



Introdução

Swi2 é uma aplicação web/mobile desenvolvida a partir de uma iniciativa entre Pedro Rabis e Yan Stivaletti. Sua idealização surgiu de um pensamento ecossocialista associado a produção em massa de artigos de vestuário e a degradação do meio ambiente que o mesmo gera. O aplicativo tem como função facilitar a troca de itens entre usuários que não desejam mais seus pertences ou que anseiam por outros itens, de forma que cria-se um canal comunitário para estabelecerem tais trocas de maneira justa e bem assegurada.

A indústria da moda tem crescido em níveis exponenciais desde a globalização, sendo responsável por grande parte do capital de várias empresas e países. Apesar do alto lucro e grande movimentação de mercado que o ramo da moda gera, muitas vezes, esquece-se que a mesma é responsável por impactos ambientais, como o uso exagerado de água para a produção de peças e a grande quantidade de lixo plástico, e impactos sociais relacionados a quase escravidão de funcionários para produzirem peças e a criação de uma indústria de pirataria e falsificação que cresce proporcionalmente ao ramo da moda.

Para solucionar alguns desses problemas o aplicativo Swi2 se apresenta como uma ótima opção, a possibilidade de trocar itens evita o consumo desnecessário de artigos que seriam facilmente descartados ou esquecidos e com este baixo consumo evita-se também a hiperprodução dos mesmos. Reduzindo a criação de artigos de moda nós conseguimos gerar não apenas um impacto ambiental, mas também social e por estes motivos o Swi2 gera um impacto econômico social capaz de reparar alguns danos que o indivíduo social causa no planeta.

Glossário

- **Ecossocialismo:** O ecossocialismo, socialismo verde, socialismo ecologista ou, ainda, ecomarxismo é uma ideologia que conjuga elementos do socialismo com a ecologia, a política verde e, por vezes, o altermundialismo.
- **Indústria da Moda:** A indústria da moda inclui uma grande diversidade de atividades econômicas, indo desde a criação de modelos de roupa (design de moda) mais ou menos personalizados até à produção de vestuário em série.
- **Produção em Massa:** Produção em massa é a expressão utilizada para denominar uma estratégia de produção para estoque, também conhecida pela expressão inglesa make-to-stock. A ideia é a empresa produzir o suficiente para ter sempre o produto em estoque e, dessa forma, diminuir o prazo de entrega aos clientes.
- **Hiperprodução:** Produção acima do que é viável comercializar, que supera a demanda; excesso de produção.

