

Universidade Regional de Blumenau – FURB  
Departamento de Sistemas e Computação  
Curso de Sistemas de Informação  
Desenvolvimento de Plataforma Móveis

DESPACK  
Versão 1.0

Bruna Schroeder  
Guilherme Barth  
Henrique A. Eichstädt  
Lucas S. Klüser  
Matheus L. S. Boing

Blumenau, 30 de maio de 2022

## 1. INTRODUÇÃO

Para pessoas que desejam descartar resíduos/materiais e sentem dificuldade em localizar pontos de coleta, o Despack é um aplicativo móvel que facilita o acesso dos usuários aos pontos de coleta, diferente de sites de busca, que não contemplam o descarte de diferentes tipos de resíduos, logo, o Despack direciona o usuário ao ponto de coleta que melhor se encaixa naquele tipo de resíduo baseado na localização.

<b>É</b> <ul style="list-style-type: none"><li>- Aplicativo móvel</li><li>- Plataforma C2C</li><li>- Mapeador</li></ul>	<b>NÃO É</b> <ul style="list-style-type: none"><li>- Aplicativo de coleta de resíduos</li><li>- Aplicativo de horários de coleta de limpeza urbana</li><li>- Aplicativo de rastreamento de coletores de resíduos</li></ul>
<b>FAZ</b> <ul style="list-style-type: none"><li>- Disponibiliza mapa com os locais dos pontos de coleta</li><li>- Informa sobre descarte consciente</li></ul>	<b>NÃO FAZ</b> <ul style="list-style-type: none"><li>- Gerenciamento de coleta de resíduos</li><li>- Gerenciamento de tratamento do resíduo</li></ul>

## 2. OBJETIVOS

**OB01** – Facilitar a busca de pontos de coleta de resíduos

**OB02** – Promover a cultura do descarte consciente

**OB03** – Reduzir a poluição ambiental por lixo

## 3. PERSONAS

**PS01** – Miguel tem 28 anos e sempre soube que cada tipo de resíduo deve ser descartado de forma específica, porém nunca soube como e onde o fazê-lo

**PS02** – Sara tem 18 anos e está indo morar sozinha pela primeira vez. Ela não sabe onde descartar os resíduos de lixo corretamente

**PS03** - O supermercado Giassi, como um ponto de coleta de resíduos, tem interesse em divulgar isto para a comunidade, bem como informar quais tipos de resíduos são descartáveis no ponto de coleta deles

## 4. USER STORIES

**H01** – Como praticante da coleta consciente, quero poder buscar pontos de coleta em um mapa para facilitar a busca pelo local mais próximo

**H02** - Como cidadão, quero buscar pontos de coleta para efetuar o descarte correto de resíduos

**H03** – Como cidadão, quero poder obter informações úteis sobre coleta e reciclagem correta ao acessar o aplicativo para que eu possa contribuir para o descarte correto

**H04** - Como usuário da aplicação, quero propor alterações no ponto de coleta como novo endereço ou sinalizar seu fechamento permanente para manter as informações sempre atualizadas

**H05** - Como usuário da aplicação, quero poder efetuar um cadastro de ponto de coleta novo para tornar a comunidade ainda mais ativa

## **5. REQUISITOS FUNCIONAIS**

**RF01** – O sistema deverá ter um cadastro de ponto de coleta

**RF02** – O sistema deverá ter login e senha para que cada usuário seja cadastrado no sistema

**RF03** – O sistema deverá ter os pontos de descarte mais próximo a localização atual do usuário

**RF04** – O sistema deverá ter avaliação do ponto de descarte

**RF05** – O sistema deverá permitir aos usuários informar se o ponto de coleta está realmente disponível por meio de avaliação

## **6. REQUISITOS NÃO FUNCIONAIS**

**RNF01** – O sistema será um aplicativo para dispositivos móveis

**RNF02** – O sistema deverá possuir integração com o Google Maps

**RNF03** – O sistema deverá possuir integração com o recurso nativo de câmera do dispositivo móvel

**RNF04** – O sistema persistirá dados com banco de dados SQL Server

**RNF05** – O banco de dados será publicado em nuvem pública Microsoft Azure

**RNF06** – O sistema será publicado em nuvem pública Heroku

**RNF07** – O sistema deverá solicitar uma confirmação de e-mail após o cadastro por questões de segurança.

**RNF08** – O sistema deverá estar disponível 24 horas por dia, 7 dias por semana

## **7. REGRAS DE NEGÓCIO**

**NEG01** – Cada usuário poderá avaliar apenas uma vez se o ponto de coleta está ou não disponível

**NEG02** – Caso a avaliação do ponto de coleta aponte 10 indicações de que o ponto está inativo, o ponto deverá ser inativado

