de sistemas gamificados que inspiraram parte da solução deste trabalho foram como o jogo “*Pokemon Go*”, que embora seja um jogo, estimula seus usuário a caminhar e a visitar lugares históricos e até mesmo o partes do aplicativo de banco virtual “Mercado Pago”.

No jogo “*Pokemon Go*”, em seu menu de usuário, é exibido várias medalhas que o usuário pode conquistar se fizer determinadas missões, como ir a uma certa quantidade de vezes a um ponto turístico, histórico ou ambiente público. O usuário determina quais missões e quando irá cumpri-las. O sistema deste trabalho utilizou como modelo de menu do jogo, exibindo informações de progresso do usuário junto à medalhas que para serem conquistadas precisam que o usuário realize determinadas ações dentro do contexto do sistema.

Já a funcionalidade “gerar intenção de resgate”, relatada no capítulo 3 deste trabalho, baseou-se na funcionalidade “abastecimento de veículo” do aplicativo de banco virtual “Mercado Pago”. As ações assemelham-se por se tratar da compra (ou resgate) de um produto em uma loja física por intermédio de um aplicativo virtual do *smartphone* do usuário. Na funcionalidade “abastecimento de veículo” no “Mercado Pago” o usuário deve informar pelo aplicativo o número exibido na bomba de gasolina do posto o qual pretende abastecer seu

veículo, informar o valor que deseja abastecer e em seguida a bomba é liberada. Ao término do abastecimento o valor é debitado da conta do usuário. A utilização do código do local foi utilizada como modelo na funcionalidade “gerar intenção de resgate”, quando o usuário informa o código do local de troca ele comunica ao sistema onde está tentando realizar a troca dessa maneira o sistema pode debitar o produto do exatamente do estoque do local informado e fazer outras verificações caso necessárias.

Capítulo 3 - Desenvolvimento

3.1 Introdução ao contexto do sistema

A cidade de Manaus AM é uma metrópole banhada por rios e rodeada pela fauna e flora da floresta Amazônica. O modo que seus habitantes se relacionam com os resíduos que produzem podem gerar sérios impactos ambientais às riquezas naturais da região. Por conta dessas características a cidade foi selecionada como delimitação regional neste trabalho. A figura 1 exhibe o Igarapé do Quarenta, poluído por diversos materiais dentre estes resíduos sólidos. Esse igarapé fica localizado na zona sul da cidade de Manaus e deságua no rio Negro.



Figura 1 – Situação do igarapé do quarenta em Manaus 08/09/2021 Fonte: o próprio

A SEMULSP é o órgão da capital do Amazonas responsável pela gestão de resíduos sólidos. Dentre as modalidades de manejo de resíduos descritas no relatório anual de gestão da secretaria, este trabalho focou inicialmente na “Coleta Domiciliar”, modalidade mais utilizada, que se refere aos resíduos recolhidos em casas, pequenas indústrias, comércio, bancos e escolas (SEMULSP, 2020).

Nesse ponto é importante destacar que a mistura de materiais recicláveis com não recicláveis inviabiliza a reutilização de alguns materiais (Zanin e Mancini, 2015). Por esse motivo, a estratégia inicial proposta neste trabalho é incentivar o descarte de resíduos sólidos via coleta seletiva antes da coleta comum e para isso foi utilizado os mecanismos de engajamento emocional e transacional da Gamificação.

O engajamento emocional foi utilizado principalmente quando o usuário leva seus resíduos recicláveis para um “Ponto de Coleta”, neste momento ele receberá “Pontos de Experiência” e “Moedas Digitais” referentes à quantidade de resíduos doada. Essa experiência desbloqueará novos níveis, desafios, medalhas e outros elementos alusivos à conquista e ao progresso do usuário. Paralelo a isso o engajamento transacional foi representado neste sistema quando as “Moedas Digitais” são trocadas por produtos dentro do aplicativo.

A troca de “Moedas Digitais” por produtos são realizadas na “Loja Virtual”, área da aplicação onde são exibidos os produtos disponíveis para troca, suas características, seu valor em “Moedas Digitais”, e o local para fazer a retirada. Empresas, lojas, indústrias, marcas, órgãos do setor público e sociedade em geral, que desejam contribuir com esta modalidade de coleta seletiva poderão ser parceiros do projeto doando produtos para abastecer a “Loja Virtual”. O resgate dos produtos serão realizados nos espaços físicos desses parceiros ou somente via aplicação em caso de benefícios intangíveis como descontos ou códigos promocionais.

Considerando o Art. 33º da lei Nº 12.305, de agosto de 2010 que obriga algumas empresas a realizarem a logística reversa, ação de recolher os resíduos gerados após o consumo dos produtos que elas produzem independente do serviço público de limpeza, e ainda levando em conta o avanço do cronograma de implementação da logística reversa, definido no Art. 56 da mesma lei (Brasil, 2010). Este trabalho considera as empresas alvo da logística reversa fortes candidatas à parceiras doadoras de produtos à “Loja Virtual” visando cumprir a legislação, incentivar a coleta seletiva e ainda divulgar sua marca.

Os “Pontos de Coleta” são definidos neste trabalho como estruturas físicas que ficarão espalhadas pela cidade e são preparadas para receber, pesar, e armazenar temporariamente os resíduos recicláveis. Eles devem contar com pelo menos um responsável. Os Pontos de Entrega Voluntária – PEVs existente na cidade de Manaus (SEMULSP, 2020), podem ser adaptados para atender a estrutura dos “Pontos de Coleta”. A figura 2 exibe um ponto de coleta seletiva semelhante ao descrito neste parágrafo.



Figura 2 - Ponto de coleta "Punto verde" do projeto "Ciudad verde" em 2017. Fonte: o próprio

O responsável pelo “Ponto de Coleta” possui a função de avaliar, classificar, contabilizar e receber de forma física e digital os resíduos doados, gerando os “Pontos de Experiência” e “Moedas Digitais” para os doadores de resíduo. Esta ação também alimenta a base de dados referente a quantidade e tipo de materiais armazenadas no “Ponto de Coleta”, gerando assim a informação que indica o melhor momento para se recolher os resíduos armazenados.

Deste modo o cenário principal deste sistema consiste em possibilitar o acúmulo de “Pontos de Experiência” e “Moedas digitais” para quem depositar seus resíduos sólidos em um “Ponto de Coleta”. As “Moedas Digitais” possuem valor real dentro do sistema podendo serem trocadas por produtos doados por parceiros, e os “Pontos de Experiência” disparam elementos digitais alusórios a conquista. Os produtos serão doados por parceiros que visam colaborar com a coleta seletiva.

3.2 Principais papéis do sistema

Este tópico descreve o perfil dos agentes participantes do contexto definido para o sistema deste trabalho. Esses foram selecionados com base no Art.1º § 1º da lei Nº 12.305 de agosto de 2010, que responsabiliza a observância desta a lei todos os agentes da sociedade que por meio de suas atividades, incluindo o consumo, gerem resíduos sólidos (Brasil, 2020). Dessa maneira os principais papéis definidos no contexto deste sistema foram definidos conforme os tópicos abaixo.

3.2.1 Administrador

Possui o papel de decidir, aprovar ou não, inserir, atualizar ou deletar logicamente do sistema os atores Coletor e Parceiro Lojista. Da mesma forma ele gerencia os tipos de materiais, as unidades de medida e os “Pontos de Coleta” que o sistema irá trabalhar. Podendo também vincular ou revogar vínculo entre Coletor e “Pontos de Coleta”.

Por ser responsável por decisões estratégicas referentes à gestão geral do ambiente do sistema, o papel na sociedade que este perfil pode representar depende de qual cenário o sistema funcionará. No caso da aplicação rodar numa cidade este papel poderá refletir o gestor da secretaria de limpeza ou o próprio prefeito sempre ficando a cargo do aplicador do sistema definir o Administrador mais apropriado.

3.2.2 Coletor

Perfil responsável pelo cadastro, avaliação e medição de quantidade das doações de resíduos sólidos recebidas no “Ponto de Coleta” o qual esteja vinculado. Este perfil busca atender aos catadores, coletores membros ou não de cooperativas, às associações de coleta seletiva, aos profissionais de limpeza, à secretaria de limpeza pública e quaisquer outros agentes dispostos a se comprometer em participar do contexto do sistema dessa forma.

3.2.3 Cidadão

Por possuir um papel estratégico fundamental no contexto criado para este trabalho, este foi o ator inicialmente selecionado como alvo das estratégias de Gamificação desenvolvidas no sistema. Ele fica encarregado em separar os resíduos sólidos recicláveis que produz e depositá-los em algum “Pontos de Coleta”. Essa ação lhe gera bonificações dentro da aplicação que podem ser trocadas por produtos dentro do aplicativo. Estes produtos serão resgatados no local em que forem disponibilizados.

Este perfil reflete qualquer membro da sociedade que destine resíduos à coleta comum (ou coleta domiciliar). Conforme o relatório anual de gestão da SEMULSP (2020), em Manaus, os atores candidatos a assumirem esse perfil seriam residentes de qualquer domicílio, pequenas indústrias, comércio, bancos e escolas.

3.2.4 Parceiro lojista

Usuário responsável por incentivar esta modalidade de coleta seletiva disponibilizando produtos para troca dentro da “Loja virtual” do sistema. Também realiza a entrega dos produtos trocados ao “Cidadão”. Este perfil visa representar indústrias que necessitem ou não realizar a logística reversa, lojas, empresas, entidades da administração públicos ou pessoas físicas.

3.3 Principais componentes do sistema

A nomenclatura componente, que foi adotada nesta etapa, refere-se ao sentido de compor algo, neste caso, compor o contexto geral do sistema. Durante as fases de concepção da ideia e análise deste trabalho, foram atribuídas prioridades na implementação de três componentes do contexto geral do sistema, pois o desenvolvimento desses componentes resultaria nas funcionalidades mínimas necessárias para o contexto geral do software.

Estes componentes, “Componente Coleta”, “Componente Loja Virtual”, “Componente Resgate de Produto” serão detalhados neste tópico. Porém (parei aqui) (como ficará esta organização?)

Dentro de cada um desses componentes existe um conjunto de funcionalidades que interagem e trocam informações entre si. Essas funcionalidades estão espalhadas pela arquitetura *RestFull*, adotada para este sistema, e suas interações criam um agrupadas dão sentido aos contextos que foram nomeadas, como por exemplo a que agrupadas dão sentidos a contextos Sendo assim, cada um dos componentes principais possuem um conjunto de funcionalidades que atuando em conjunto dão forma a um contexto dentro do sistema.

Foram utilizados Para descrever esses componentes do sistema foram utilizados melhor exemplificar ComPara a descrição dos componentes fora utilizadas Foi utilizado diagramas de caso de uso para descrever Este tópico contém Esse contexto Essas funcionalidades DPara detal o funcionamento do software desenvolvido serão apresentadas três principais partes de

funcionalidades que o mesmo implementa. desobre o funcionamento e da nomenclatura que se utilizou para apresentar o funcionamento do sistema desenvolvido neste o separamos em três partes que se assemelham contextualmente. Essas partes juntas compõem Há três cenários principais que serão abordados neste momento. O primeiro deles representa as interações entre o “Cidadão”, “Componente Coleta” e “Coletor”. Onde o “Cidadão” doa seus resíduos recicláveis em um “Ponto de Coleta”, o “Coletor” responsável pelo ponto cadastra as doações e o sistema gera pontuação e “Moedas digitais” para o cidadão. Estas interações estão representadas abaixo pela Figura 3.