Definição de Requisitos do Usuário

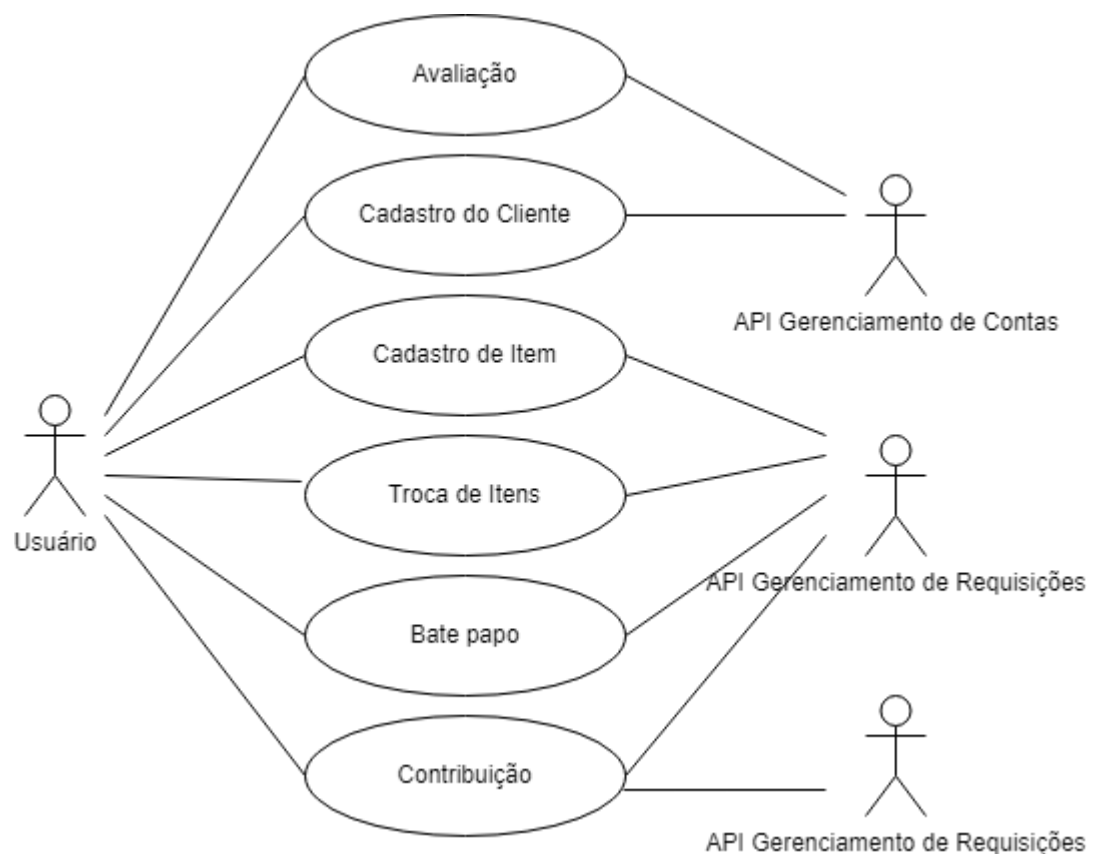
Dentro do Swi2 o usuário poderá conversar com outros usuários para fornecer ou adquirir informações sobre a troca que deseja realizar, a validação e cadastros regulados são extremamente importantes para manter o ambiente comunitário que o aplicativo pretende oferecer.

O cadastro de produtos ofertados pelo usuário é dado de forma que comprovantes de compra e cupons fiscais devem ser inseridos e validados para assegurar a existência do item em questão. Quanto mais produtos forem trocados pelo usuário de forma bem sucedida, aumenta-se a capacidade de produtos que o indivíduo pode ofertar dentro de seu perfil.

A procura de itens específicos pode ser usada para filtrar a busca do cliente dentro do Swi2, a mesma pode ser categorizada por especificação de item, local ou usuário.

O sistema de chat do aplicativo busca manter um diálogo entre dois usuários para que ambos possam planejar e realizar as trocas de produtos de maneira rápida e diplomática.

Para a avaliação de usuários cria-se um sistema de 'gameificação', após o recebimento da doação o cliente cadastrado pode dar um certo número de estrelas (de 1 a 5) a entrega do item, certificando a qualidade dos itens que outro usuário coloca em sua loja e fortalecendo o senso comunitário da aplicação.