**NEG03** – Não poderá permitir mais de um ponto de coleta no mesmo endereço

## 8. LEAN CANVAS

Lean Canvas		DESPACK		15-09-2021
				Versão 1
<b>Problema</b> . Cidadãos ou empresas que não sabem onde ou como descartar resíduos corretamente, ou lugares que querem ser um ponto de coleta destes resíduos.	<b>Solução</b> . Aplicação móvel que conecta usuários e pontos de coleta de resíduos e informa sobre descarte correto de resíduos	<b>Proposta de Valor</b> . Única plataforma que filtra o tipo do resíduo para descarte e provê informações precisas baseadas na localização do usuário.	<b>Vantagem Competitiva</b> . Primeira ferramenta para o objetivo no Brasil, o que vai refletir em cidades mais limpas com resíduos propriamente descartados.	<b>Segmento de Clientes</b> . Cidadãos em busca de descarte correto de resíduos e pontos de coleta em busca de informar a comunidade sobre este serviço.
	<b>Métricas-Chave</b> . Quantidade de pontos de coleta; . Quantidade de resíduos corretamente coletados.		<b>Canais</b> . Redes sociais (facebook e instagram).	
<b>Estrutura de Custos</b> . Hospedagem do servidor. . Publicação do aplicativo em loja. . Marketing. . Desenvolvimento do aplicativo			<b>Fontes de Receita</b> . Propagandas. . Doações	
PRODUTO			MERCADO	

## 9. PROTÓTIPO

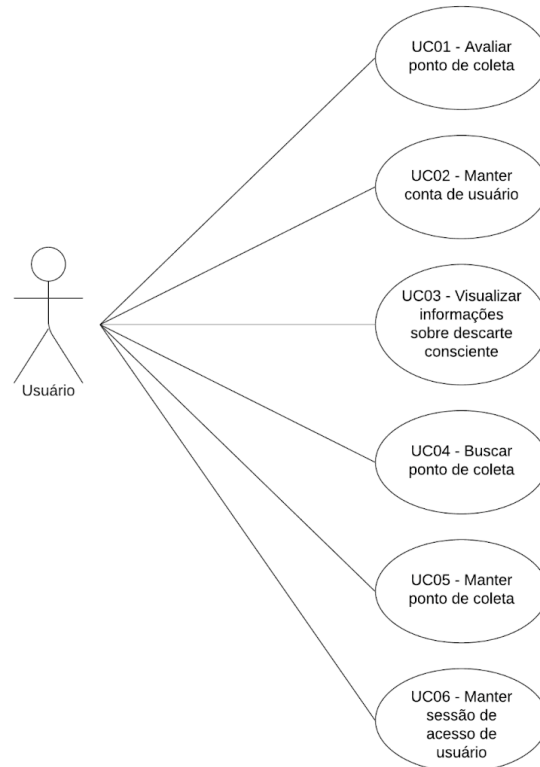
<https://www.figma.com/file/PbLpll4nPXVluAF2IFqSy7/App-de-coleta?node-id=21617%3A260>

## 10. DIAGRAMA DE CASO DE USO

[https://lucid.app/lucidchart/2bdfda6b-b993-4e5b-b6a1-8c8526e9a45f/edit?invitationId=inv\\_9179aa9d-e63b-4c32-bdf4-228aba685a61&page=.Q4MUjXso07N#](https://lucid.app/lucidchart/2bdfda6b-b993-4e5b-b6a1-8c8526e9a45f/edit?invitationId=inv_9179aa9d-e63b-4c32-bdf4-228aba685a61&page=.Q4MUjXso07N#)

uc 3.4. Diagrama de Casos de Uso

DESPACK



**Caso de uso:** UC06 - Manter sessão de acesso de usuário

**Ator:** Usuário

**Pré-condições:** Usuário deve ter uma conta cadastrada e ativa.

**Pós-condições:** Sessão de acesso do usuário ao aplicativo iniciada/encerrada.

**Cenário Principal:**

1. Usuário abre o aplicativo;
2. Usuário digita seu e-mail e sua senha;
3. Usuário aciona o botão de autenticação;
4. A sessão é iniciada.