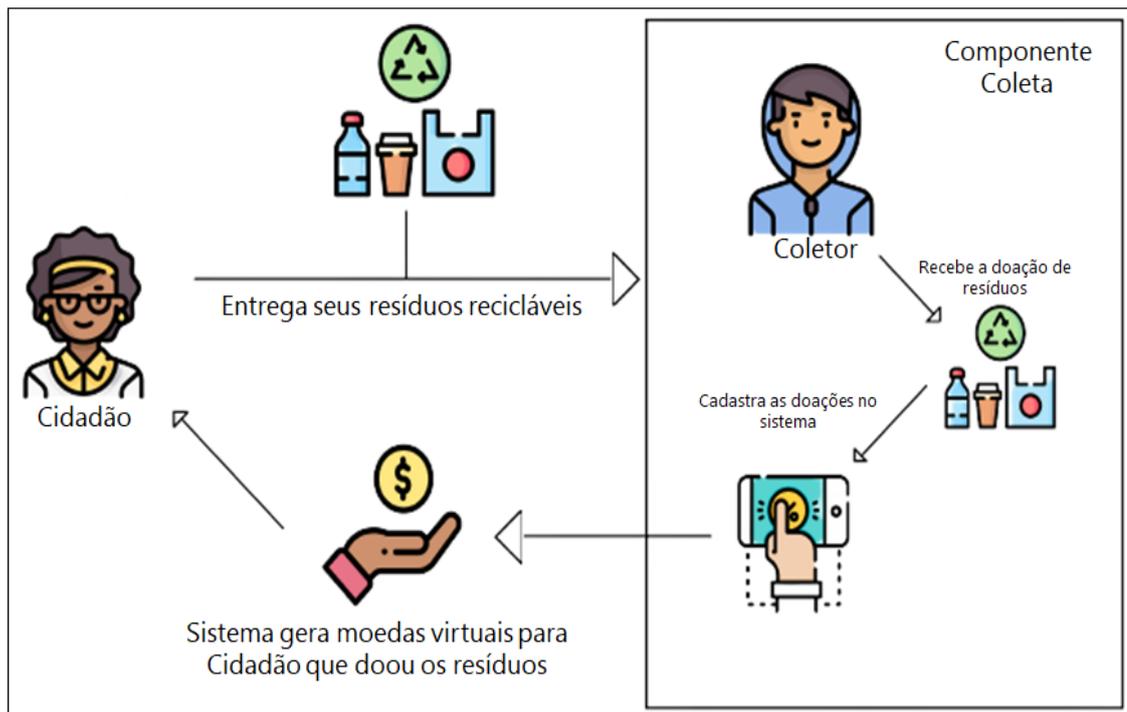


Figura 3 - Interação entre "Cidadão" e "Ponto de Coleta". Fonte: o autor; ícones: Flaticon

O segundo cenário representa as ações realizadas pelo “Cidadão” para trocar moedas por produtos e posteriormente resgatá-los na loja parceira que o disponibilizou, conforme a Figura 4 abaixo. Somados, esses cenários representam as principais funcionalidades do contexto do sistema proposto e desenvolvido neste trabalho.

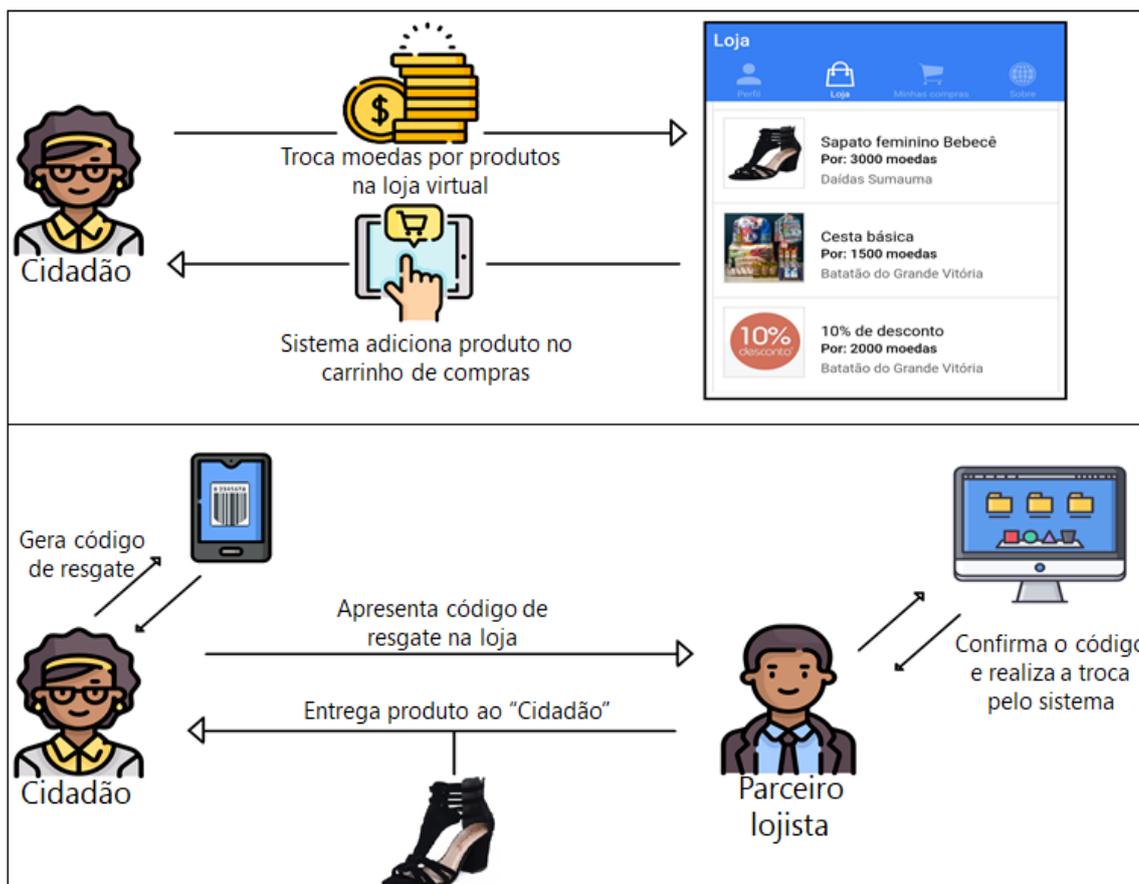


Figura 4 - Troca e resgate de produtos no sistema. Fonte: o autor; ícones: Flation

Nos tópicos dessa seção serão descritos, exibidos os casos de uso e detalhadas as funcionalidades mais complexas ou interessantes dos componentes que formam a parte essencial para o cumprimento do que foi propôsto nos objetivos gerais e específicos deste trabalho.

3.3.1 Componente “Coleta”

Contém todas as ações, objetos, usuários participantes e aplicações definidas e desenvolvidas para a concepção do contexto de recebimento dos resíduos sólidos recicláveis. O principal ator deste componente é o “Coletor”, Sua principal funcionalidade é o recebimento de resíduos sólidos doados pelo “Cidadão”. A Figura 5 abaixo exhibe o diagrama de caso de uso do componente.

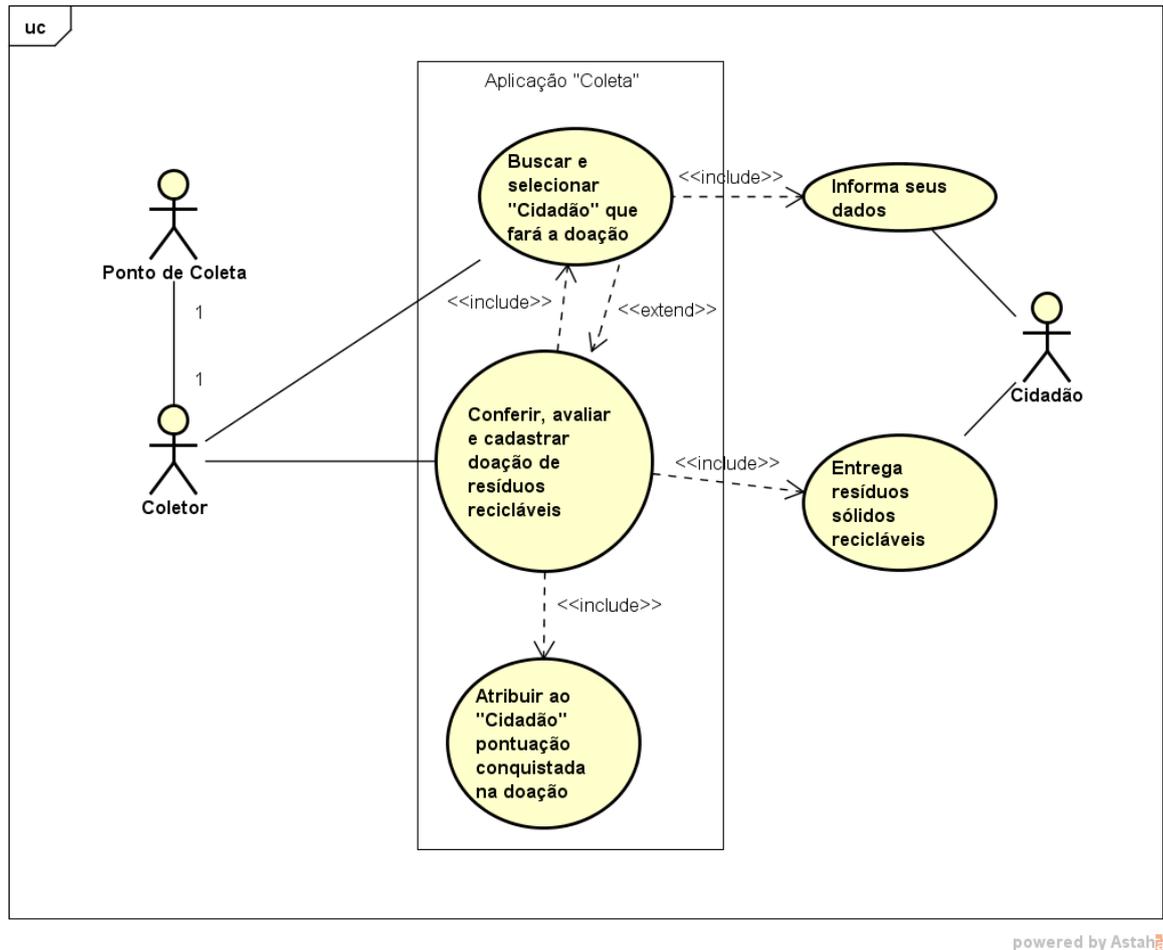


Figura 5 - Diagrama de caso de uso do componente “Coleta”. Fonte: o próprio via Astah

No diagrama acima temos a relação entre o “Coletor” e o “Ponto de Coleta”, indicando que um “Coletor” está associado a um “Ponto de Coleta” por vez. Dentro do retângulo observa-se as funcionalidades da aplicação “Coleta”, onde o ator “Cidadão” informa seus dados para que o “Coletor” os pesquise, encontre seu cadastro e selecione o “Cidadão” indicando que este fará uma doação de resíduos. Em seguida, o aplicativo libera a funcionalidade de cadastro de doação de resíduos, que permite ao “Coletor” cadastrar, contabilizar, avaliar e armazenar os resíduos doados pelo “Cidadão”. Os resíduos cadastrados serão associados ao “Ponto de Coleta” que o “Coletor” está associado. Após essa ação, o sistema automaticamente gera e armazena a pontuação adquirida pelo “Cidadão” que doou os resíduos.

3.3.1.1 Componente “Cadastro de Resíduos”

Esta funcionalidade foi desenvolvida parte na aplicação cliente “Coleta” na plataforma móvel, e no servidor web. A aplicação cliente foi feita em Ionic e AngularJS e o servidor na

plataforma Node.js com Express e seus componentes auxiliares. Na figura abaixo (figura 4) imagens da tela de cadastro de doações.

