|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Plataformas de Integração via API: conheça razões para você escolher a sua  - Startupi | | | | |
|  |  |  | |  |
| Serviços e Interoperabilidade de Sistemas  2021/2022  Cláudio Martins, 2190760; | | |
|  | |  |

Instituto Politécnico de Leiria

Torres Vedras

Curso Técnico Superior Profissional de Programação de Sistemas de Informação

Docente Mário Fernandes

Avaliação Periódica

10 de Janeiro de 2022

Cláudio Martins

*Índice*

[***Introdução*** 4](#_Toc92384213)

[*Ferramentas de Desenvolvimento:* 5](#_Toc92384214)

[Pontos de Desenvolvimento 6](#_Toc92384215)

[CRUD*:* 6](#_Toc92384216)

[Category: 6](#_Toc92384217)

[Consumo: 7](#_Toc92384218)

[Login: 8](#_Toc92384219)

[Products: 9](#_Toc92384220)

[Purchases: 10](#_Toc92384221)

[Signup: 11](#_Toc92384222)

[User: 12](#_Toc92384223)

[*Mosquitto:* 13](#_Toc92384224)

[Objetivo 13](#_Toc92384225)

[Ferramentas de Desenvolvimento: 13](#_Toc92384226)

[Interface Gráfica: 13](#_Toc92384227)

[Conclusão 14](#_Toc92384228)

# ***Introdução***

Em contexto da unidade curricular de Serviços e Interoperabilidade de Sistemas do 1º Semestre do 2º ano do Curso Técnico Superior Profissional de Sistemas de Informação do Instituto Politécnico de Leiria, foi elaborado um projeto em concordância com a cadeira de Acesso Móvel a Sistemas de Informação.

O tema é de livre escolha, ao qual decidimos assim apelidar o projeto de *SnackRestaurant*, que irá ter como alvo um café local escolhido pelo grupo, e que irá consistir numa plataforma em que será possível efetuar os pedidos na aplicação móvel, de forma a facilitar o desempenho do mesmo, ou seja, as filas para pedidos ao balcão irão ser reduzidas, tal como o tempo de espera da entrega à mesa dos mesmos pedidos.

Como tal, esta plataforma terá duas componentes, sendo elas uma REST API, que será desenvolvida na presente cadeira e uma aplicação móvel, que será desenvolvida na cadeira de Acesso Móvel a Sistemas de Informação.

A aplicação móvel será a interface que o cliente irá utilizar de modo a aceder ao menu no café e a consequentemente realizar os pedidos.

A REST API, por sua vez, irá fazer a conexão entre os dados com o sistema do café e a aplicação móvel, de modo a comunicar os pedidos, tal como sempre que necessário, atualizar em tempo real o menu do estabelecimento.

Nesta mesma REST API é necessário também realizar um sistema de messaging, em que o enquadramento na aplicação irá ser relativo aos pedidos dos clientes, receção dos mesmos por parte do estabelecimento, tal como também informar o cliente aquando do término da confeção do pedido.

*Enquadramento Teórico/Técnico*

## *Ferramentas de Desenvolvimento:*

A REST API foi desenvolvida com a utilização da Framework Yii2, basic, na qual utilizámos o IDE PhpStorm para o desenvolvimento do projeto.

Foi utilizado também o do GitHub para a partilha ativa da plataforma entre membros do grupo.

Por fim, utilizámos o navegador Chrome para a realização de testes e verificação da base de dados.

# Pontos de Desenvolvimento

# EndPoints*:*

## Category:

### Métodos de Implementação:

**GET**

**Utiliza-se este método de modo a obter uma lista completa das categorias que estejam registadas na plataforma.**

Curl

-H "Content-Type: application/json"

-X GET http://192.168.1.189:1884/v1/category?access-token=F\_Fu2do9PM8hdn0LCX4\_YPpTtDgsJIZi



**POST**

**O método POST tem a função de registar uma nova categoria na plataforma. Este pode apenas ser executado pelo administrador.**

Curl

-d “{\"name\": \”comida\”}”

-H "Content-Type: application/json"

-X POST http://192.168.1.189:1884/v1/category/post?access-token=F\_Fu2do9PM8hdn0LCX4\_YPpTtDgsJIZi