**Cenário de Exceção (e-mail inválido):**

No passo 3 do cenário principal, caso o usuário digite um e-mail inválido:

1. Uma mensagem de erro é exibida;
2. Volta para o passo 2 do cenário principal.

**Cenário de Exceção (senha inválida):**

No passo 3 do cenário principal, caso o usuário digite uma senha inválida:

3. Uma mensagem de erro é exibida;
4. Volta para o passo 2 do cenário principal.

**Cenário de Exceção (e-mail não confirmado):**

No passo 3 do cenário principal, caso o e-mail do usuário não tenha sido confirmado:

5. Uma mensagem de erro é exibida;
6. Volta para o passo 2 do cenário principal.

**Cenário Alternativo (encerrar sessão):**

1. Usuário abre o menu do aplicativo;
2. Usuário aciona o botão de encerrar sessão;
3. A sessão de acesso do usuário é encerrada.

**Caso de uso:** UC02 - Manter conta de usuário

**Ator:** Usuário

**Objetivo:** Garantir o processo de inclusão, alteração e exclusão.

**Pré-condições:** Novo usuário do sistema ou um existente com necessidade de alterar ou excluir sua conta.

**Pós-condições:** Inclusão/Alteração/Exclusão do cadastro com sucesso.

**Cenário Principal (Criação de conta):**

1. Usuário abre aplicativo e acessa página de cadastro de novo usuário.
2. Usuário insere os dados solicitados.
3. Usuário submete o formulário.
4. Sistema valida os dados.
5. Sistema finaliza o cadastro com sucesso e navega para a página principal da aplicação.

**Cenário Alternativo (Atualização de conta):**

1. Usuário abre aplicativo.
2. Usuário efetua autenticação.
3. Usuário acessa página de cadastro do usuário
4. Usuário efetua edição nos campos desejados
5. Usuário submete o formulário.
6. Sistema valida os dados.
7. Sistema conclui atualização de dados e retorna mensagem de sucesso.

**Cenário Alternativo (Exclusão de conta):**

1. Usuário abre aplicativo.
2. Usuário efetua autenticação.
3. Usuário acessa página de cadastro de usuário.
4. Usuário clica em excluir conta.

5. Sistema exibe mensagem de confirmação de exclusão e formulário para informar a senha do usuário.
6. Usuário informa a senha.
7. Usuário submete o formulário.
8. Sistema conclui exclusão de conta e retorna para a tela de login do sistema.

**Cenário Alternativo (Recuperação de senha):**

1. Usuário abre aplicativo.
2. Sistema exige dados de login.
3. Usuário clica em recuperação de senha.
4. Sistema envia e-mail com link para troca de senha.
5. Usuário acessa link e efetua atualização de senha.

**Cenário de Exceção (Criação de conta):**

No passo 4, caso tenha algum dado mal preenchido, o sistema retorna mensagem de erro com informações sobre o erro no cadastro e retorna para o passo 2.

**Cenário de Exceção (Atualização de conta):**

No passo 6, caso tenha algum dado mal preenchido, o sistema retorna mensagem de erro com informações sobre o erro no cadastro e retorna para o passo 4.

**Cenário de Exceção (Exclusão de conta):**

No passo 7, caso o usuário preencha sua senha de forma incorreta, o sistema retorna mensagem de erro informando que a senha está incorreta e retorna para o passo 6.

**Caso de uso: UC04 - Buscar ponto de coleta**

**Ator:** Usuário

**Objetivo:** Encontrar um ponto de coleta

**Pré-condições:** Sessão de acesso do usuário iniciada

**Pós-condições:** Informações do ponto de coleta em exibição

**Cenário Principal:**

1. Usuário informa o endereço que deseja buscar;
2. O aplicativo apresenta os pontos de coletas disponíveis na região do endereço;
3. O usuário seleciona um ponto de coleta;
4. As informações do ponto de coleta são exibidas.

