deti universidade de aveiro
departamento de eletrónica, telecomunicações e informática

TQS: Product specification report

Dinis Cruz [93080], Duarte Mortágua [92963], José Sousa [93019], Tiago Oliveira [93456]

v2020-05-28

1 Introdução

1.1 Visão global do projeto

Este projeto tem como objetivo a implementação de um sistema de entregas de produtos em Spring. Damos uso a todas as tecnologias de teste de software que aprendemos na disciplina e testamos a nossa aplicação em diferentes níveis.

A nossa aplicação chama-se “CITO” que significa “rapidamente” em latim. Esta aplicação é baseada num sistema de entrega de produtos - em específico entrega de farmacêuticos. A nossa aplicação (como todas as outras aplicações que proporcionam o serviço de entregas ao domicílio - UberEats,Telepizza, etc…), tem o objetivo de poupar ao cliente a deslocação para comprar o que ele necessita em troca de uma taxa.

Após definirmos o tema, distribuímos papéis para cada membro da equipa:

- Team Manager: Duarte Mortágua
- Product Owner: Tiago Oliveira
- DevOps Master: José Sousa
- QA engineer: Dinis Cruz
- Developers: The whole team

1.2 Limitações

- Possibilidade do manager personalizar o serviço consoante a loja que ele registou, como por exemplo oferecer serviços diferentes.
- A plataforma (frontend) apenas suporta uma farmácia e por isso o cliente está limitado a encomendar apenas por uma farmácia. Apesar disto, o backend tem a lógica de multi-farmácias implementada.
- Lógica de utilizadores não implementada no frontend.
- Devido a erros do CORS, na client-web-app o checkout faz um post direto para o engine principal (cito-engine). Foram testadas diversas configurações de CORS em ambos os engines e até no frontend, mas falharam todas. Assumimos que mais vale ter a funcionalidade a funcionar “não tão bem” do que não estar a funcionar de todo. O resto dos pedidos da client-web-app passam todos pelo client-engine e são proxied normalmente para o cito-engine.

2 Conceito do Produto

2.1 Visão

A nossa aplicação serve para fazer a entrega de farmacêuticos ao domicílio. Os estafetas são responsáveis por ir buscar os farmacêuticos e entregá-los nas casas dos clientes. A interação com a interface é relativamente básica porque o cliente apenas tem de se registar, adicionar os produtos que quer

ao carrinho, encomendar e num certo espaço de tempo os produtos devem ser entregues em sua casa. Este tipo de aplicações como a UberEats e a Telepizza (e o CITO) que fazem entregas ao domicílio têm sempre o objetivo de resolver o problema da deslocação do cliente, seja por falta de tempo ou por outra razão qualquer.

2.2 Personas

Ana:

- A Ana é uma mulher de 27 anos que mora em Aveiro e estuda Engenharia Informática na Universidade de Aveiro.
- Ela dedica a maior parte do seu tempo aos estudos e mora longe do centro de Aveiro pois foi onde conseguiu arranjar um quarto mais barato. Ela deita-se tarde e acorda cedo por causa do seu horário preenchido, então costuma ter dores de cabeça no seu dia.
- Para não perder tempo a deslocar-se para a farmácia (que fica longe de sua casa), a Ana precisa de uma aplicação que lhe permita encomendar produtos e que estes sejam entregues em sua casa.

João:

- O João é um homem de 30 anos que mora em Coimbra. Trabalha por turnos como engenheiro de segurança numa empresa.
- O João, como trabalha por turnos, tem muito tempo livre e quer encontrar um part time que lhe dê algum dinheiro extra.
- Como o João tem uma mota económica e já alguns anos de experiência a conduzir ele gostava de ter um part time em algo relacionado com entregas ao domicílio.

Fernando:

- O Fernando tem 33 anos e mora no Porto com a sua mãe. Trabalha num escritório como gestor financeiro de uma empresa.
- O Fernando tem a sua mãe doente em casa sozinha. De vez em quando, ela sente-se enjoada e com dores de cabeça e precisa de algo para aliviar este mau-estar.
- Como o Fernando trabalha todo o dia e não tem possibilidade de ir a casa levar o medicamento que a mãe precisa, ele procura uma aplicação que faça entrega de medicamentos ao domicílio em que ele possa escolher o local de entrega (neste caso a sua casa para entregar à sua mãe).