**PUT**

**O método PUT designa-se a alterar na plataforma, o registo da categoria e é exclusivamente executado pelo administrador.**

Curl

-d “{\"name\": \”comida\”}”

-H "Content-Type: application/json"

-X PUT http://192.168.1.189:1884/v1/category/put/1?access-token=F\_Fu2do9PM8hdn0LCX4\_YPpTtDgsJIZi



**DELETE**

**O método delete designa-se a eliminar na plataforma, o registo da categoria e é exclusivamente executado pelo administrador.**

Curl

-H "Content-Type: application/json"

-X DELETE <http://192.168.1.189:1884/v1/category/delete/1?access-token=F_Fu2do9PM8hdn0LCX4_YPpTtDgsJIZi>



## Consumo:

### Métodos de Implementação:

**GET**

**Utiliza-se este método de modo a obter uma lista completa do consumo que esteja registado na plataforma.** **Este pode apenas ser executado pelo administrador.**

Curl

-H "Content-Type: application/json"

-X GET http://192.168.1.189:1884/v1/consumo?access-token=F\_Fu2do9PM8hdn0LCX4\_YPpTtDgsJIZi



**POST**

**O método POST tem a função de registar um novo consumo na plataforma.**

Curl

-d “{\"id\_pedido\": 4,\”id\_product\”:1,\”quantidade\”:1}”

-H "Content-Type: application/json"

-X POST http://192.168.1.189:1884/v1/consumo/post?access-token=F\_Fu2do9PM8hdn0LCX4\_YPpTtDgsJIZi



**PUT**

**O método PUT designa-se a alterar na plataforma, o registo do consumo e é exclusivamente executado pelo administrador.**

Curl

-d “{\"id\_pedido\": 4,\”id\_product\”:1,\”quantidade\”:1}”

-H "Content-Type: application/json"

-X PUT http://192.168.1.189:1884/v1/consumo/put/1?access-token=F\_Fu2do9PM8hdn0LCX4\_YPpTtDgsJIZi



**DELETE**

**O método delete designa-se a eliminar na plataforma, o registo do consumo e é exclusivamente executado pelo administrador.**

Curl

-H "Content-Type: application/json"

-X DELETE http://192.168.1.189:1884/v1/consumo/delete/1?access-token=F\_Fu2do9PM8hdn0LCX4\_YPpTtDgsJIZi



**Custom:**

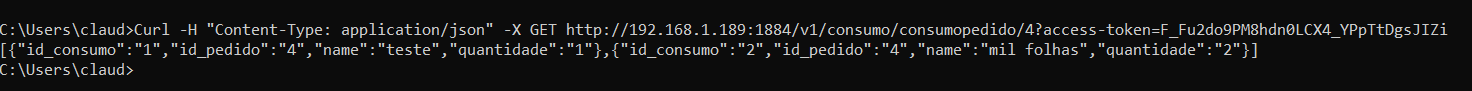
**GET**

**Utiliza-se este método de modo a obter uma lista completa do consumo do utilizador, que esteja registado na plataforma.**

Curl

-H "Content-Type: application/json"

-X GET <http://192.168.1.189:1884/v1/consumo/consumopedido/4?access-token=F_Fu2do9PM8hdn0LCX4_YPpTtDgsJIZi>



## Login:

### Métodos de Implementação:

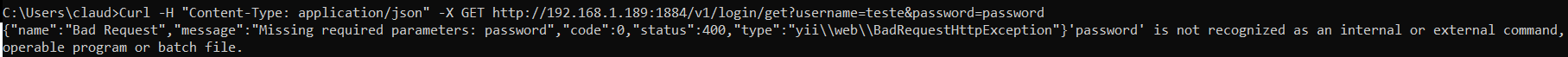
**Custom:**

**GET**

**Este método tem o objetivo de validar a autenticação dos utilizadores, e por sua vez apenas pode ser executado pelo utilizador indicado pelo Access Token.**