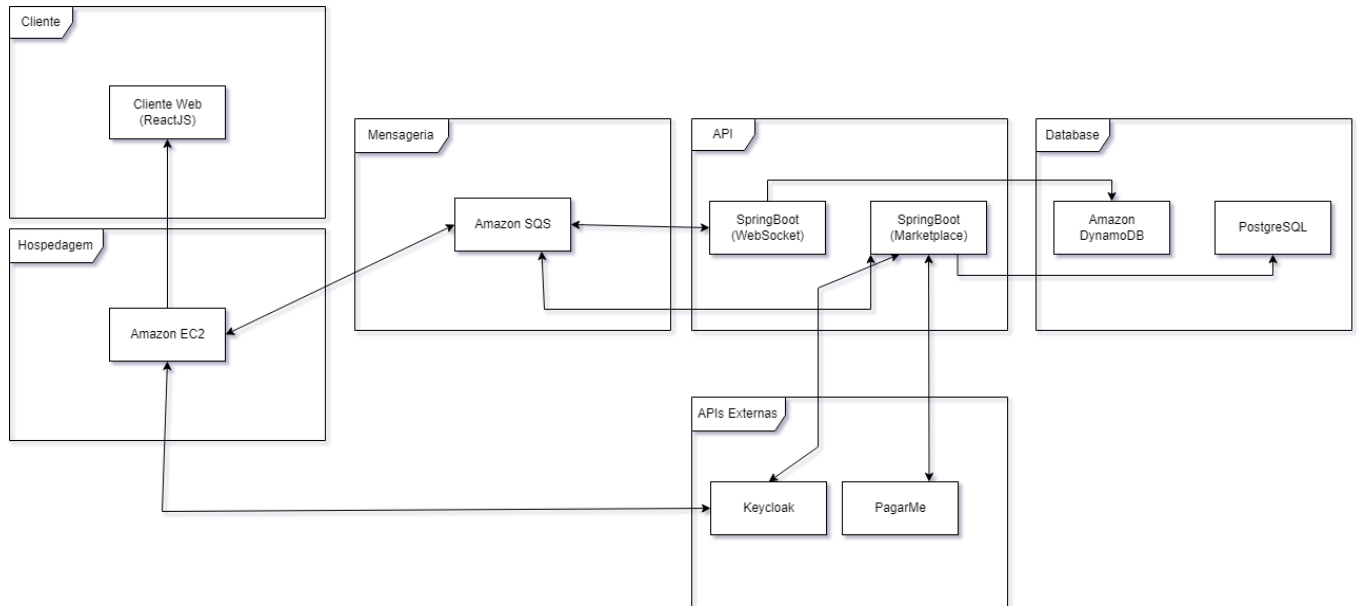


Arquitetura do Sistema

Softwares utilizados:

- Design e implementação de conceitos: Draw.io, Figma;
- Implementação de interfaces: ReactJS;

- Hospedagens de Software e Dados: Amazon S3, PostgreSQL, Amazon Dynamo;
- API's internas e externas: Spring Boot (websocket & marketplace), Keycloak;



Requisitos do Sistema

Os requisitos da aplicação são padrões de um aplicativo baseado em buscas em localidades próximas, o sistema deverá ter um banco de dados contendo as informações de todos os usuários, histórico de compras, avaliação do mesmo de acordo com outros usuários. O sistema deverá ser integrado ao GPS para buscar as ofertas de troca mais próximas, consequentemente é necessário também uma conexão constante com a internet para permitir uso do GPS e do chat dentro da aplicação. Também deve-se ter sistema de especificação de itens e filtragem de artigos para otimizar a busca do usuário dentro do Swi2.

Casos de Uso

UC01 - Cadastro de Cliente

| | |
|----------------------|--|
| Atores | Cliente; Keycloak |
| Interessados | Realizadores do Projeto e Instituições Parceiras |
| Pré-condições | Nenhum |
| Pós-condições | Validação do cadastro por email |
| Visão Geral | Processo de cadastro de um Cliente no sistema, sendo este sua primeira |

| | |
|--|-----------------------------|
| | experiência com a aplicação |
|--|-----------------------------|

Cenário Principal de Sucesso

1. O **sistema** apresenta uma tela exigindo o e-mail, a senha do cliente e uma opção de cadastro;
2. O **cliente** acessa a opção de novo cadastro;
3. O **sistema** apresenta uma tela exigindo nome, e-mail e senha do cliente;
4. O **cliente** preenche os dados necessários e confirma o processo;
5. O **sistema** envia um email de verificação para o cliente, com link para confirmação de cadastro;
6. O **cliente** acessa o link de confirmação de cadastro;
7. O **sistema** confirma o cadastro e concede acesso ao cliente.