Mariana:

- A Mariana tem 47 anos e mora no Porto. Ela é gestora de uma empresa de entrega de medicamentos.
- A Mariana gostava de aderir a uma plataforma em que pudesse registar a sua empresa e os seus produtos e para gerir encomendas por parte dos seus clientes.

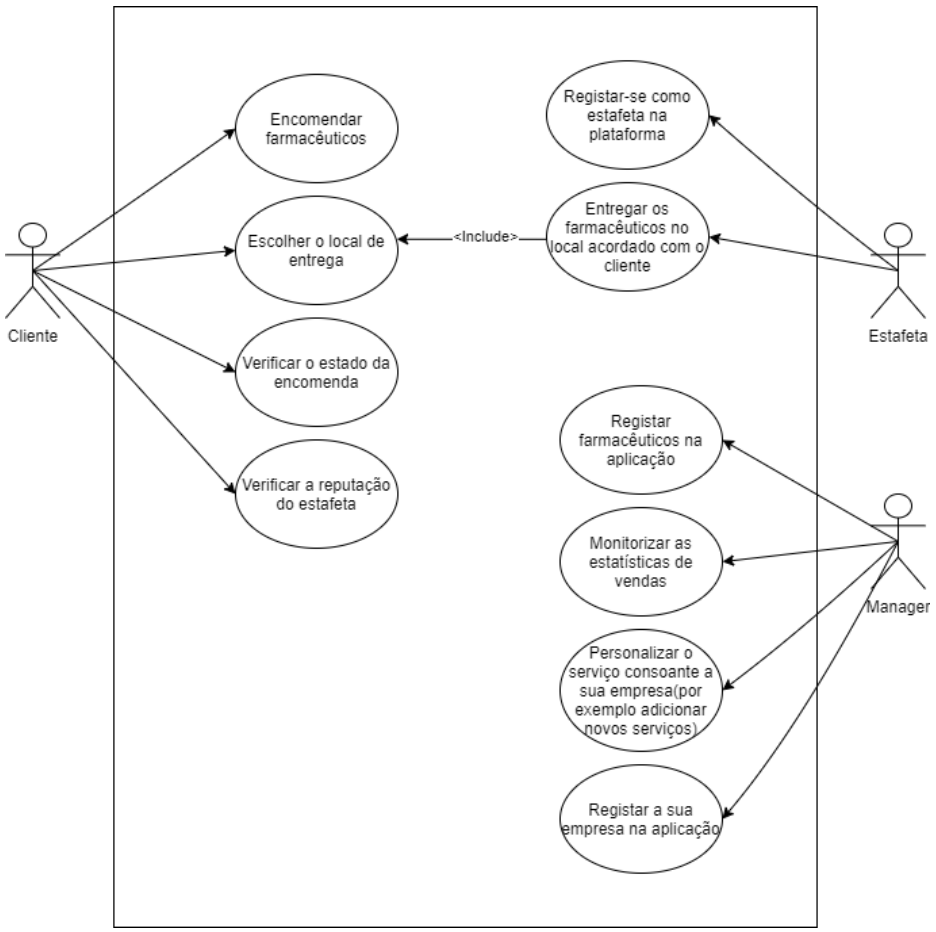

Fig. 1 : Diagrama de casos de uso

2.3 Cenários principais

Primeiro cenário:

A Ana quer comprar aspirinas para as suas dores de cabeça mas não quer ter de se deslocar até à farmácia. Se ela aceder à nossa aplicação e encomendar as aspirinas estas serão entregues em sua casa.

Segundo cenário:

O João quer ter um part time para fazer entregas ao domicílio. Se ele aceder à nossa aplicação pode registar-se como estafeta e começar a fazer entregas ao domicílio.

Terceiro cenário:

O Fernando quer comprar os medicamentos para a mãe e quer que eles sejam entregues em casa dela enquanto ele está no trabalho. Se ele aceder à nossa aplicação apenas precisa de selecionar os

medicamentos que quer, morada de entrega dos mesmos e dentro de minutos eles serão entregues na morada desejada.