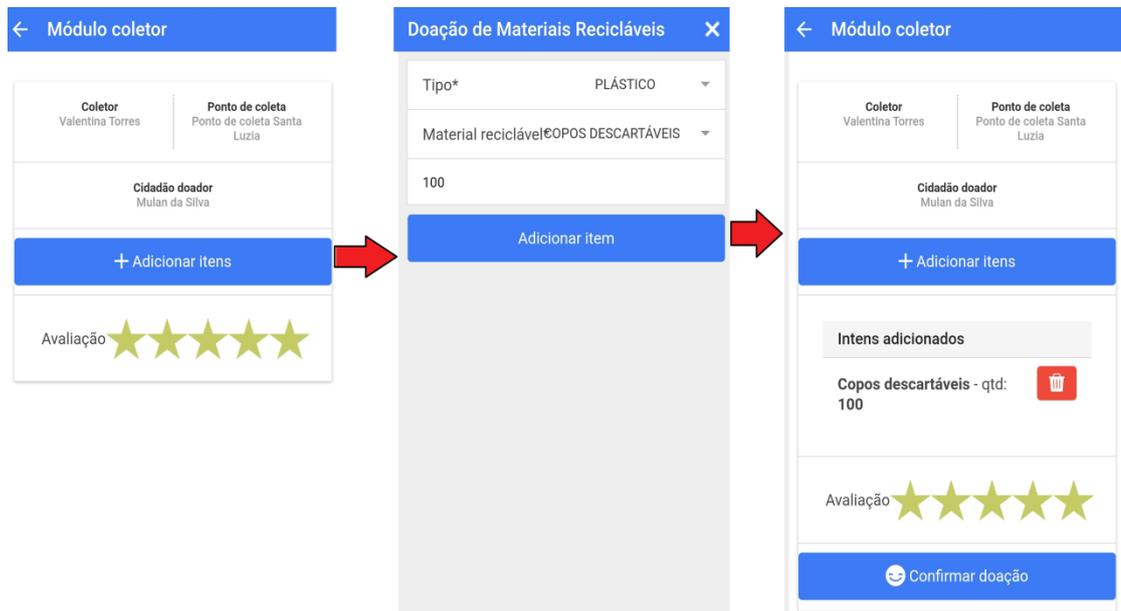


Figura 6 - Telas da aplicação "Coleta". Fonte: o próprio

Na Figura 6, da esquerda para a direita a tela inicial da funcionalidade cadastro de doação de resíduos, mostrando o “Coletor” logado no sistema, o “Ponto de Coleta” que ele está associado e o “Cidadão” selecionado para fazer a doação. Ao clicar no botão “Adicionar itens” a segunda tela é exibida para que o “Coletor” possa selecionar o tipo, material reciclável e a quantidade doada deste material. Ao finalizar a adição de itens a segunda tela fecha e o item inserido aparece na tela inicial. É permitido que seja adicionado outros itens e a avaliação exibida deve ser feita referente a doação de forma geral e não a cada item individualmente.

A tela descrita acima é implementada em HTML, sendo encapsulada por duas tags do Ionic: <ionic-view> e <ionic-content>. Essas tags indicarão ao Cordova que o código HTML dentro delas será convertido para a plataforma selecionada (Android ou IOS). As funcionalidades que tornam a tela dinâmica são implementadas em Javascript com AngularJS.

Por ser um *framework* baseado no modelo MVC o arquivo HTML que contém o código da tela representa o objeto de Visão, o arquivo Javascript com as funcionalidades representa o Modelo e o Controlador de forma que o objeto \$scope, disponibilizado pela diretiva *controller* do AngularJS, realiza automaticamente a ligação entre os dados da tela HTML e do arquivo em Javascript.

```

1 <ion-view title="Módulo coletor" id="page2">
2   <ion-content padding="true" class="has-header">
3
4     <div class="card">
5       <div class="item item-text-wrap" style="padding: 8px;">
6         <div class="row">
7           <div class="col col-50" style="text-align: center">
8             <h4 style="color: #999;">
9               <b style="color: #333;">Coletor</b><br>{{coletorSelecioneado.pessoa.nome}}
10            </h4>
11          </div>
12
13          <div class="col col-50" style="text-align: center; border-left: 1px dotted #999">
14            <h4 style="color: #999;">
15              <b style="color: #333;">Ponto de coleta</b><br>{{coletorSelecioneado.ponto[0].nome}}
16            </h4>
17          </div>
18        </div>
19      </div>
20    </div>
21
22    <div class="item item-text-wrap" style="padding: 8px;">
23      <div class="row">
24        <div class="col" style="text-align: center">
25          <h4 style="color: #999;">
26            <b style="color: #333;">Cidadão doador</b><br>{{cidadao.nome}}
27          </h4>
28        </div>
29      </div>
30    </div>
31  </ion-content>
32 </ion-view>

```

Figura 7 - HTML de parte da tela de cadastro de resíduos. Fonte: o autor

```

27   $scope.preencheDoacao = function(){
28     $scope.doacao.cidadao = $rootScope.cidadao._id;
29     $scope.doacao.coletor = $scope.coletorSelecioneado.usuario;
30     $scope.doacao.ponto = $scope.coletorSelecioneado.ponto[0]._id;
31     $scope.doacao.residuo_qtd = $scope.listaResiduoQtDs.map(function(obj) {
32       let objItemReciclavel = {
33         item_reciclavel: obj.material._id,
34         quantidade: obj.quantidade
35       }
36
37       return objItemReciclavel;
38     });
39   }
40

```

Figura 8 - Parte do código controller em Javascript/ AngularJS. Fonte: própria

Na Figura 7, nas linhas 1 e 2 são exibidas as tags Ionic explicadas acima, no final da linha 9, ao lado da tag `
`, há o acesso ao nome do coletor selecionado: `{{coletorSelecioneado.pessoa.nome}}`. Essa estrutura entre chaves são disponibilizadas pelo AngularJS e chamados templates de arquivo HTML. Estes templates acessam os dados atribuídos ao objeto `$scope` do arquivo Javascript (*controller*). Já na Figura 8 das linhas 27 à 39 é exibida a criação e implementação de uma função, chamada “preencheDoacao”, dentro do arquivo de *controller*. Na linha 29 e 30 o objeto “`$scope.coletorSelecioneado`” passa dados de seus atributos a outros objetos do *controller*. O objeto “coletorSelecioneado” exibido na Figura 8 compartilha a mesma referência do objeto dentro do template de chaves exibido na

Figura 7, dessa forma é exemplificado como o AngularJS disponibiliza os dados do *controller* para a página HTML.

Quando o “Coletor” finaliza o cadastro de doação de resíduos clicando no botão “Confirmar doação”, na terceira tela da Figura 6, é disparado uma solicitação para o servidor referente a função que implementa a ação, passando como parâmetro os dados da “Doação”, do “Cidadão”, do “Ponto de Coleta” e do “Coletor” nele associado.

```

78  api.adiciona = async function(req, res){
79      var doacao = req.body;
80      var residuoqtd = JSON.parse(JSON.stringify(doacao.residuo_qtd));
81      doacao.residuo_qtd = [];
82      var pontuacao = 0;
83      var pessoa = {pontuacao: 0, moedas: 0};
84
85      // Adiciona lista de resíduos
86      if(residuoqtd){
87          for(i in residuoqtd){
88              // Carrega dados do item para fazer o cálculo de pontuação
89              var item_reciclavel = null;
90              try {
91                  var item = await model_item_reciclavel.findById(residuoqtd[i].item_reciclavel)
92                      .populate('tipo')
93                      .populate('medida')
94                  if(!item) item_reciclavel = null;
95                  item_reciclavel = item;
96                  pontuacao += item_reciclavel.taxa_pontuacao * residuoqtd[i].quantidade;
97                  residuoqtd[i]._id = new mongoose.Types.ObjectId();
98
99                  var residuoqtd_retorno = await model_residuo_qtd.create(residuoqtd[i])
100                  console.log('Resíduo adicionado: '+ residuoqtd_retorno);
101                  doacao.residuo_qtd.push(residuoqtd_retorno);
102
103              } catch (error) {
104                  item_reciclavel = null;
105                  console.log(error);
106                  res.status(500).json(error);
107              }
108          }
109      }

```

Figura 9 - Função cadastro de doação: parte 1 calcular pontuação. Fonte: o próprio

Na linha 78 da Figura 9 é criado dinamicamente a função assíncrona “adiciona” dentro do objeto “api”. Em seguida é armazenado na variável “doacao” o conteúdo de “req.body” que é o objeto corpo da requisição feita pela aplicação cliente, e neste caso possui os dados do Ponto de Coleta, Cidadão, Coletor, e os dados dos itens doados. Na linha 80 é criada a variável *residuoqtd* que recebe como valor um objeto JSON com os dados vindos em “doacao.residuo_qtd”. Na linha 82 e 83 são criadas e iniciadas as variáveis *pontuacao* e “pessoa” respectivamente.

Apartir da linha 86 começa a parte da funcionalidade que faz a contabilização dos pontos acumulados com a doação feita. Na linha 87 existe um laço “for” que percorrerá o vetor `residuoqtd` sendo “i” o índice atual do laço de repetição. Na linha 91, é feito uma chamada `findById` no objeto “`model_item_reciclavel`”, isto é uma busca a um único dado de “`itens_reciclavel`” na base de dados `Mongoose`, passando o ID do item desejado, este ID é acessado utilizando o índice (“i”) para acessar o objeto corrente (“`residuoqtd[i].item_reciclavel`”). De posse do item reciclável, na linha 96 é calculado a pontuação do primeiro item utilizando a taxa de pontuação do item multiplicada pela quantidade doada, a variável “`pontuacao`” acumulará as pontuações dos itens recicláveis de acordo com a execução do laço `for`. Nas linhas 99 a 101 são armazenados na estrutura `MongoDB` `residuos_qtd` os dados do item doado e na sequência o lançamento e exceções caso houver.

```

113 | // Calcula pontuação da doação
114 | if (doacao.avaliacao == 5){
115 |     doacao.pontuacao = pontuacao * 1.2;
116 | }
117 | else if (doacao.avaliacao == 4){
118 |     doacao.pontuacao = pontuacao * 1.1;
119 | }
120 | else if(doacao.avaliacao == 3){
121 |     doacao.pontuacao = pontuacao;
122 | }
123 | else if(doacao.avaliacao == 2){
124 |     doacao.pontuacao = pontuacao * 0.8;
125 | }
126 | if(doacao.avaliacao == 1){
127 |     doacao.pontuacao = pontuacao * 0.5;
128 | }
129 |
130 | doacao.data = new Date();

```

Figura 10 - Função cadastro de doacao, parte 2 ajustando a pontuação conforme a avaliação. Fonte própria

A Figura 10 exibe a continuação do código “adiciona” mostrado na Figura 9. Após o laço `for`, da linha 114 a linha 128 é verificada a avaliação atribuída pelo “Coletor” a doação cadastrada, se a doação for bem avaliada a pontuação pode ser mantida ou acrescida de 10% a 20% no fator da pontuação. Caso seja mal avaliada o “Cidadão” pode perder de 20% a 50% do valor de pontuação da doação. Uma doação é bem avaliada quando o “Coletor” nota que as condições de armazenamento e segregação dos materiais doados foram bem realizadas, caso

contrário, além da avaliação recomenda-se que o “Coletor” ensine ao “Cidadão” qual a melhor forma de acondicionar o referido material. E por fim na linha 130 é criada e armazenada a data da transação de coleta seletiva dentro do objeto “doacao”.

```

132 | // Inserir de pontuação do cidadão
133 |
134 | // Obter objeto pessoa em cidadao
135 | try {
136 |     retorno_pessoa = await model_pessoa.findById(doacao.cidadao).populate('pessoa');
137 |     if(!retorno_pessoa) throw Error('Cidadao não encontrado');
138 |     if(!retorno_pessoa.pontuacao) retorno_pessoa.pontuacao = 0;
139 |     if(!retorno_pessoa.moedas) retorno_pessoa.moedas = 0;
140 |     pessoa.pontuacao = retorno_pessoa.pontuacao;
141 |     pessoa.moedas = retorno_pessoa.moedas;
142 | } catch (error) {
143 |     console.log(error);
144 |     res.status(404).json(error);
145 | }
146 | // Fim obter objeto pessoa
147 |
148 | // Atualizar pontuação
149 | setTimeout( async ()=>{
150 |     pessoa.pontuacao += doacao.pontuacao;
151 |     pessoa.moedas += doacao.pontuacao;
152 |     try {
153 |         await model_pessoa.findByIdAndUpdate(doacao.cidadao, pessoa);
154 |         console.log('Pontuação alterada com sucesso!');
155 |     } catch (error) {
156 |         console.log(error);
157 |         res.status(500).json(error);
158 |     }
159 |
160 | }, 150);
161 | // Fim atualizar pontuação

```

Figura 11 - Função cadastro de doação. Parte 3 atualizando pontuação do "Cidadão". Fonte: o próprio