Cenários Alternativos: Exceções

Exceção 1: Falha na confirmação de cadastro

1. O **cliente** volta ao passo 2 (opção de novo cadastro).

Exceção 2: Link de confirmação com tempo limite excedido

1. O **cliente** volta ao passo 2 (opção de novo cadastro).

UC02 - Cadastro de Item

| | |
|----------------------|---|
| Atores | Cliente, Amazon SQS |
| Interessados | Cliente |
| Pré-condições | Cliente cadastrado; Cliente logado. |
| Pós-condições | Nenhum |
| Visão Geral | Processo de cadastro de um Item novo no sistema, sendo este feito por um Cliente logado no sistema. |

Cenário Principal de Sucesso

1. O **cliente** acessa a página do próprio perfil;
2. O **sistema** apresenta uma opção para cadastro de Item novo;
3. O **cliente** acessa a opção de novo cadastro de Item;
4. O **sistema** apresenta uma tela exigindo nome, cidade, fotos, tipo, marca e tempo de uso do novo item;
5. O **cliente** preenche os dados necessários e confirma o processo;
6. O **sistema** cadastra o novo item e o deixa visível na tela de perfil deste usuário;

Cenários Alternativos: Exceções

Exceção 1: Cliente desiste do cadastro

1. O **cliente** sai do passo 4 (tela de cadastro);
2. O **sistema** exibe um aviso de cancelamento do cadastro do Item;
3. O **cliente** confirma a opção de cancelamento;
4. O **sistema** cancela o procedimento de cadastro.

Exceção 2: Informações inválidas

1. O **cliente** preenche os dados com informações inválidas;
2. O **sistema** exibe mensagem que as informações são inválidas e não continua o processo de cadastro até que as mesmas sejam validadas.

Cenários Alternativos: Variantes

Variante 1: Cliente deseja ocultar o Item

1. O **cliente** seleciona a opção de 'oculto';
2. O **sistema** cadastra o Item, mas não o deixa visível.

UC03 - Troca de Item

| | |
|----------------------|--|
| Atores | Cliente, Amazon SQS |
| Interessados | Realizadores do Projeto, Instituições Parceiras e Cliente |
| Pré-condições | Clientes cadastrados; Itens cadastrados |
| Pós-condições | Emissão de Token de troca de Item |
| Visão Geral | Processo de troca entre Itens de vestuário de dois Cliente |

Cenário Principal de Sucesso

1. O **sistema** apresenta uma tela informando os Itens disponíveis na cidade do Cliente;
2. O **cliente** preenche um formulário para filtragem e/ou preenche parcialmente o nome do Item desejado;
3. O **sistema** apresenta uma tela contendo os Itens solicitados;
4. O **cliente** escolhe a opção de Item que melhor lhe atende;
5. O **sistema** apresenta uma tela com informações detalhadas do Cliente que cadastrou o Item e do Item em si;
6. O **cliente** entra em contato com o Cliente que cadastrou o item e ambos iniciam a troca;
7. O **sistema** emite um Token para a troca de Item;
8. O **cliente** efetua e confirma a troca de itens;
9. O **sistema** valida o Token de troca de Item.