Quarto cenário:

A Mariana quer registar a sua empresa numa aplicação já preparada para entregas de farmacêuticos ao domicílio. Se ela aceder à nossa aplicação, pode registar a sua empresa e rapidamente começar a fornecer entregas ao domicílio.

2.4 Project epics and priorities

1. **Iteração**
    - Definição do tema
    - Setup do repositório git e drive partilhada

2. **Iteração**
    - Definir a arquitetura
    - Prototipagem da interface
    - Definição das user stories a serem implementadas
    - Deployment da base de dados

3. **Iteração**
    - Implementação das user stories mais importantes :
        1. Como um consumer quero verificar o estado da encomenda
        2. Como um rider quero dar update ao estado da encomenda
        3. Como um manager quero registar um produto na plataforma
        4. Como um manager quero verificar todos os produtos na minha plataforma
        5. Como um consumer quero procurar produtos
        6. Como um consumer quero fazer uma encomenda
    - Implementação da CI pipeline
    - Desenvolvimento da API

4. **Iteração**
    - Implementação das restantes user stories:
        1. Como um manager quero registar a minha loja
        2. Como um rider quero-me registar como rider
        3. Como um manager quero me registar como manager
        4. Como um manager quero verificar qual o meu lucro até ao momento
        5. Como um consumer quero verificar a reputação do rider que me foi matched
        6. Como um consumer quero avaliar o serviço de entrega do rider que me foi matched
    - Implementação da CD pipeline
    - Elaboração do QA manual