Após o cálculo da pontuação, dando sequência ao método “adiciona”, na Figura 11 da linha 136 a 141 é feito a busca no MondoDB pelo objeto que representa o “Cidadão”, esses dados são armazenados na variável “retorno_pessoa”, caso não haja dados referentes a pontuação e moedas esses atributos serão criados dinamicamente e iniciados com o valor zero, caso contrário é mantido o valor resgatado e nas linhas 140 e 141 esses valores são adicionados nos campos pontuacao e moeda do objeto pessoa (pessoa.pontuacao, pessoa.moeda). Nas linhas 150 e 151 a pontuação atual, que foi calculada, é somada a pontuação que o “Cidadão” já obtinha, na linha 153 através do método “findByIdAndUpdate” disponibilizado pelo objeto Mongoose model_pessoa a pontuação é alterada no banco de dados, sendo o restante do código logs e tratamentos de exceções.

```

163 | // Cadastra doação
164 | setTimeout( async ()=> {
165 |     try {
166 |         doacao_retorno = await model.create(doacao)
167 |         console.log('Doação cadastrada!');
168 |         res.json(doacao_retorno);
169 |     } catch (error) {
170 |         console.log(error);
171 |         res.status(500).json(error);
172 |     }
173 | }, 200);
174 | // Fim do cadastro de doação
175 |
176 | }, 300);
177 |
178 | }

```

Figura 12 - Função cadastro de doação, parte 5: armazenando doação na base de dados. Fonte: o autor

Por fim, após o cálculo e atualização da pontuação do usuário, na linha 166 da Figura 12 exibe o cadastro do objeto doação na base de dados MongoDB via função create, e na linha 168 é feito o retorno para a aplicação cliente que solicitou a função, retornando o código 200 de sucesso (*success*) e o JSON do objeto “doacao” após salvo na base. Na Figura 13 abaixo, é exibido um exemplo de como fica um objeto “doacao” resultante da função “cadastro de doação” e armazenado na base do MondoDB.

```

  _id: ObjectId("5ea64e55feb9b700173783dc")
  ↓ residuo_qtd: Array
    0: ObjectId("5ea64e54feb9b700173783d6")
    1: ObjectId("5ea64e55feb9b700173783d8")
    2: ObjectId("5ea64e55feb9b700173783da")
  cidadao: ObjectId("5e9ce136083bfd00178cc4ab")
  ponto: ObjectId("5ae9259e6f8032d69c46c0f2")
  coletor: ObjectId("5aecbf624fa3c404290354f8")
  avaliacao: 5
  pontuacao: 900
  data: 2020-04-27T03:15:33.356+00:00
  __v: 0

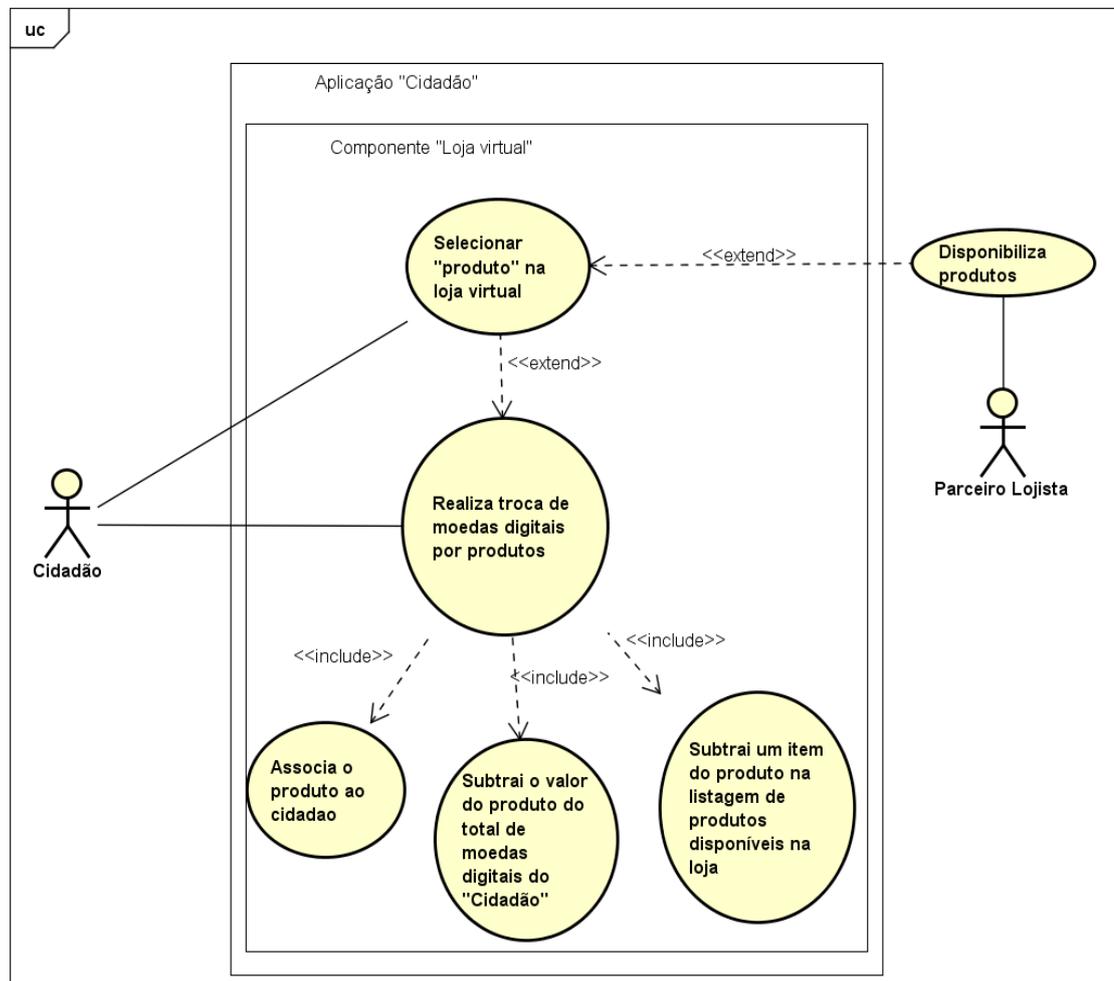
```

Figura 13 - Exemplo de objeto doação de resíduos salvo na base de dados do MongoDB. Fonte: o próprio

3.3.2 Componente “Loja virtual”

Consiste na parte do sistema que exibe os produtos disponibilizados pelos parceiros para que o “Cidadão” possa escolher um e trocá-lo por suas “Moedas digitais”. Possui como

principal ator é o perfil “Cidadão”. A Figura 14 abaixo contém o diagrama de caso de uso referente a esta funcionalidade.



powered by Astah

Figura 14- Caso de uso do componente “Loja virtual”. Fonte: o próprio

No diagrama da Figura 14 o componente “Loja virtual” está localizado dentro da “Aplicação cidadão” simbolizando que loja é parte dessa aplicação que também possui outros componentes como o perfil de usuário e a listagem de produtos trocados disponíveis para resgate que nesse momento foram omitidos. A principal funcionalidade é a troca de produtos por “Moedas Digitais”, que dispara três ações no servidor, como a associação do produto trocado à lista de produtos para resgate do “Cidadão”, a atualização da quantidade de “Moedas digitais” desse usuário, e a subtração da quantidade do produto trocado na listagem de produtos disponíveis para a troca. A Figura 15 abaixo possui imagens referentes a listagem de produtos disponíveis para troca por moedas.

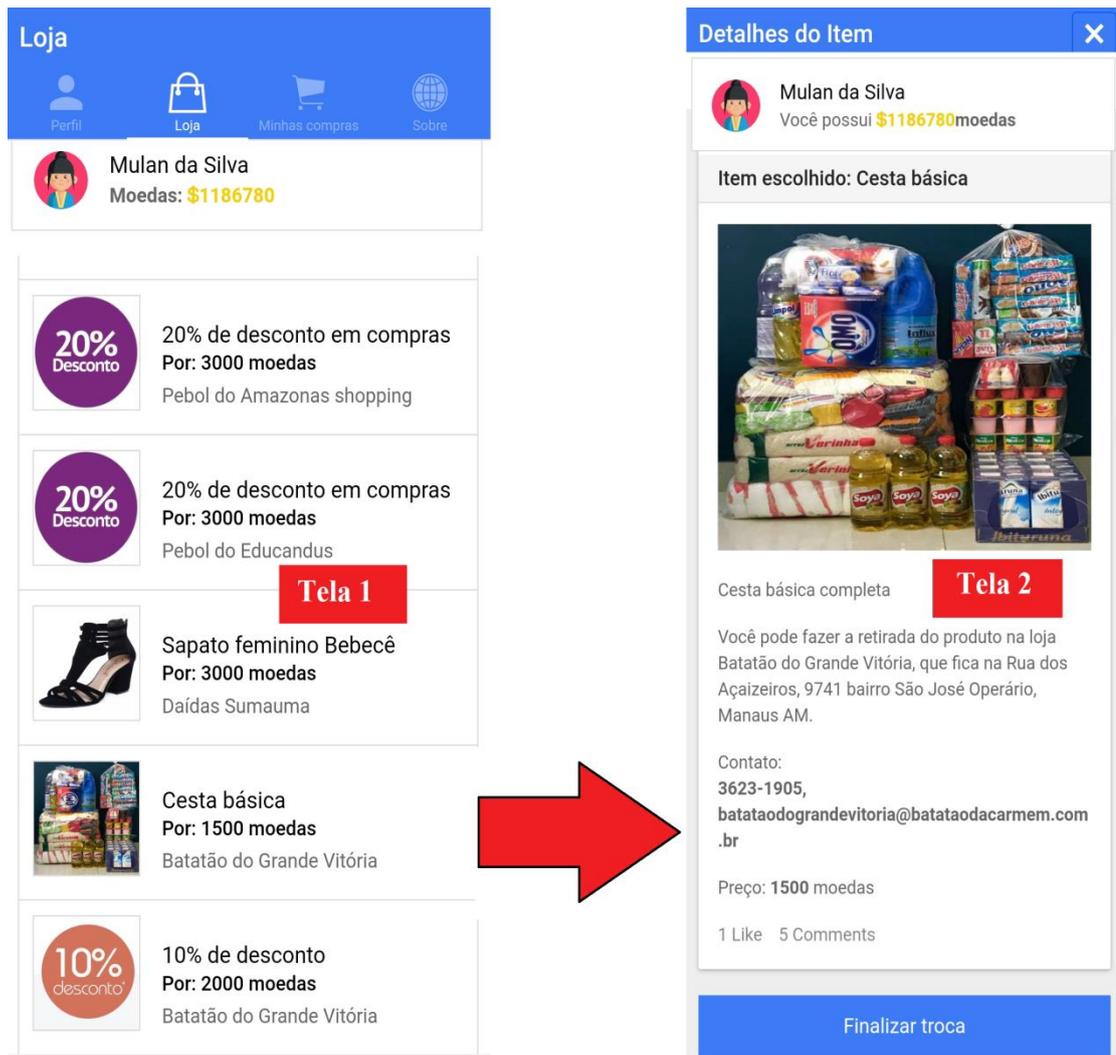


Figura 15 - Telas do componente “Loja virtual”. Fonte: o autor

Na Figura 15 a “tela 1” a esquerda exibe a listagem de produtos disponíveis para troca, sua imagem, descrição, preço em “Moedas digitais” e nome da loja que o forneceu. Clicando em cima de um dos produtos a tela “Detalhes do item” aparece, conforme a “tela 2” na figura acima que representa a seleção do produto cesta básica. Quando o usuário clicar no botão “finalizar troca” na “tela 2” será disparado o evento para o servidor responsável por realizar a ação de troca.

3.3.2.1 Funcionalidade troca de produtos por moedas digitais

```

202  api.realizarTroca = async function(req, res){
203
204    try {
205      var dados_pessoa = await pessoa.findById(req.body.pessoa);
206      var item_selecionado = await item_model.findById(req.body.item).populate('produto');
207      var moedas_restante = dados_pessoa.moedas - item_selecionado.produto.preco;
208      if (moedas_restante < 0){
209        error = new Error('Saldo insuficiente');
210        return res.status(500).json({erro: error, mensagem: 'Saldo insuficiente'});
211      }
212      else {
213        dados_pessoa.moedas = moedas_restante;
214        item_selecionado.quantidade = item_selecionado.quantidade-1;
215        if(item_selecionado.quantidade < 0){
216          return res.status(500).json({erro: '500', mensagem: 'Item selecionado esgotou no estoque'});
217        }
218        else {
219          /** Atualiza quantidade de moedas da pessoa */
220          await pessoa.findByIdAndUpdate(dados_pessoa._id, dados_pessoa);
221          console.log('Moedas descontadas com sucesso.');
```

Figura 16 – Primeira parte do código da função troca de produtos por moedas digitais . Fonte: o autor

A primeira parte do código de troca é exibido na Figura 16, onde na linha 202 a função “realizarTroca” é criada com os parâmetros de requisição (req) e de resposta (res), e em seguida é atribuída ao objeto “api”. Na linha 205 são utilizados os dados do “Cidadão” que solicitou a troca e o método “*findById*” do Mongoose para resgatar no banco os dados que refletem o usuário no contexto da aplicação, sendo esses atribuídos à variável “dados_pessoa”. Na linha 206 é feita a mesma coisa da linha anterior, porém é resgatado os dados do item e do produto que está associado ao item, este último via o comando “*populate*” do Mongoose, estes dados são armazenados na variável “item_selecionado”.