**Cenário de Exceção (endereço inválido):**

No passo 1 do cenário principal, caso o usuário digite um endereço inválido:

1. Uma mensagem de erro é exibida;
2. Volta para o passo 1 do cenário principal.

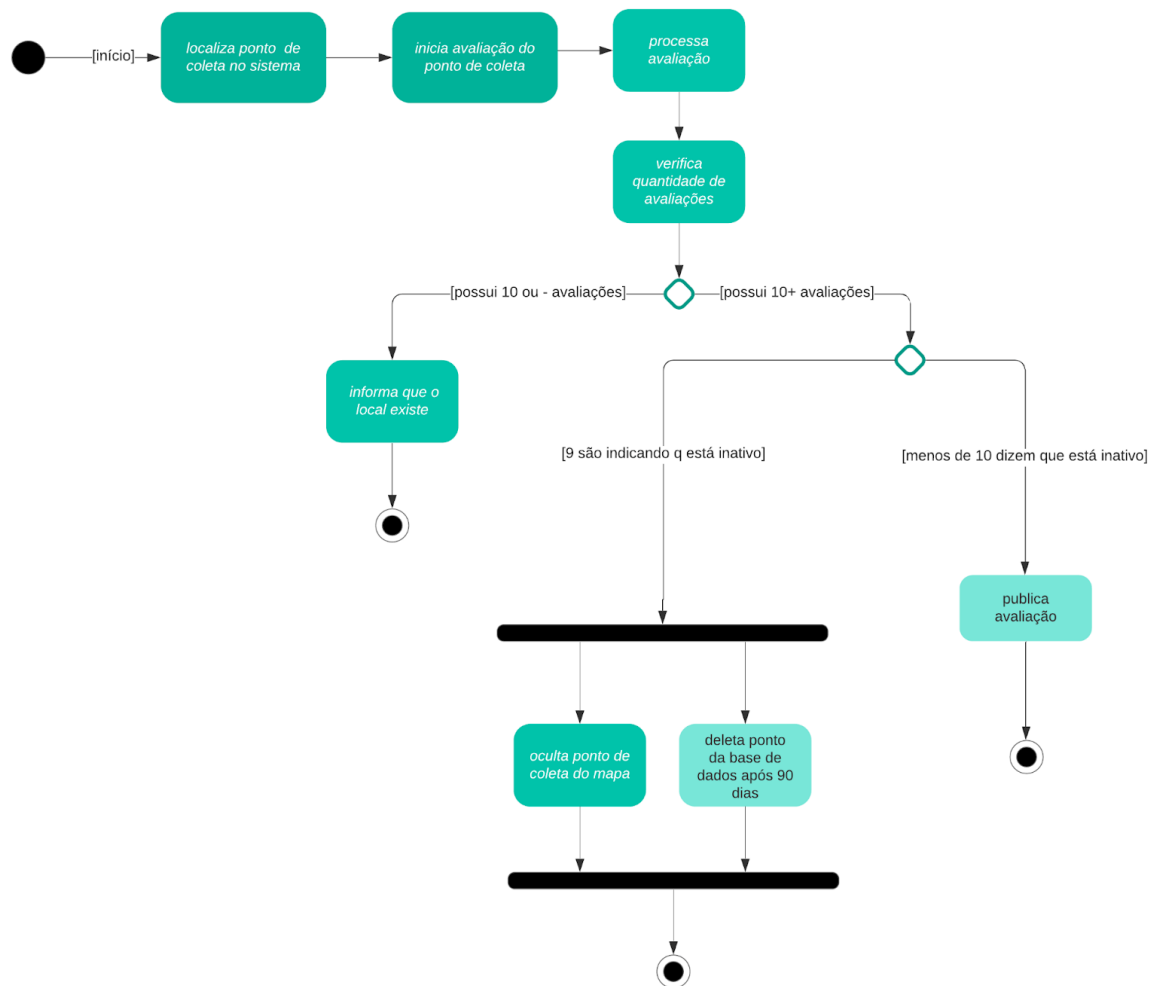
### Cenário Alternativo (Utilizar o GPS):

1. Usuário abre o aplicativo;
2. Usuário aciona a barra de pesquisa de endereço;
3. Usuário clica no botão de GPS;
4. Sistema retorna localização atual do usuário.

## 11. DIAGRAMA DE ATIVIDADES

Processo do diagrama: avaliação do ponto de coleta

[https://lucid.app/lucidchart/061d7128-c9d8-4536-b6e8-0e8470427eb6/edit?invitationId=inv\\_4609f8a0-fc33-4349-9788-783d1bebb7b1&page=0\\_0#](https://lucid.app/lucidchart/061d7128-c9d8-4536-b6e8-0e8470427eb6/edit?invitationId=inv_4609f8a0-fc33-4349-9788-783d1bebb7b1&page=0_0#)