5. Iteração

- Retoques no produto final
- Elaboração do product specification report

3 Modelo do Domínio

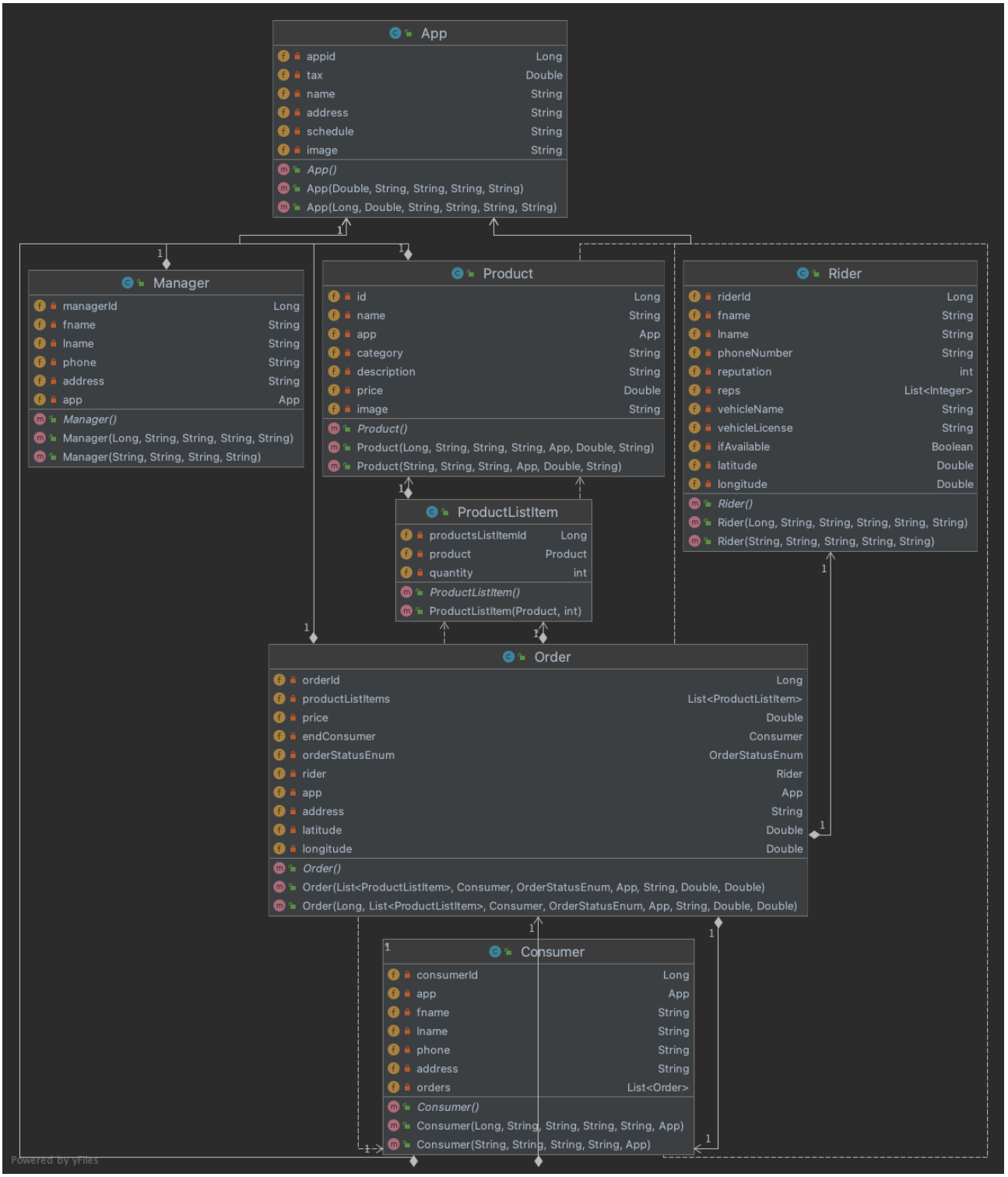

Fig.1 : Diagrama do modelo do domínio

4 Caderno de Arquitetura

4.1 Requisitos chave

Requisitos chave:

- Para aceder ou criar dados precisamos de uma API
- Para conseguirmos verificar se a cada nova user story implementada o que está para trás funciona precisamos de uma CI pipeline
- Para conseguirmos aceder à plataforma sem ser localmente precisamos de uma CD pipeline
- Caso o sistema vá abaixo para não perdermos o conteúdo da plataforma precisamos de uma base de dados

Requisitos chave que nos levaram á nossa arquitetura:

- Qual a melhor maneira de guardar os dados da plataforma? Que sistema de base de dados devemos utilizar?
- Migrar a plataforma para uma versão mobile ou não?

4.2 Arquitetura

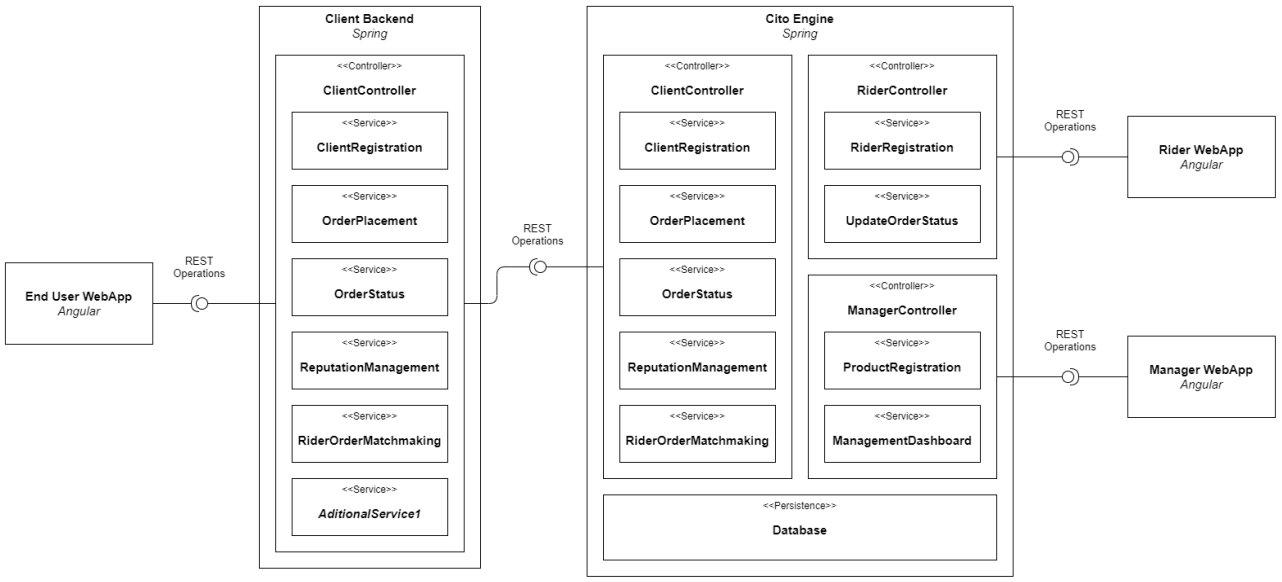

Fig. 2 : Diagrama de arquitetura

Há três web apps diferentes: Manager App, Rider App e Client App. O consumer utiliza o client- backend para fazer os seus pedidos e este client-backend consome a API que o cito-engine disponibiliza. Esta API é composta por todos os endpoints da plataforma e lida com os requests do client-backend, com os do