Mesmo sendo feita a verificação se o “Cidadão” solicitante possui moedas suficientes para trocar por aquele produto, o processo é repetido no lado do servidor pois outras aplicações poderão consumir seus serviços. Conforme a Figura 16, essa verificação é feita na linha 207 e 208, e em seguida, nas linhas 210 e 211 é configurado e lançada uma mensagem de erro para a aplicação cliente solicitante. Havendo a quantidade de moedas necessárias, é verificado se o produto solicitado ainda tem em estoque uma vez que o produto pode ter sido alvo de outra solicitação ao mesmo tempo por outro “Cidadão” solicitante sendo lançada a mensagem de erro conforme a linha 216 da Figura 16. Após a verificação dessas conformidades, caso nenhum erro tenha sido lançado, através do método “*findByIdAndUpdate*” são subtraídas a quantidade de moedas do cidadão e a quantidade do produto em estoque respectivamente conforme as linhas 220 e 223 da Figura 16.

```

225
226 // Gerando troca
227 var troca = {};
228 troca.codigo = Date.now(); // gerando código
229 troca.status = true;
230 troca.data = new Date();
231 troca.pessoa = new mongoose.Types.ObjectId(dados_pessoa._id);
232 troca.item = new mongoose.Types.ObjectId(item_selecionado._id);
233 /** Insere troca */
234 const dados_troca = await troca_model.create(troca);
235 console.log('Transação realizada com sucesso!');
236 retorno = {
237   pessoa: dados_pessoa,
238   item : item_selecionado,
239   troca : dados_troca
240 }
241 console.log('\n');
242 return res.json(retorno);
243 // console.log('Erro na transação! \n Executando rollback');
244 }
245

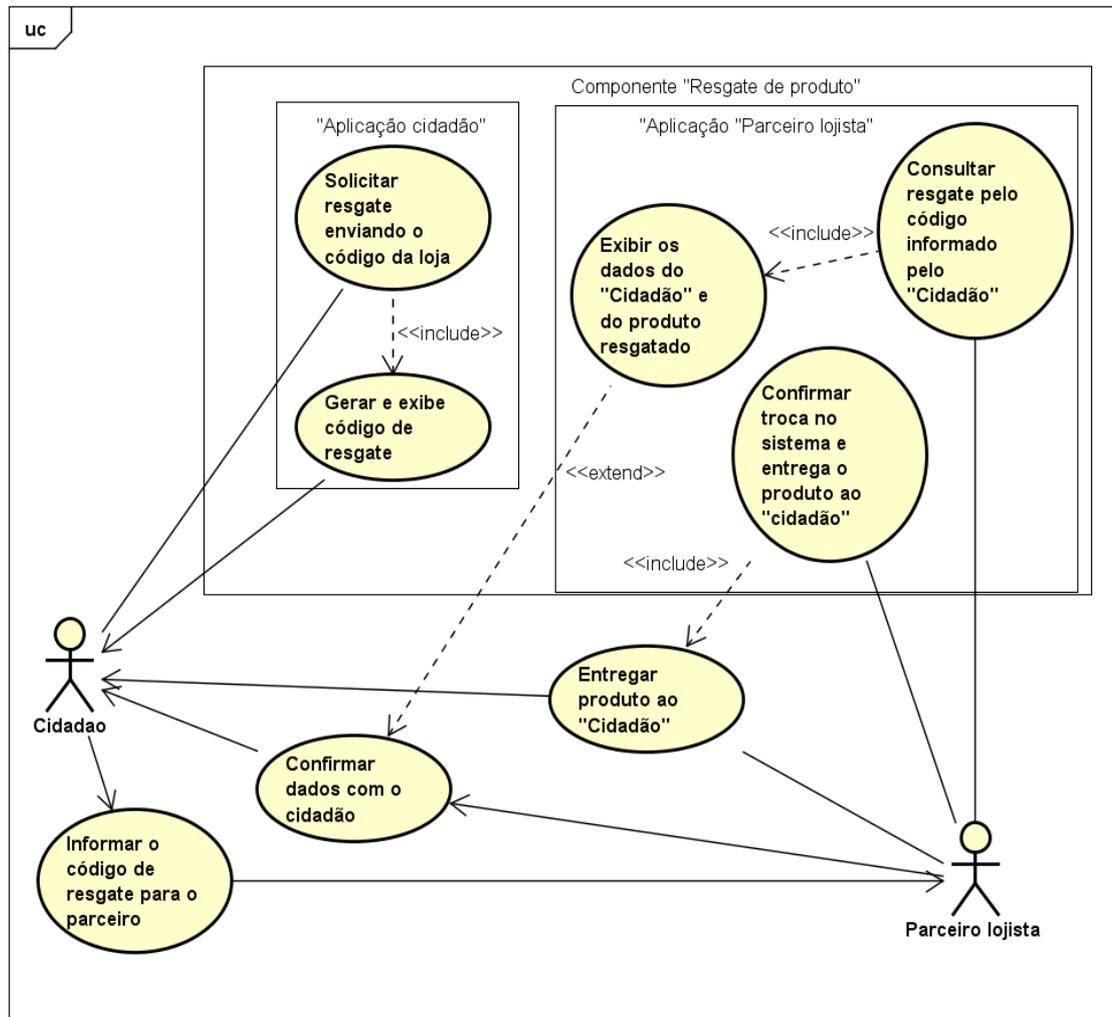
```

Figura 17 - Parte 2 do código de troca de produtos por moedas. Fonte: o autor

Dando sequência ao código de troca a Figura 17 mostra, no intervalo entre as linhas 227 e 232, a criação do objeto JSON “troca” e declaração dos seus atributos, sendo que nas linhas 231 e 232 são atribuídos os valores do identificador do usuário solicitante e do item selecionado aos atributos “troca.pessoa” e “troca.item”. Na linha 234 o método “create” do Mongoose aciona o comando de criação de objeto na base de dados do MongoDB, na sequência é impresso um mensagem de sucesso no servidor e é configurado um objeto de retorno nas linhas 236, 237, 238 e 239 contendo a pessoa, o item e o objeto trocado. E por fim, utilizando o atributo “res” carregado pelo express para a função, é dado a aplicação cliente um retorno com o código 200 referente a sucesso e o objeto de retorno configurado.

3.3.3 Componente “Resgate de produto”

Este componente possui dois atores principais, sendo estes o “Cidadão” e o “Parceiro Lojista”. A principal funcionalidade deste componente é a função de resgate de produto, que para ser realizada os dois atores citados precisarão interagir via sistema ao mesmo tempo.



powered by Astah

Figura 18 - Caso de uso do componente resgate de produto trocado. Fonte: o autor

Os retângulos exibidos na Figura 18 representam, de cima para baixo da esquerda para direita, o componente “Resgate de Produtos” com suas funções distribuídas entre a “Aplicação cidadão” e a “Aplicação parceiro lojista”. O resgate é solicitado pelo “Cidadão” através da função “Solicitar resgate enviando o código da loja”, esta função retorna o “Código de resgate” do produto para o “Cidadão”, este comunica o “Código de resgate” ao “Parceiro Lojista” que insere o código no sistema via função “Consultar resgate pelo código”, essa última lhe retorna os dados do “Cidadão” e do “Produto” para que o “Parceiro lojista” confirme tais dados junto ao usuário solicitante. Após a confirmação o “Parceiro lojista” confirma a troca no sistema e entrega o produto para o “Cidadão”.

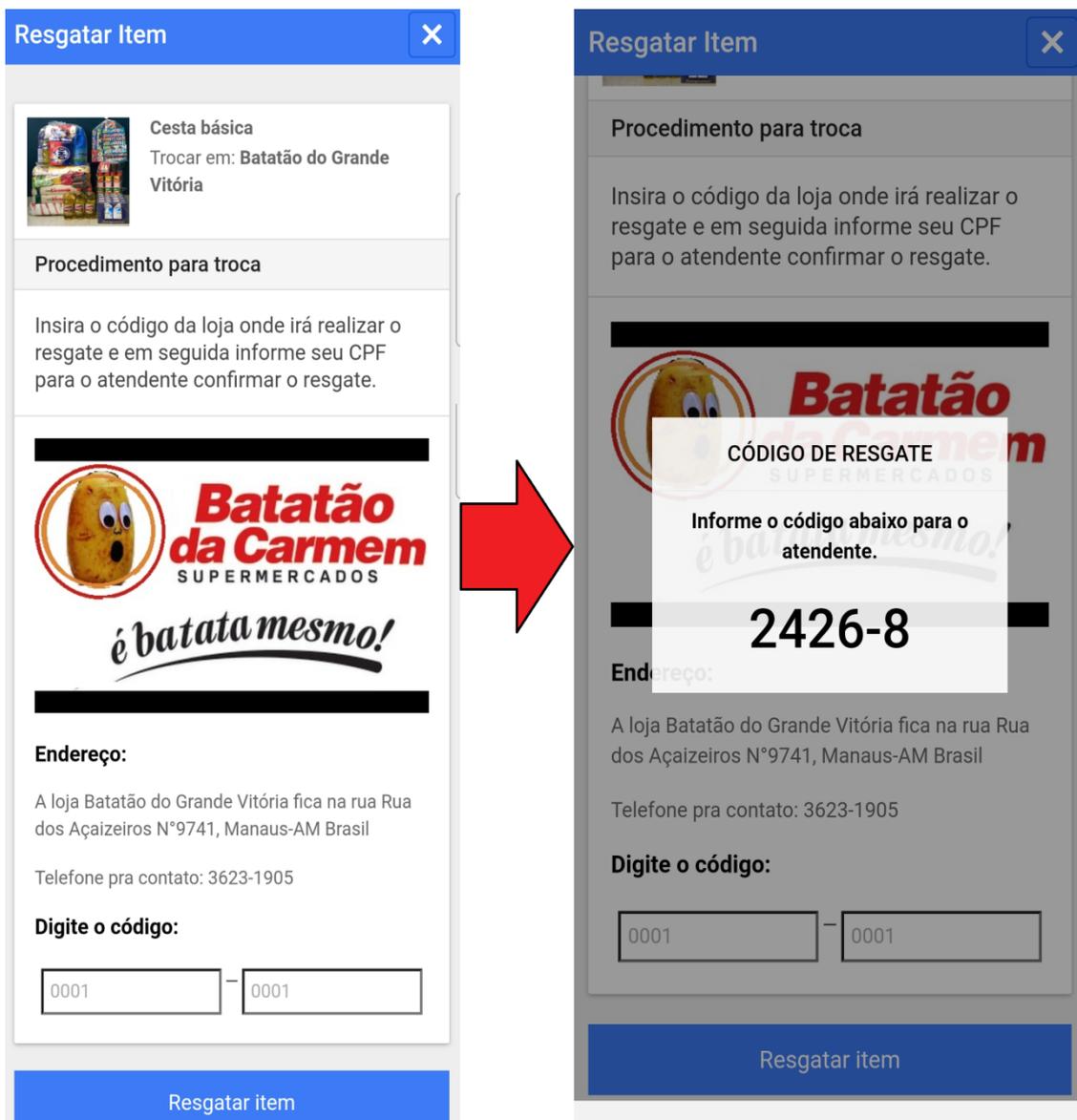


Figura 19 - Tela da "Aplicação cidadão" referente ao componente de "Resgate de produto". Fonte: do autor

Na esquerda da Figura 19 é exibida a tela que detalha os dados do item selecionado para resgate. O botão “Resgatar item” dispara a função “Solicitar resgate enviando o código da loja“ exibida no caso de uso da Figura 18 e exibe o “Código de resgate” para que o usuário o informe ao atendente (“Parceiro lojista”).