Cenários Alternativos: Variantes

Variante 1: Nova busca

1. O **cliente** escolhe a opção de realizar uma nova busca;

2. O **sistema** retorna ao passo 1 (informar os itens disponíveis).

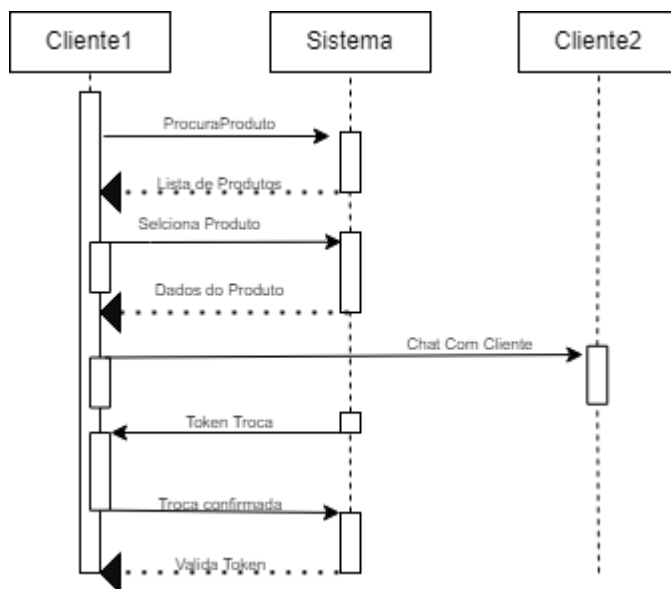
Cenários Alternativos: Exceções

Exceção 1: Cancelamento de Troca

1. O **cliente** escolhe a opção de cancelar a Troca;
2. O **sistema** altera o estado da Troca para 'cancelado' e invalida os Tokens.

Exceção 2: Não geração de Tokens

1. O **sistema** envia os Tokens para os dois Clientes envolvidos na Troca;
2. O **cliente** tenta validar a Troca;
3. O **sistema** não aceita a validação da Troca
4. O **cliente** seleciona a opção de 'renovar Tokens';
5. O **sistema** envia novos Tokens válidos.



UC04 - Bate-papo entre 2 Clientes

| | |
|----------------------|--|
| Atores | Cliente |
| Interessados | Cliente |
| Pré-condições | Clientes cadastrados; Clientes logados |
| Pós-condições | Nenhum |
| Visão Geral | Processo de troca de mensagens em tempo real entre Clientes interessados em realizar troca de Item |

Cenário Principal de Sucesso

1. O **cliente** entra em contato com o Cliente que cadastrou o Item que o mesmo

deseja;

2. O **sistema** apresenta uma tela de chat;
3. O **cliente** efetua a troca de mensagens com outro Cliente;
4. O **sistema** salva o histórico de mensagens.

Cenários Alternativos: Variantes

Variante 1: Apagar conversa

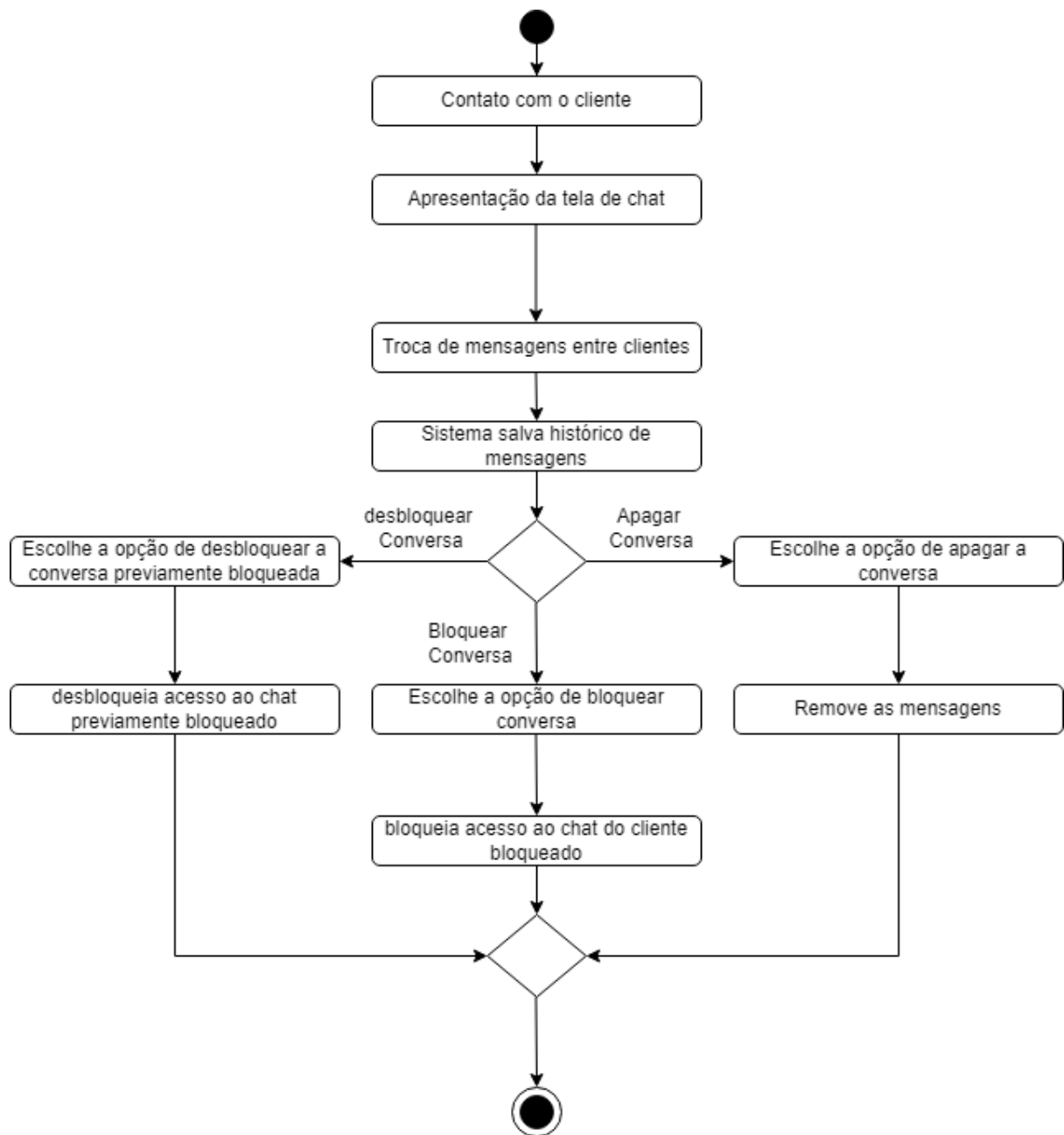
- O **cliente** escolhe a opção de apagar a conversa;
- O **sistema** remove as mensagens (apenas para o cliente solicitante).