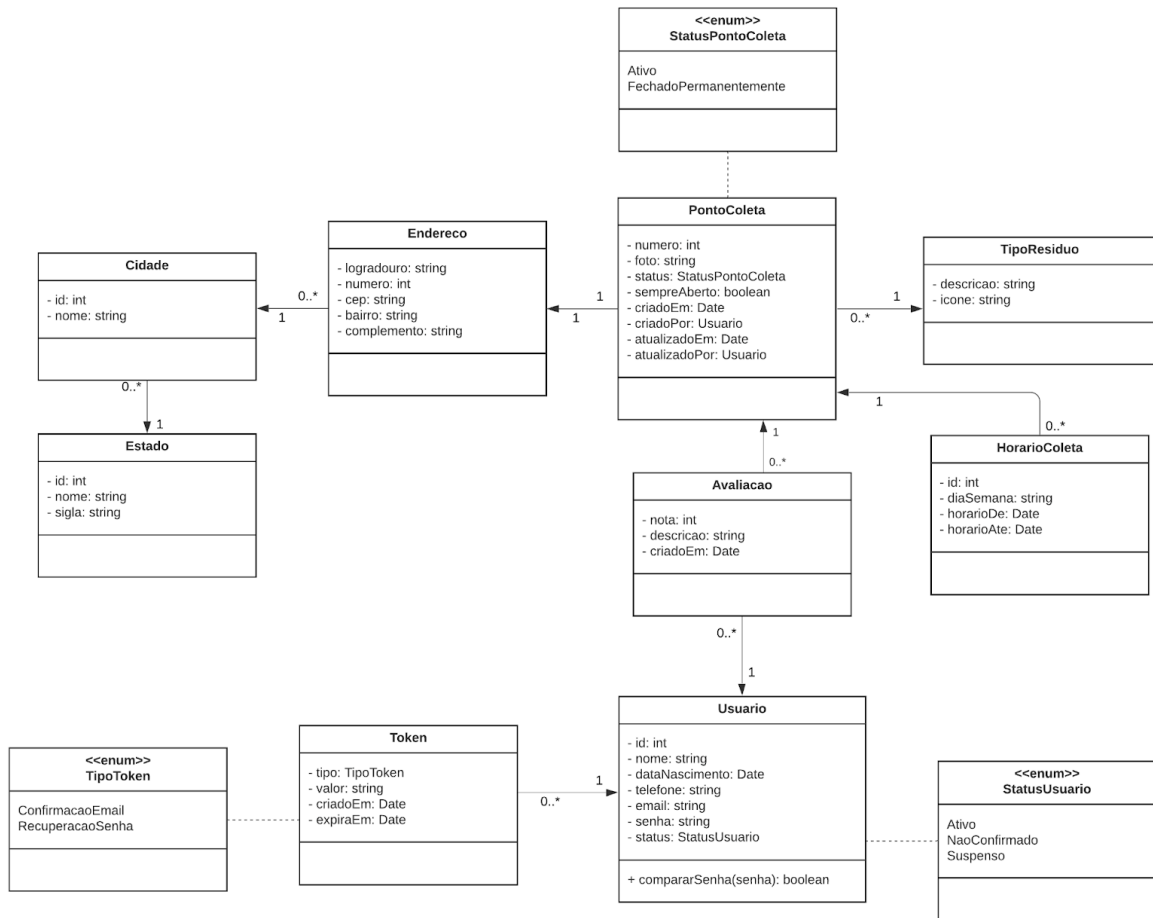


## 12. DIAGRAMA DE CLASSES

[https://lucid.app/lucidchart/7bd11fd3-6ab1-4316-a7b8-6d8727eccc33/edit?viewport\\_loc=-51%2C-21%2C2626%2C1261%2CHWEp-vi-RSFO&invitationId=inv\\_ee7afc29-f6e2-49c0-8fc6-3108ccaa6458](https://lucid.app/lucidchart/7bd11fd3-6ab1-4316-a7b8-6d8727eccc33/edit?viewport_loc=-51%2C-21%2C2626%2C1261%2CHWEp-vi-RSFO&invitationId=inv_ee7afc29-f6e2-49c0-8fc6-3108ccaa6458)