The screenshot shows a web interface for a store redemption application. At the top, there is a green header with 'Foto' and 'Compo' links. The main title is 'Resgate de itens na loja'. Below the title, there are three dropdown menus for selection: 'Selezione a empresa:' with 'Rufalo venda de comidas Ltda', 'Selezione uma loja da empresa:' with 'Batatão do Grande Vitória', and 'Selezione o parceiro que fará o atendimento:' with 'Parceiro 1'. In the center, there is a placeholder image of a character (Naruto) and a text prompt 'Informe o código de resgate do cidadão'. At the bottom, there is a green button labeled 'CONSULTAR DADOS'.

Figura 20 - Tela da "Aplicação lojista" referente ao componente de resgate de produtos. Fonte: o autor

The screenshot shows a modal dialog box titled 'Confirmar dados' overlaid on the previous screen. The dialog contains the following information: 'Nome do solicitante: Mulan da Silva' and 'Item solicitado : Cesta básica completa'. Below the text is a photograph of a grocery basket filled with various food items. At the bottom of the dialog, there are two buttons: a green 'CONFIRMAR RESGATE' button and a 'CANCELAR' button. The background screen is dimmed, showing the same selection dropdowns and the 'CONSULTAR DADOS' button.

Figura 21 - Tela da "Aplicação lojista" referente ao componente de resgate de produtos. Fonte: o autor

Na Figura 20 observa-se a tela de resgate de produtos da “Aplicação lojista” e na Figura 21 os dados referente ao resgate para confirmação com o “Cidadão”. Quando o usuário “Parceiro lojista” clica no botão “Confirmar resgate”, na Figura 21, o resgate é finalizado via sistema e o parceiro pode entregar o produto ao “Cidadão”. Nas figuras abaixo temos as

mensagens de sucesso exibidas para a “Aplicação Cidadão” na Figura 22 e na “Aplicação parceiro lojista” na Figura 23.

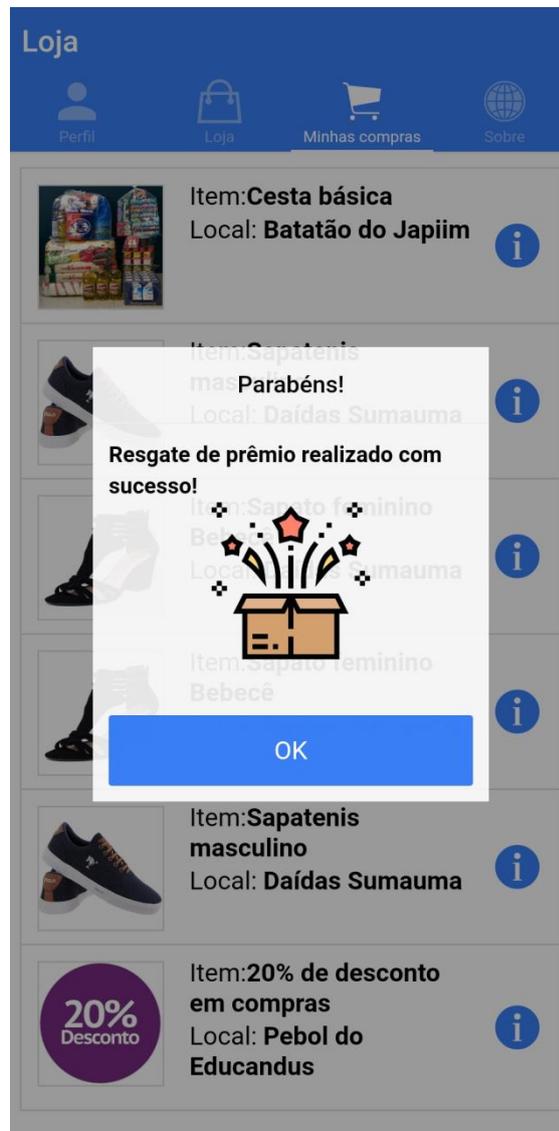


Figura 22 - "Aplicação Cidadão" mensagem de resgate concluído com sucesso. Fonte: o próprio

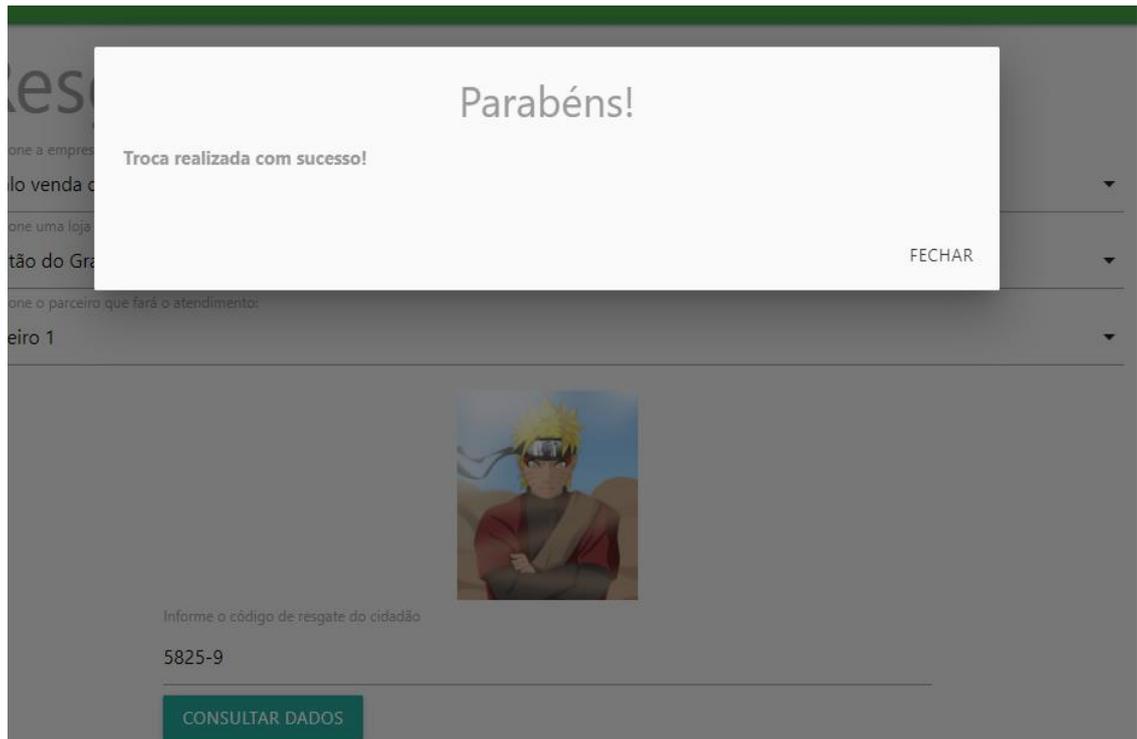


Figura 23 - "Aplicação parceiro lojista" mensagem de resgate concluído com sucesso. Fonte: o próprio

Os subtópicos a seguir, detalham a implementação das principais funcionalidades do componente “Resgate de produto”.

3.3.3.1 Funcionalidade gerar intenção de resgate

A intenção de resgate é gerada pelo usuário “Cidadão” quando o mesmo vai até a loja parceira resgatar o produto que o parceiro disponibilizou. Essa recorte da funcionalidade é exibida no diagrama de caso de uso da figura 18, e esse foi isolado no diagrama de caso de uso contido na figura 24 abaixo.

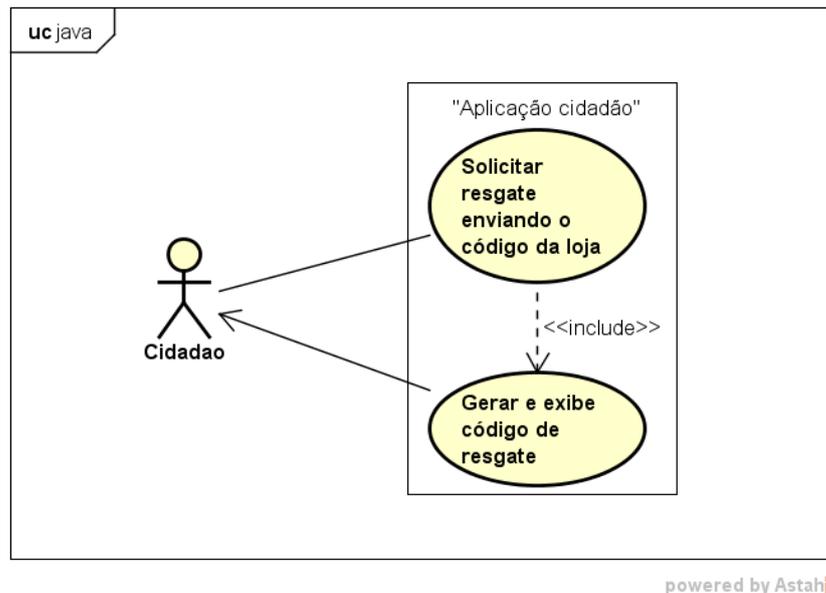


Figura 24 – Caso de uso “Criar intenção de resgate”. Fonte: o próprio

Na figura 24 o ator “Cidadao” representa nosso usuário que, com o seu aplicativo, sinaliza que deseja resgatar um de seus itens da sua lista de compras. O aplicativo envia o sinal de resgate para o serviço no *webservice* que processa essa solicitação e retorna um código único, que chamamos de “Código de troca”. Esse código será informado pelo “Cidadão” ao atendente no balcão de troca da loja e assim o processo de resgate segue para a próxima etapa, que será detalhada no próximo tópico deste capítulo. A figura 25 abaixo exibe código que implementa essa parte da funcionalidade seguido da sua explicação linha a linha.

```

57  api.adiciona = async function(req, res) {
58
59      var intencaoObjReq = req.body;
60      try {
61          var loja = await lojaModel.findOne({cod: intencaoObjReq.codigo.split("-")[0]});
62          if(loja == null || loja._id == null) throw "Loja inválida ou não enviada.";
63          var troca = await trocaModel.findOne({_id: intencaoObjReq.troca, status: true}).populate('item');
64          if(troca == null || troca._id==null) throw "Troca não encontrada.";
65
66          if( !(loja._id.equals(troca.item.loja)))
67              throw "Código da loja não corresponde a loja do item selecionado. Por favor tente novamente.";
68
69          var parceiro = await parceiroModel.findOne({cod: intencaoObjReq.codigo.split("-")[1]});
70          if(parceiro == null || parceiro._id==null) throw "Parceiro inválido ou não enviado."
71
72          if( !loja._id.equals(parceiro.loja)){
73              throw "Código da loja não compatível. Por favor repita o processo.";
74          }
75
76          var itensCadastradosPeloUsuario =
77              await intencaoDeresgateModel.find({
78                  pessoa: intencaoObjReq.pessoa,
79                  loja,
80                  parceiro,
81                  trocado: false
82              });
83
84

```

Figura 25 - Funcionalidade gerar intenção de resgate parte 1. Fonte: o autor

Conforme os códigos exibidos anteriormente, o código exibido na linha 57 da Figura 25 mostra a declaração da função, na linha 59 o dados contidos no *body* da requisição são armazenados na variável “intencaoObjreq”. Os dados do corpo da requisição que essa função espera receber da aplicação cliente são o código da loja e os identificadores de troca do “Cidadão”.

O código da loja é dividido em dois sendo a primeira parte dele referente a loja e a segunda ao “Parceiro lojista” dessa loja. Ainda na figura 25 na linha 61, a primeira parte do código é utilizada para consultar os dados da loja no banco de dados, sendo esses validados na linha 62. O procedimento é repetido na linha 63 dessa vez consultando os dados da troca e do produto associado a ela, sendo esse dado validado na linha 64. Nas linhas 66 e 67 é verificado se o item a ser resgatado está cadastrado na loja. Nas linhas 69 e 70 a segunda parte do código enviado é utilizada para recuperar os dados do parceiro atendente e feita a validação. Nas linhas 72, 73 e 74 é feita a confirmação que aquele parceiro está vinculado a essa loja.