manager e com os do rider. O engine serve-se de uma base de dados onde estão persistidos todos os dados que não se podem perder (por exemplo, informações sobre os utilizadores e encomendas).

4.3 Deployment architecture

O deployment é feito no Heroku. O deploy é feito sempre que há merge no branch 'develop'. O Heroku dá listen de mudanças nestes branches e, quando detecta mudanças, espera que a pipeline de CI acabe e, desde que os jobs tenham passado com sucesso, as mudanças são integradas no Heroku.

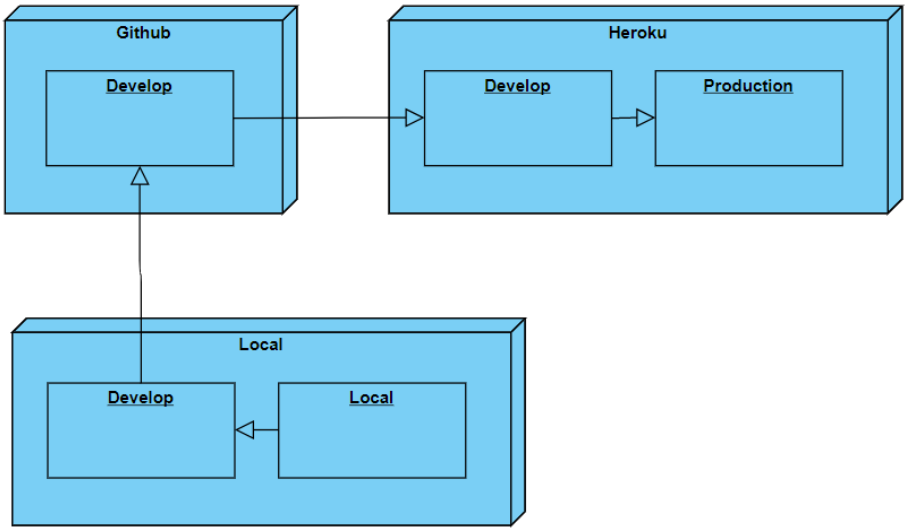

Fig. 3 : Diagrama de deployment

5 API for developers

Manager the Manager API

GET	`/managerApi/{managerId}/products`	Manager gets all the products.
POST	`/managerApi/{managerId}/products`	Manager registers a product.
POST	`/managerApi/{managerId}/app/register`	Register a app in the platform.
POST	`/managerApi/register`	Register a manager in the platform.
GET	`/managerApi/{managerId}/products/{productId}`	Manager gets specific product.
DELETE	`/managerApi/{managerId}/products/{productId}`	Deletes product by id.
GET	`/managerApi/{managerId}/info`	Retrieve manager info.
GET	`/managerApi/{appid}/statistics`	Manager gets app statistics.