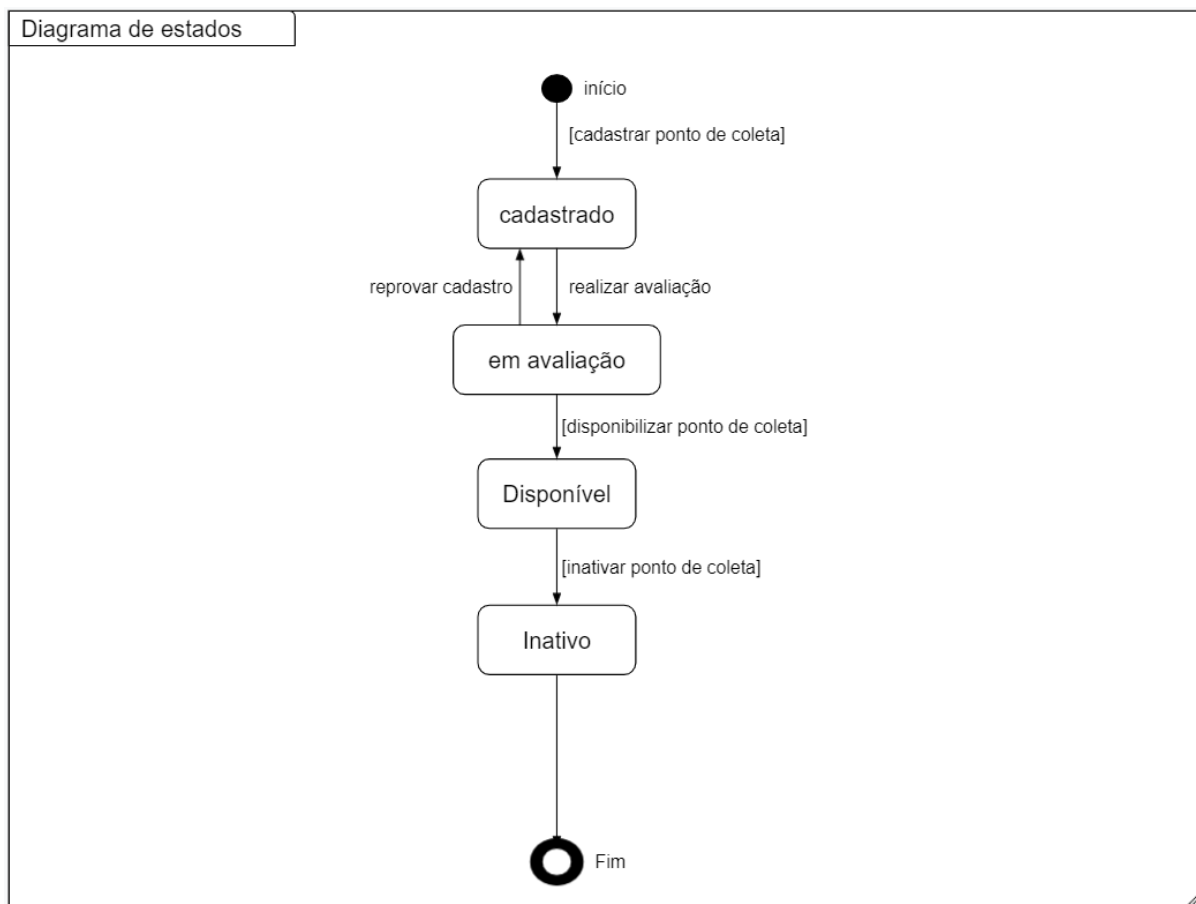


## Projeto Despack - Diagrama de Classes



## 13. DIAGRAMA DE ESTADOS

[https://docs.google.com/drawings/d/1ykAkGTcT1QzwrFG\\_iGklInslgUbbq7qUQU7Teby2-Lk/edit](https://docs.google.com/drawings/d/1ykAkGTcT1QzwrFG_iGklInslgUbbq7qUQU7Teby2-Lk/edit)