Prosseguindo, nas linhas 76 até a 82 da Figura 25, é feito uma busca por todas as intenções de resgate criadas pelo “Cidadão” solicitante, para aquela loja, àquele parceiro e que não tenham sido trocados, isto é, todas as intenções criadas de forma redundante.

```

84      /*** Caso o usuário marque mais de um item para a troca no mesmo estabelecimento
85      * todos os itens serão deletados e apenas o mais recente ficará.
86      *
87      * Caso o campo trocado da intenção esteja true o item já foi resgatado e não
88      * deve ser deletado para fins de auditoria (Subir o código e testar amanhã 28/07/21)
89      */
90      if (itensCadastradosPeloUsuario.length > 0) {
91          await intencaoDeresgateModel.deleteMany({
92              _id: itensCadastradosPeloUsuario.map(item => {
93                  return item._id;
94              })
95          });
96      }
97

```

Figura 26 - Funcionalidade gerar intenção de resgate parte 2. Fonte: o autor

Ao usuário “Cidadão”, não é impedido criar “intenções de resgate” quantas vezes quiser referente ao mesmo produto trocado. Porém, no código da Figura 26 na linha 90 a 96 são percorridos e deletados todos os dados redundantes que podem ser gerados por essa ação. Esses dados são obtidos pela consulta feita nas linhas anteriores 76 a 82 da figura 25.

```

98     intencaoObjReq.trocado = false;
99     intencaoObjReq.poderesgatar = false;
100    intencaoObjReq.data = new Date();
101    intencaoObjReq.loja = loja;
102    intencaoObjReq.parceiro = parceiro;
103    intencaoObjReq.codintencao = geraCodigo();
104
105    var intencaoSalva = await intencaoDeresgateModel.create(intencaoObjReq);
106    res.json(intencaoSalva);
107
108    } catch (error) {
109        console.log('Erro: ', error);
110        res.status(500).json(error);
111    }
112
113 }
...

```

Figura 27 - Funcionalidade gerar intenção de resgate parte 3. Fonte: o autor

Nas linhas 98 à 103 da Figura 27 são atualizados os dados do objeto “intencaoObjReq”, onde na linha 100 a data da intenção é gerada e na linha 103 o “Código de troca” é gerado. Na linha 105 é feito o armazenamento da intenção no banco de dados e na linha 106 é dada a resposta ao solicitante já com o “Código de troca” para ser exibido na tela do “Cidadão”. O restante do código refere-se ao tratamento de possíveis exceções que possam ocorrer no sistema.

3.3.3.2 Funcionalidade consultar intenção de resgate

```

156    var dataIntencao = new Date(intencaoDeResgates.data);
157    var dataParceiro = new Date();
158
159    if(compararDatas(dataParceiro, dataIntencao)){
160        // Tudo certo para o resgate
161        console.log('Continua com o resgate do item.');
```

// Após consulta validada alterar flag para permitir o resgate

```

162        await intencaoDeresgateModel.findByIdAndUpdate(intencaoDeResgates._id, {poderesgatar: true});
163        var item = await itemModel.findById(intencaoDeResgates.troca.item).populate('produto');
164        var resposta = {
165            intencao : intencaoDeResgates,
166            item
167        }
168    }
169    res.json(resposta);
170 }
171 else{
172     // Faz tempo que a itenção foi criada deletar
173     throw "Tempo para troca expirou. Por favor, peça para o usuário repetir o processo";
174 }
```

Figura 28 - Funcionalidade consultar intenção de resgate. Fonte: o autor

O código exibido na Figura 28 é acionado quando o “Parceiro Lojista” consulta o “Código de resgate” informado pelo “Cidadão”. Com o código de resgate enviado pela aplicação cliente, o sistema carrega os dados da itenção e faz validações. Na linha 156 é

gerado um objeto *Date* da intenção criada pelo “Cidadão” e em seguida na linha 157 é criado uma data atual. Na linha 159 as datas são passadas para o método “compararDatas”, este verifica se a intenção de resgate criada pelo usuário já expirou, essa expira quando criada há mais de 5 minutos. Na linha 163 o atributo “poderesgatar” do objeto contendo os dados da intenção de resgate alvo da função é alterado para true, liberando essa intenção para o resgate. Nas linhas 165 à 169 a resposta de sucesso (200) é configurada e enviada para a aplicação cliente. O restante do código da Figura 28 refere-se ao tratamento de possíveis exceções que podem ocorrer quando a função é executada.

3.3.3.3 Funcionalidade confirmar resgate de produto

Etapa final para resgate de produto, feita via sistema após a criação e consulta da “Intenção de troca”.

```

321 | try {
322 |   var troca = await troca_model.findById(idTroca);
323 |   var intencao = await intencaoderesgate_model.findById(idIntencao);
324 |
325 |   if (!intencao.poderesgatar){
326 |     throw "Usuário não concedeu a permissão para o resgate desse produto! Relatando ao administrador...";
327 |   }
328 |
329 |   if (troca == null || intencao == null){
330 |     throw "Não foi possível encontrar a troca informada! Peça para o usuário repetir o processo";
331 |   }
332 |
333 |   if (!troca._id.equals(intencao.troca)){
334 |     throw "Item da troca divergente. Relatando ao administrador...";
335 |   }
336 |
337 |   var dataIntencao = new Date(intencao.data);
338 |
339 |   if(compararDatas(new Date(), dataIntencao)){
340 |     await troca_model.findByIdAndUpdate(idTroca, {status: false});
341 |     await intencaoderesgate_model.findByIdAndUpdate(idIntencao, {trocado: true});
342 |
343 |     troca = await troca_model.findById(idTroca);
344 |     intencao = await intencaoderesgate_model.findById(idIntencao);
345 |
346 |     res.json({troca, intencao});
347 |   }
348 |   else {
349 |     throw "Tempo para troca expirou. Por favor, peça para o usuário repetir o processo";
350 |   }
351 | }
352 |
353 | catch (error) {
354 |   console.log('Erro: ', error);
355 |   res.status(500).json(error);
356 | }
357 |
358 | }

```

Figura 2924 - Funcionalidade confirmação de resgate de produto

Na Figura 29 linhas 322 e 323, são carregados, de acordo com a informação trazida do corpo da requisição feita pelo cliente, dados de troca e intenção. Da linha 325 à 335 são

realizadas verificações básicas de conformidade entre os dados desses objetos. Nas linhas 337 e 339 é verificado se foi expirado o tempo de 5 minutos entre a criação da intenção e a confirmação final. Se tudo estiver nos conformes o status do objeto troca é alterado para *false*, conforme a linha 340, isto faz que o mesmo não apareça mais na listagem de produtos disponíveis para resgate. Em seguida, na linha 341 é atualizado o atributo “trocado” do objeto intenção para “*true*”, indicando que essa intenção foi concluída. Nas linhas 343 e 344 são carregados os dados atualizados e enviados no corpo da resposta de sucesso (*status 200*) para a aplicação cliente. As demais linhas contém lançamento e tratamento de exceções caso ocorram erros na execução da função.

Apresentadas as principais funcionalidades e detalhes do seu desenvolvimento, faz-se necessário considerar que a complexidade do ecossistema neste projetado e desenvolvido é muito maior do que as apresentadas. Vale ressaltar este trabalho baseou-se em linguagens e bibliotecas de código livre. Questões como escalabilidade e velocidade de resposta não foram abordadas para que a demonstração da ideia principal pudesse ser transmitida de modo claro e preciso.

Capítulo 4 – Considerações finais

Este trabalho foi concebido utilizando conceitos da Tecnologia, Análise e Desenvolvimento de Software para interferir positivamente na questão ambiental relativa a coleta seletiva de resíduos sólidos. Ao longo da concepção deste foram aplicadas pesquisas referentes ao tema ambiental para entendimento do cenário. A partir disso técnicas e conceitos de organização e métodos, análise de sistemas, designer de interação entre humanos e sistemas, marketing, Banco de Dados e programação foram aplicados na formulação do contexto e implementação da ferramenta digital.

4.1 Trabalhos futuros

Integrações com outras tecnologias podem melhorar funcionalidades neste desenvolvidas. Como por exemplo a integração com máquinas de compra de doces, refrigerantes e salgadinhos que funcionam de maneira autônoma poderiam automatizar todo o processo da componente “Resgate de itens”. Bem como a integração com máquinas que, com o avanço da tecnologia, conseguem “ler” certos tipos de materiais recicláveis. Podendo também automatizar parte da funcionalidade de “Doação de resíduos”.

4.2 Lições aprendidas

Tendo em vista as dificuldades tecnológicas encontradas, a construção de sistemas baseada na linguagem Javascript utilizando MongoDB, Express, AngularJS e Node.js (MEAN) são bastante específicas e necessitam de conhecimento aprofundado das ferramentas, da tecnologia de redes de computadores e sistemas operacionais. Dentre as principais dificuldades, destaca-se a necessidade da configuração de acesso ao banco de dados e porta de acesso ao servidor que precisou ser hospedado na nuvem para dar sequencia ao desenvolvimento da aplicação móvel. Esta dificuldade foi contornada mediante à criação de variáveis de ambiente do contexto do Express e do banco de dados. A ajuda para vencer essa dificuldade teve origem em consultas à comunidade de desenvolvedores.

Outra lição aprendida está associada ao entendimento do contexto de coleta de resíduos sólidos, coligindo a realidade da cidade de Manaus e a legislação vigente. Neste caso as barreiras e dificuldades foram mitigadas com a aproximação das autoridades municipais, das cooperativas de coleta de resíduos, empresas locais de reciclagem e do estudo aprofundado da legislação.

4.3 Conclusão

No desenvolvimento deste trabalho foram exercitadas os conhecimentos adquiridos na graduação, sendo necessário a aquisição de novos conhecimentos, e a superação de obstáculos tecnológicos inerentes a tecnologia empregada.

O presente trabalho demonstra a documentação e implementação do protótipo de um ecossistema que é capaz de motivar uma comunidade, bairro, cidade, unidade da federação e até mesmo um país a criar uma estrutura de escoamento e captação de resíduos sólidos envolvendo grande parte dos agentes participantes da geração desses resíduos. De modo a contribuir para a redução da poluição e quantidade de rejeitos.

Anexo I - Diagrama de classes geral do sistema e descrição

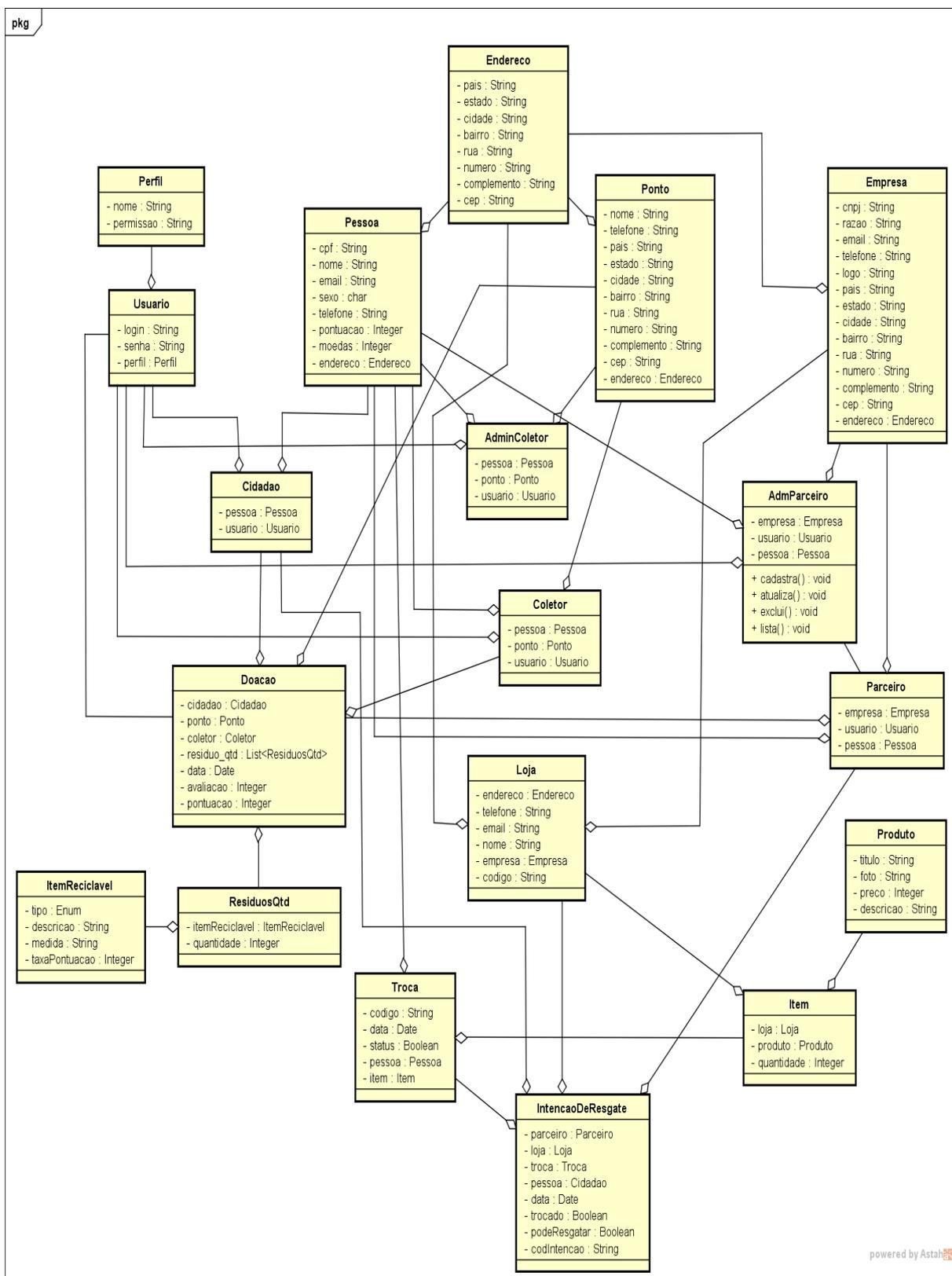


Figura 25 - Diagrama de classes geral do projeto. Fonte: o próprio

O diagrama de classes exibido acima na figura 25, contém todas as classes mapeadas no projeto e seus relacionamentos por composição, onde os atributos de uma classe são utilizados para compor dados da classe alvo do relacionamento de composição.