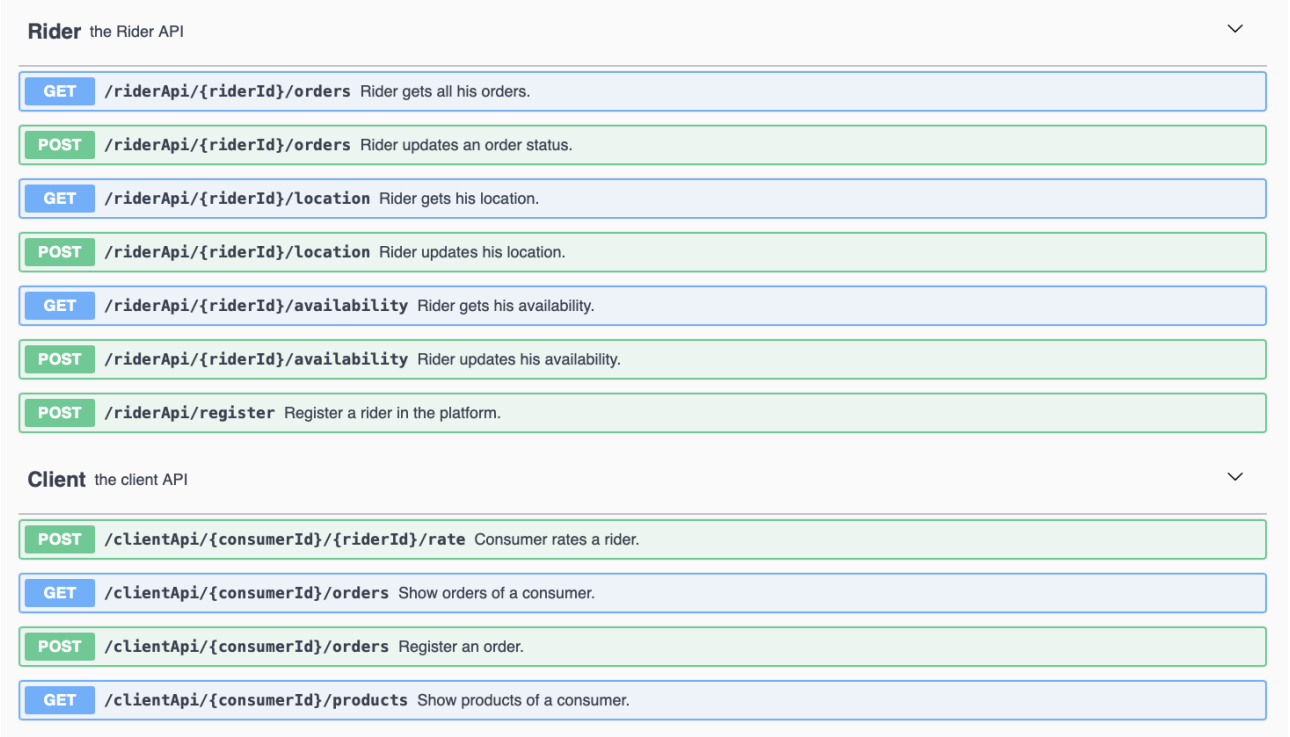

Fig. 4 : Documentação da API do engine

A figura 4 representa a documentação swagger da API do cito-engine. Cada endpoint tem uma descrição daquilo que ele fornece a cada tipo de utilizador.

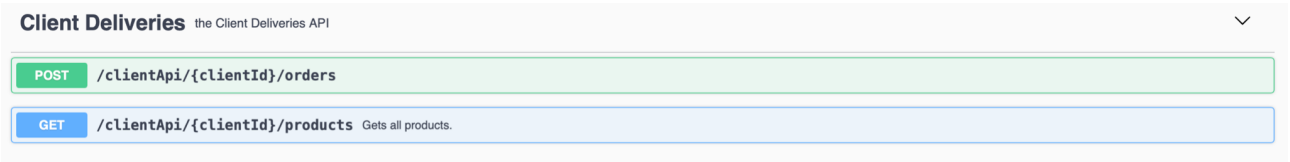

Fig. 5 : Documentação da API do Client

A figura 5 representa a documentação swagger da API do cito-client que fornece os endpoints do client, cada endpoint tem uma descrição daquilo que ele fornece.

6 References and resources

- Project repositories: https://github.com/92963-93080-93456-93019
- QA dashboard: https://sonarcloud.io/dashboard?id=92963-93080-93456-93019_cito-engine

- Engine API: https://cito-engine.herokuapp.com/swagger-ui.html#/
- Client API: https://cito-client.herokuapp.com/swagger-ui.html#/

- Engine deployment: https://cito-engine.herokuapp.com/
- Client deployment: https://cito-client.herokuapp.com/
- Client Web App deployment: https://cito-client-app.herokuapp.com/
- Rider Web App deployment: https://cito-rider-app.herokuapp.com/
- Rider Android App download.
- Manager Web App deployment: https://cito-manager-app.herokuapp.com/