Variante 2: Bloquear conversa com cliente

- O **cliente** escolhe a opção de bloquear a conversa;
- O **sistema** bloqueia o acesso ao chat com cliente bloqueado.

Variante 3: Desbloquear conversa com cliente

- O **cliente** escolhe a opção de desbloquear uma conversa previamente bloqueada;
- O **sistema** desbloqueia o acesso ao chat com cliente anteriormente bloqueado.



UC05 - Contribuição do Cliente

| | |
|----------------------|--|
| Atores | Cliente, API Pagamento, Amazon SQS |
| Interessados | Realizadores do Projeto |
| Pré-condições | Clientes cadastrados |
| Pós-condições | Emissão de Token de troca de Item |
| Visão Geral | Processo de troca entre Itens de vestuário de dois Cliente |

Cenário Principal de Sucesso

1. O **cliente** escolhe a opção de doação;
2. O **sistema** apresenta uma tela informando os Itens disponíveis;
3. O **cliente** preenche um formulário para filtragem e/ou preenche parcialmente o nome do Item desejado;
4. O **sistema** apresenta uma tela contendo os Itens solicitados;
5. O **cliente** escolhe a opção de Item que melhor lhe atende;
6. O **sistema** apresenta uma tela com informações detalhadas do Usuário que cadastrou o Item e do Item em si;
7. O **cliente** entra em contato com o Cliente que cadastrou o item e ambos iniciam a troca;
8. O **sistema** emite um Token para a troca de Item;
9. O **cliente** efetua e confirma a troca de itens;
10. O sistema valida o Token de troca de Item.