Todas as classes e atributos do diagrama da figura 25 acima estão armazenadas no banco de dados *No-SQL MongoDB* via biblioteca *Mongoose*. Os relacionamentos por composição foram utilizados para melhor exemplificar a forma que o *Mongoose* trabalha com os relacionamentos entre os documentos *JSON* que são salvos no banco de dados da aplicação.

Tomando como exemplo a classe “Doacao”, localizada no centro a esquerda da figura 25, observa-se que abaixo do nome da classe temos a lista de seus atributos que são: “cidadao” do tipo “Cidadao”; “ponto” do tipo “Ponto”; “coletor” do tipo “Coletor”; “residuo_qtd” do tipo “List<ResiduoQtd>”; “data” do tipo “Date”; “avaliacao” do tipo “Integer”; e “pontuacao” do tipo “Integer”. Os atributos “cidadao”, “ponto”, “coletor” e “residuo_qtd” possuem a tipagem referente às suas classes e estão ligados via composição à classe “Doacao”. Todas as outras classes do diagrama da figura 25 seguem o mesmo exemplo da Classe “Doacao”.

A figura 9, que encontra-se no item 3.3.1.1 do capítulo 3 na página 27 deste trabalho, utiliza o modelo *Mongoose* referente a classe “Doacao” ao na função “Cadastro de doação”. A figura 26 abaixo exhibe o resultado da configuração do mapeamento *Mongoose* feito para a classe “Doacao”.

```

app > models > JS doacao.js > [schema] ponto
1  var mongoose = require('mongoose');
2
3  var schema = mongoose.Schema({
4
5      cidadao: {
6          type: mongoose.Schema.Types.ObjectId,
7          ref: 'Cidadao',
8          required: true
9      },
10     ponto: {
11         type: mongoose.Schema.Types.ObjectId,
12         ref: 'Ponto',
13         required: true
14     },
15     coletor:{
16         type: mongoose.Schema.Types.ObjectId,
17         ref: 'Coletor',
18         required: true
19     },
20     residuo_qtd: [{
21         type: mongoose.Schema.Types.ObjectId,
22         ref: 'Residuoqtd',
23         required: true
24     }],
25     data: {
26         type: Date,
27         required: true
28     },
29     avaliacao: {
30         type: Number,
31         required: true
32     },
33     pontuacao: {
34         type: Number,
35         required: true
36     }
37 });
38
39 mongoose.model('Doacao', schema);

```

Figura 26 - Mapeamento Mongoose da classe "Doacao". Fonte: o próprio.

Anexo II – Diagrama de controle do componente “Resgate de produto” “função criar intenção de resgate”

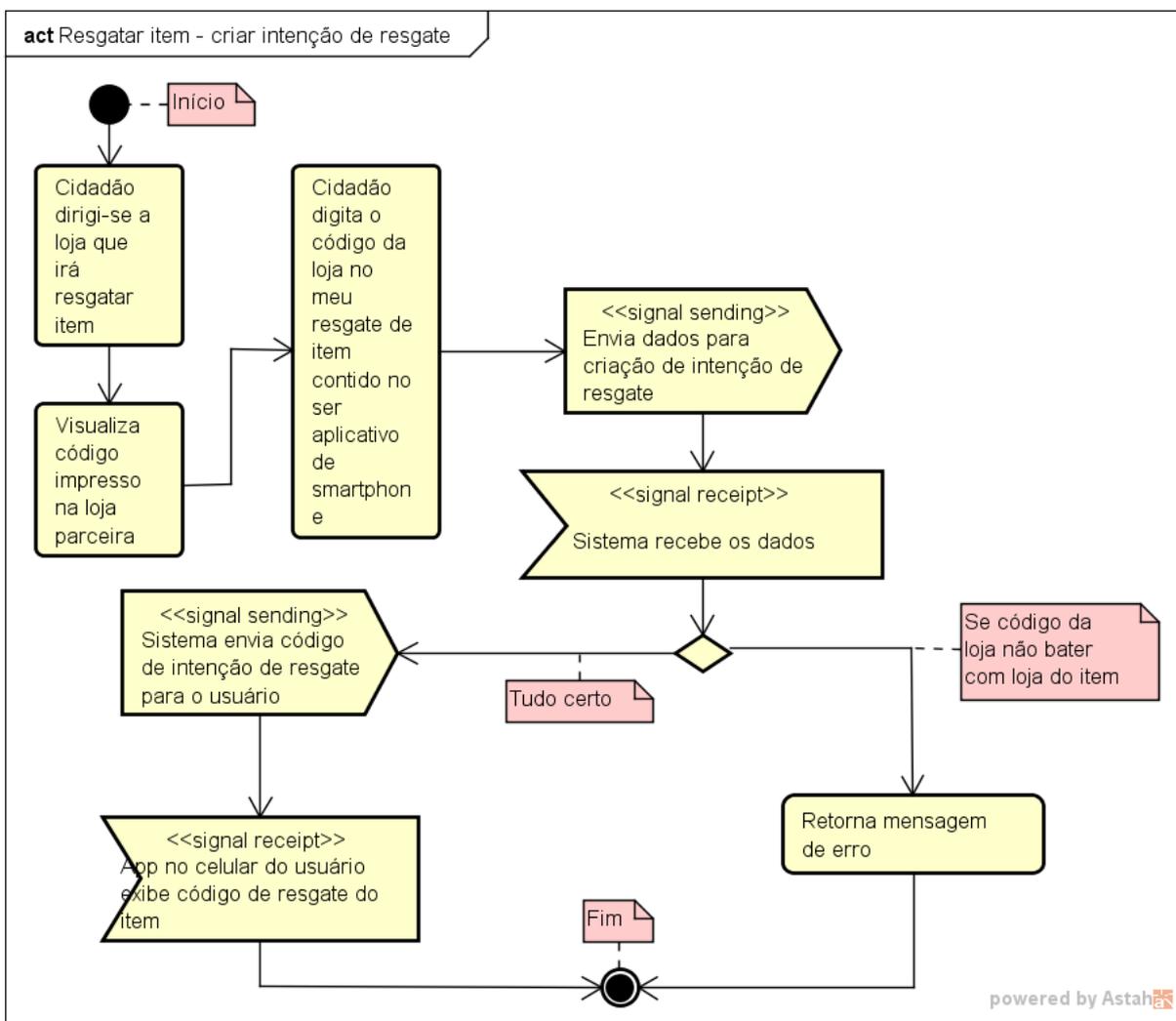


Figura 27 - Diagrama de controle da função "criar intenção de resgate". Fonte: o próprio

O diagrama de controle da figura 27 é um documento complementar a descrição da funcionalidade contida no tópico 3.3.3.1 do capítulo 3.

Referências bibliográficas

ALMEIDA, Flávio. **MEAN**: full stack Javascript para aplicações web com MongoDB, Express, Angular e Node. São Paulo: Casa do Código, 2018. 423 p.

KRIEGER, Maria da Graça et al. **Dicionário de direito ambiental**: terminologia das leis do meio ambiente. 2. ed. Rio de Janeiro: Lexikon, 2008. 296p.

BURKE, Brian. **Gamificar**: Como a gamificação motiva as pessoas a fazerem coisas extraordinárias. São Paulo: DSV Editora, 2015. 192 p.

BRASIL, Lei Federal 2010/12305, § 1 Art. 1, Capítulo V, VII Art. 3, Art. 33, Art. 56.
Disponível em < http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2010/lei/112305.htm >.
Acesso em: 29 nov. 2021.

BUSARELLO, Raul Inácio; UBRICHT, Vania Ribas; FADEL, Luciane Maria. **A gamificação e a sistemática de jogo**: conceitos sobre a gamificação como recurso motivacional. In: FADEL, Luciane Maria et al. (Org.). Gamificação na Educação. 1. ed. São Paulo: Pimenta Cultural, 2014. cap. 1, p. 11-37. Disponível em:
<<https://www.pimentacultural.com/gamificacao-na-educacao>>. Acesso em: 27 out. 2021.

GOIS, Adrian. **Ionic Framework**: Construa aplicativos para todas as plataformas mobile. São Paulo: Casa do Código, 2017. 162 p.

NANI, Everton Luiz. **Meio Ambiente e Reciclagem**: Um Caminho a ser Seguido. 1. ed. Curitiba: Juruá Editora, 2007. 58 p.

PEREIRA, Michael Henrique R.. **AngularJS: Uma abordagem prática e objetiva**. São Paulo: Novatec Editora Ltda, 2014. 208 p.

SEMULSP, **Relatório Anual de Gestão**. Manaus, 2020. 16 p. Disponível em: <<https://semulsp.manaus.am.gov.br/wp-content/uploads/2020/12/Relatorio-Semulsp-2013-a-2020-parcial.pdf> /> Acesso em: 29 nov 2021

ZANIN, Maria; MANCINI, Sandro Donnini. **Resíduos plásticos e reciclagem: aspectos gerais e tecnologia**. 2. ed. São Carlos: EdUFSCAR, 2015. 138 p. Disponível em: <<http://books.scielo.org>>. Acesso em: 27 out. 2017.

FGV, Fundação Getúlio Vargas. **Brasil tem dois dispositivos digitais por habitantes, revela pesquisa da FGV**: O estudo revela ainda que são quatro celulares vendidos para um aparelho de TV no País. 2021. Disponível em: <https://portal.fgv.br/noticias/brasil-tem-dois-dispositivos-digitais-habitante-revela-pesquisa-fgv?utm_source=portal-fgv&utm_medium=fgvnoticias&utm_campaign=fgvnoticias-2021-05-25>. Acesso em: 29 nov. 2021.

IBGE, Instituto Brasileiro de Geografia e Estatística. **Pesquisa nacional por amostra de domicílios contínua PNAD contínua 2018**: pesquisa complementar TIC por pessoas, utilização da internet no período de referências dos últimos três meses. Tabela 2.1.1.2. Disponível em: <https://ftp.ibge.gov.br/Trabalho_e_Rendimento/Pesquisa_Nacional_por_Amostra_de_Domicilios_continua/Anual/Acesso_Internet_Televisao_e_Posse_Telefone_Movel_2018/ods/PNAD_Continua_2018_TIC_tabelas_pessoas_ods.zip>. Acesso em: 29 nov. 2021.

FOWLER, Susan J. **Microserviços prontos para a produção**: Construindo sistemas padronizados em uma organização de engenharia de software. São Paulo: Novatec Editora Ltda, 2019. 224 p.

FLATICON, **Ícones da Figura 3, Figura 4 e Ícones utilizados no sistema**. Disponível em: <<https://www.flaticon.com/>>. Acesso em: 06 out. 2021.