## GLOSSÁRIO

**Azure** – Plataforma de computação em nuvem.

**AWS** – Plataforma de computação em nuvem.

**C2C** – *Customer to customer*; modelo de negociação/transação entre consumidores (usuários).

**Descarte** – Ação ou efeito de descartar algo que não é mais útil ou aquilo que não se quer mais.

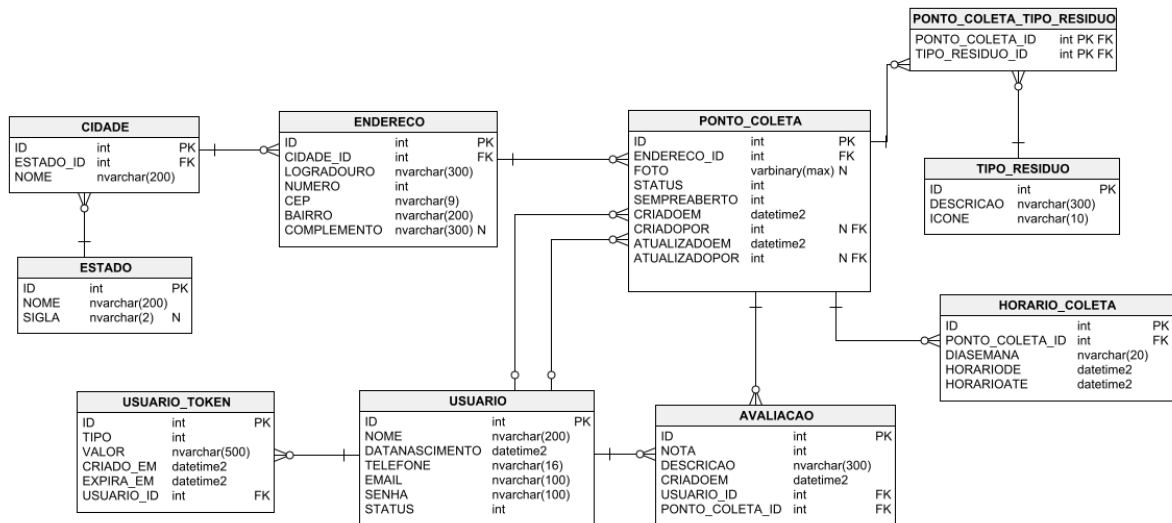
**MySQL** – Sistema de gerenciamento de banco de dados que utiliza a linguagem SQL como interface.

**Ponto de Coleta** – Locais - normalmente estabelecimentos comerciais - que contam com um espaço disponível para o recebimento, armazenamento e, posteriormente, entrega de resíduos para o descarte.

**Resíduo** – Todo material, substância, objeto ou bem descartado resultante de atividades humanas em sociedade, tais como: seringas, luvas, baterias e embalagens.

**Token** – código que armazena objetos JSON com os dados que permitem a autenticação da requisição.

## MODELO ENTIDADE RELACIONAMENTO (MER)



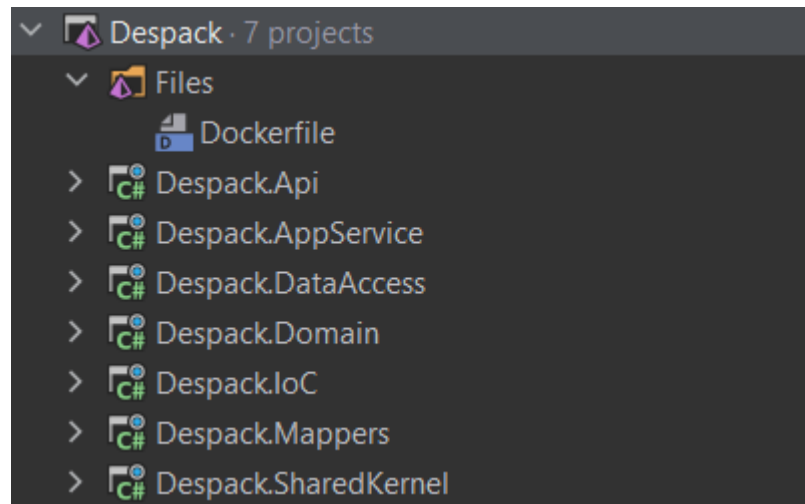
## IMPLEMENTAÇÃO

### 1. APPLICATION PROGRAMMING INTERFACE (API)

A Application Programming Interface (API) será desenvolvida utilizando tecnologias Microsoft como a linguagem orientada a objetos C# e demais frameworks desenvolvidos para o .NET 6. A API contará com uma documentação Swagger, que permite fazer o design e gerar as interfaces de uma API em forma de documentação, facilitando seu uso pelos desenvolvedores da aplicação cliente.

Para persistência de dados, será utilizado o banco de dados relacional SQL Server, que será hospedado em nuvem pública a partir dos recursos de computação em nuvem da Microsoft Azure. Para a criação das estruturas do banco de dados, será utilizado o Object Relational Mapper (ORM) Entity Framework Core, também desenvolvido pela Microsoft, ao qual permite configurar a estrutura do banco de dados da aplicação por meio de migrações, gerar as estruturas e efetuar todo o processo de Create, Read, Update, Delete (CRUD).

A API será desenvolvida utilizando a arquitetura em camadas, onde a camada de regra de estrutura de API, regra de negócios e acesso a dados serão separadas em projetos distintos, permitindo uma maior organização de código e separação das responsabilidades de cada projeto.