Curl

<http://192.168.1.189:1884/v1/login/get?username=teste&password=password>



## Products:

### Métodos de Implementação:

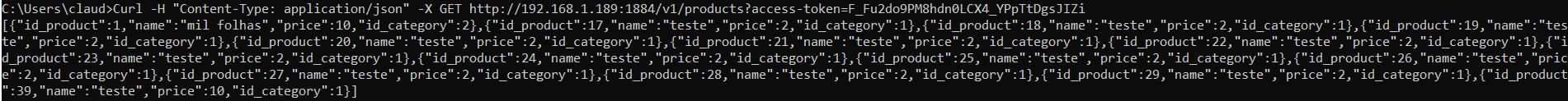
**GET**

**Utiliza-se este método de modo a obter uma lista completa dos produtos que estejam registado na plataforma.**

Curl

-H "Content-Type: application/json"

-X GET http://192.168.1.189:1884/v1/products?access-token=F\_Fu2do9PM8hdn0LCX4\_YPpTtDgsJIZi



**POST**

**O método POST tem a função de registar um novo produto na plataforma. Este pode apenas ser executado pelo admistrador.**

Curl

-d “{\"name\": \"teste\",\”price\”:10.80,\”id\_category\”:1}”

-H "Content-Type: application/json"

-X POST http://192.168.1.189:1884/v1/products/post?access-token=F\_Fu2do9PM8hdn0LCX4\_YPpTtDgsJIZi



**PUT**

**O método PUT designa-se a alterar na plataforma, o registo do produto e é exclusivamente executado pelo administrador.**

Curl

-d “{\"name\": \"teste\",\”price\”:10.80,\”id\_category\”:1}”

-H "Content-Type: application/json"

-X PUT http://192.168.1.189:1884/v1/products/put/1?access-token=F\_Fu2do9PM8hdn0LCX4\_YPpTtDgsJIZi



**DELETE**

**O método delete designa-se a eliminar na plataforma, o registo do produto e é exclusivamente executado pelo administrador.**

Curl

-H "Content-Type: application/json"

-X DELETE http://192.168.1.189:1884/v1/products/delete/1?access-token=F\_Fu2do9PM8hdn0LCX4\_YPpTtDgsJIZi



## Purchases:

### Métodos de Implementação:

**GET**

**Utiliza-se este método de modo a obter uma lista completa dos produtos que estejam registados na plataforma.**

Curl

-X GET http://192.168.1.189:1884/v1/purchases?access-token=F\_Fu2do9PM8hdn0LCX4\_YPpTtDgsJIZi



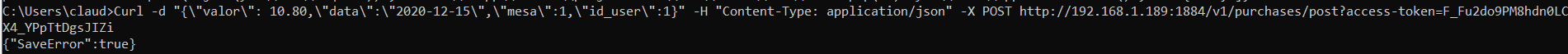
**POST**

**O método POST tem a função de registar um nova compra na plataforma.**

Curl

-d “{\"valor\": 10.80,\”data\”:\"2020-12-15\",\”mesa\”:1,\”id\_user\”:1}”

-X POST http://192.168.1.189:1884/v1/purchases/post?access-token=F\_Fu2do9PM8hdn0LCX4\_YPpTtDgsJIZi



**DELETE**

**O método delete designa-se a eliminar na plataforma, o registo da compra e é exclusivamente executado pelo administrador.**

Curl

-X DELETE <http://192.168.1.189:1884/v1/purchases/delete/1?access-token=F_Fu2do9PM8hdn0LCX4_YPpTtDgsJIZi>



**Custom:**

**GET**

**Utiliza-se este método de modo a obter uma lista completa das compras do utilizador, que esteja registado na plataforma.**

**Curl**

-X GET <http://192.168.1.189:1884/v1/purchases/purchasesuser/4?access-token=F_Fu2do9PM8hdn0LCX4_YPpTtDgsJIZi>



## Signup:

### Métodos de Implementação:

**Custom:**

**POST**

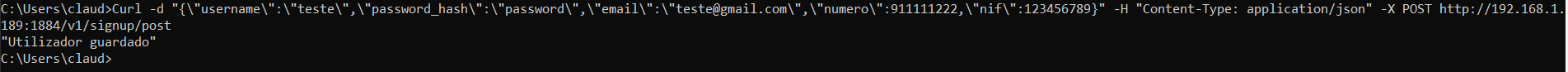
**Neste método não é obrigatório o Access Token e tem a finalidade de registar qualquer tipo de utilizador.**

Curl

-d “{\"username\":\”teste\”,\”password\_hash\”:\"password\",\”email\”:\"teste@gmail.com\",\”numero\”:911111222,\”nif\”:123456789}”