UC06 - Avaliação do Cliente

| | |
|----------------------|---|
| Atores | Cliente |
| Interessados | Clientes |
| Pré-condições | Clientes cadastrados, Clientes logados |
| Pós-condições | Ajuste de Avaliação do Cliente |
| Visão Geral | Processo de ganho ou perda de pontos após o recebimento de um item de outro usuário |

Cenário Principal de Sucesso

1. O **cliente** confirma o recebimento do pedido;
2. O **sistema** apresenta uma tela para a avaliação;
3. O **cliente** preenche um formulário com a quantidade de estrelas que deseja dar ao usuário que doou o item, podendo utilizar de observações;
4. O **cliente** confirma suas observações ;
5. O **sistema** envia a pontuação para o sistema e o mesmo faz a média com os pontos que o usuário avaliado já recebeu;

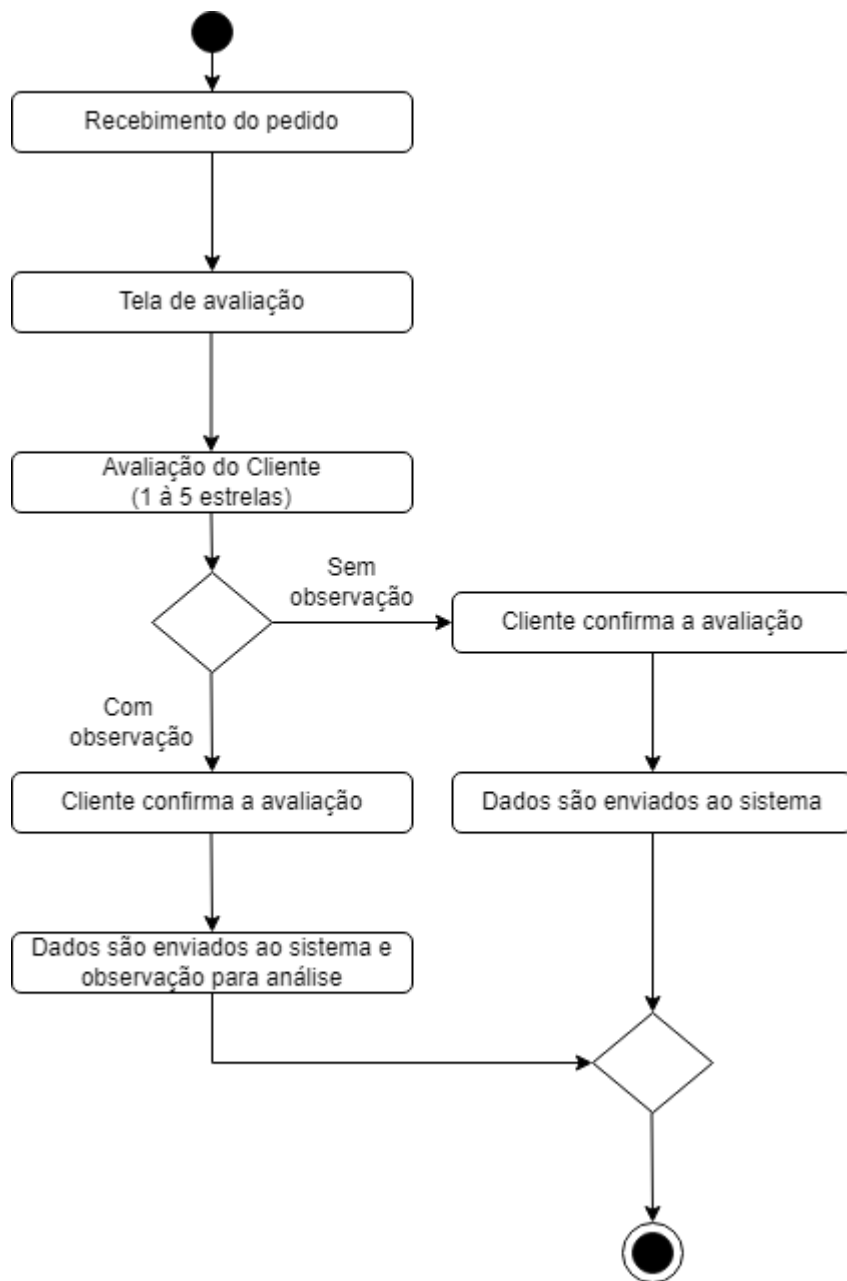


Diagrama de Classes