Estrutura da solução da API

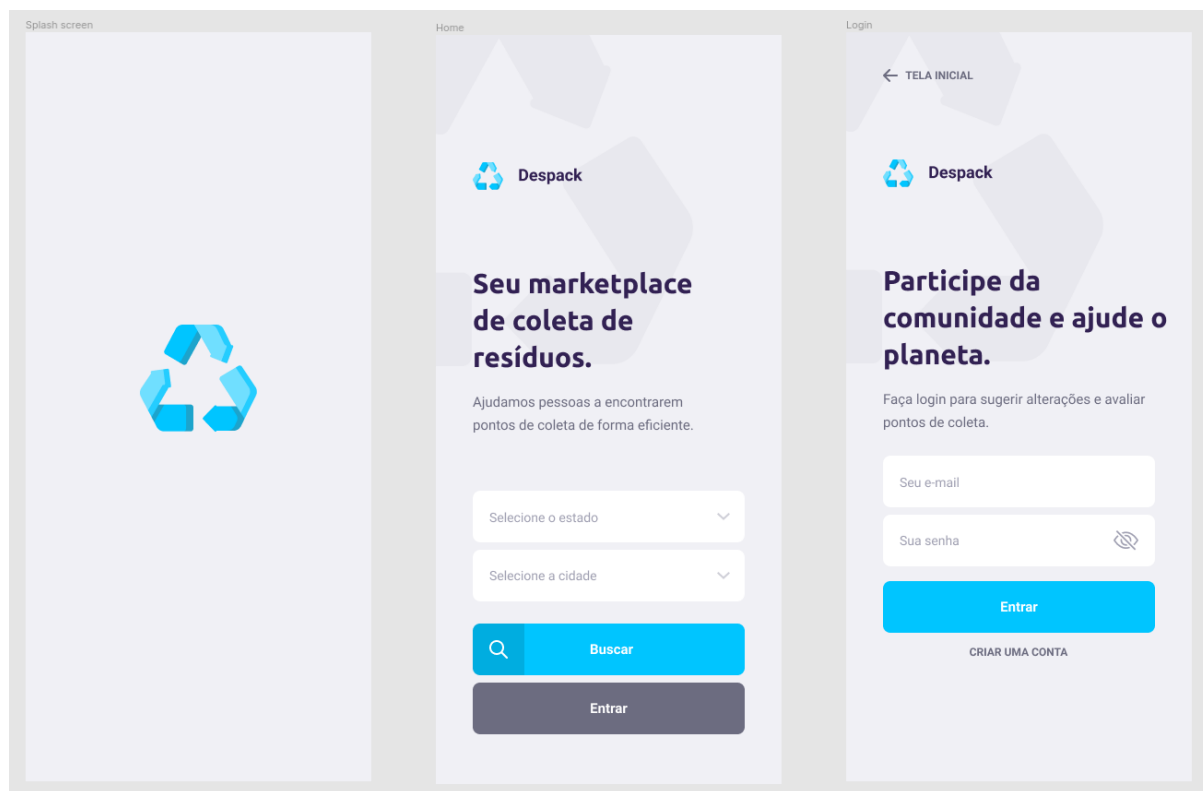
Será utilizado também princípios de Domain Driven Design (DDD), onde conforme estrutura apresentada na figura anterior, o projeto Despack.Api que é responsável pelo design da API, não tem acesso direto ao projeto Despack.AppService que é responsável pela regra de negócio da aplicação. As chamadas para regras de negócio são feitas através do uso dos princípios de Dependency Injection (DI) e Inversion of Control (IoC), que utilizam interfaces para instanciar de forma concreta um objeto no projeto de design da API.

A publicação da API será feita utilizando recursos de cloud da Heroku, utilizando uma imagem docker que será publicada na nuvem e será disponibilizada para acesso a partir do link <http://despack.herokuapp.com/swagger/index.html>.

## 2. APLICATIVO MÓVEL

O aplicativo móvel será desenvolvido utilizando a linguagem de programação Javascript junto do framework React Native. Embora o framework permita a compilação para dispositivos iOS, apenas dispositivos Android serão utilizados nos testes durante a construção do aplicativo móvel.

A comunicação entre o aplicativo móvel e a API do projeto Despack se dará através do protocolo de comunicação HTTP, e não haverá replicação ou sincronia de dados entre a API e o aplicativo móvel, portanto, não será possível utilizar o aplicativo sem uma comunicação estável com a internet.



Exemplo de telas implementadas no aplicativo móvel