-H "Content-Type: application/json"

-X POST <http://192.168.1.189:1884/v1/signup/post>



## User:

### Métodos de Implementação:

**Custom:**

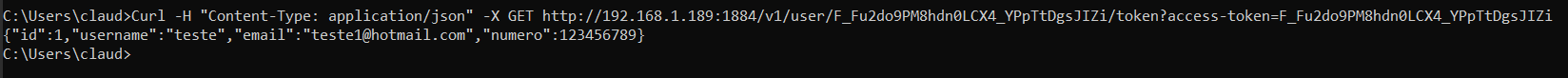
**GET**

**Utiliza-se este método de modo a obter os dados do utilizador que esteja registado na plataforma.**

Curl

-H "Content-Type: application/json"

-X GET <http://192.168.1.189:1884/v1/user/F_Fu2do9PM8hdn0LCX4_YPpTtDgsJIZi/token?access-token=F_Fu2do9PM8hdn0LCX4_YPpTtDgsJIZi>



**PUT**

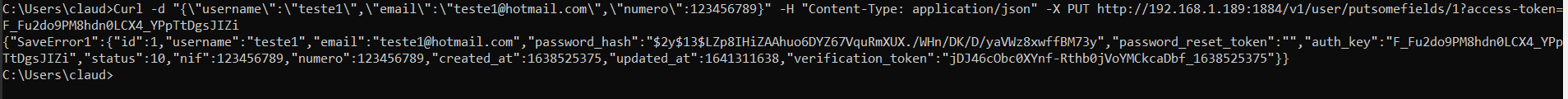
**Utiliza-se este método para alterar os dados de um utilizador que esteja registado na plataforma.**

Curl

-d “{\"username\":\”teste1\”,\”email\”:\”teste1@hotmail.com\”,\”numero\”:123456789}”

-H "Content-Type: application/json"

-X PUT <http://192.168.1.189:1884/v1/user/putsomefields/1?access-token=F_Fu2do9PM8hdn0LCX4_YPpTtDgsJIZi>



# *Mosquitto:*

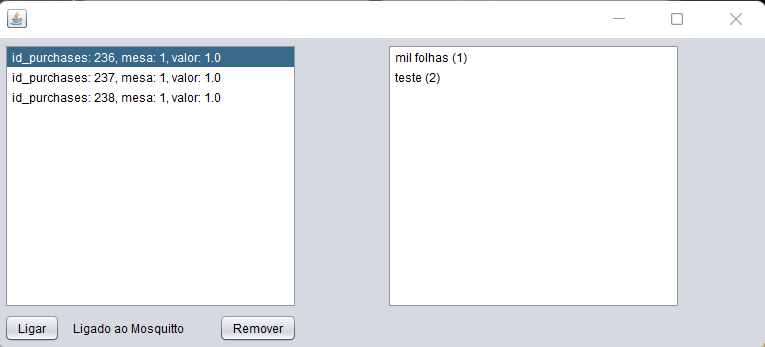
## Objetivo

O mosquitto tem como objetivo facilitar o colaborador, ao nível de gestão dos pedidos, ou seja, quando o cliente dá entrada do pedido na plataforma, é realizado o método publish para o Mosquitto de modo que o colaborador, na aplicação móvel, receba o pedido do cliente, através do método subscribe.

## Ferramentas de Desenvolvimento:

O Mosquitto foi desenvolvido em linguagem Java, no qual foi utilizado o IDE NetBeans para o desenvolvimento do mesmo.

## Interface Gráfica:



O botão Ligar serve para fazer conexão com o servidor do Mosquitto.

Por sua vez, o botão Remover tem a finalidade de remover o pedido da lista referente à Textbox localizada à esquerda.

Para se poder ver o consumo, ou seja, os produtos presentes no pedido, tem que ser selecionado o pedido a ser observado.

## Conclusão

Neste projeto consegui cumprir com os objetivos que tinha definido e com o que era pedido.

Senti algumas dificuldades, apesar delas consegui superar e avançar com o projeto.

Com o trabalho, aprendi algumas formas de uma boa api e um bom messaging.

Com este projeto concluído consegui realizar um trabalho de uma API.

Mesmo tendo tido algumas dificuldades ao longo do projeto, consegui ultrapassá-las e realizar um trabalho bastante razoável.