|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 5](#_Toc92561136)

[Enquadramento Teórico/Técnico 6](#_Toc92561137)

[Ferramentas de Desenvolvimento 6](#_Toc92561138)

[Pontos de Desenvolvimento - EndPoints 7](#_Toc92561139)

[Métodos de Implementação 7](#_Toc92561140)

[Category 7](#_Toc92561141)

[Consumo 9](#_Toc92561142)

[Login 11](#_Toc92561143)

[Products 12](#_Toc92561144)

[Purchases 14](#_Toc92561145)

[Signup 16](#_Toc92561146)

[User 17](#_Toc92561147)

[Mosquitto 18](#_Toc92561148)

[Objetivo 18](#_Toc92561149)

[Ferramentas de Desenvolvimento 18](#_Toc92561150)

[Aplicação na Prática 19](#_Toc92561151)

[Interface Gráfica 21](#_Toc92561152)

[Conclusão 22](#_Toc92561153)

**Índice de Figuras**

[Figura 1 - Category, Método Crud Get 7](#_Toc92561154)

[Figura 2 - Category, Método Crud Post 7](#_Toc92561155)

[Figura 3 - Category, Método Crud Put 8](#_Toc92561156)

[Figura 4 - Category, Método Crud Delete 8](#_Toc92561157)

[Figura 5 - Consumo, Método Crud Get 9](#_Toc92561158)

[Figura 6 - Consumo, Método Crud Post 9](#_Toc92561159)

[Figura 7 - Consumo, Método Crud Put 9](#_Toc92561160)

[Figura 8 - Consumo, Método Crud Delete 10](#_Toc92561161)

[Figura 9 - Consumo, Método Crud Get Custom 10](#_Toc92561162)

[Figura 10 - Login, Método Crud Get Custom 11](#_Toc92561163)

[Figura 11 - Products, Método Crud Get 12](#_Toc92561164)

[Figura 12 - Products, Método Crud Post 12](#_Toc92561165)

[Figura 13 - Products, Método Crud Put 13](#_Toc92561166)

[Figura 14 - Products, Método Crud Delete 13](#_Toc92561167)

[Figura 15 - Purchases, Método Crud Get 14](#_Toc92561168)

[Figura 16 - Purchases, Método Crud Post 14](#_Toc92561169)

[Figura 17 - Purchases, Método Crud Put 14](#_Toc92561170)

[Figura 18 - Purchases, Método Crud Delete 15](#_Toc92561171)

[Figura 19 - Purchases, Método Crud Get Custom 15](#_Toc92561172)

[Figura 20 - Signup, Método Crud Post Custom 16](#_Toc92561173)

[Figura 21 - User, Método Crud Get Custom 17](#_Toc92561174)

[Figura 22 - User, Método Crud Put Custom 17](#_Toc92561175)

[Figura 23 - Mosquitto, Função aftersafe 19](#_Toc92561176)

[Figura 24 - Mosquitto, FazPublish 20](#_Toc92561177)

[Figura 25 - Interface Gráfica Mosquitto 21](#_Toc92561178)

# 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**

## Métodos de Implementação

### Category

**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>



Figura 1 - Category, Método Crud Get

**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>



Figura 2 - Category, Método Crud Post

**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>



Figura 3 - Category, Método Crud Put

**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>



Figura 4 - Category, Método Crud Delete

### Consumo

**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>



Figura 5 - Consumo, Método Crud Get

**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>



Figura 6 - Consumo, Método Crud Post

**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>



Figura 7 - Consumo, Método Crud Put

**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>



Figura 8 - Consumo, Método Crud Delete

**GET - Custom**

**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>

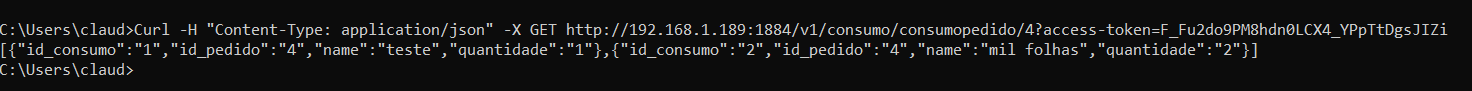


Figura 9 - Consumo, Método Crud Get Custom

### Login

**GET - Custom**

**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>

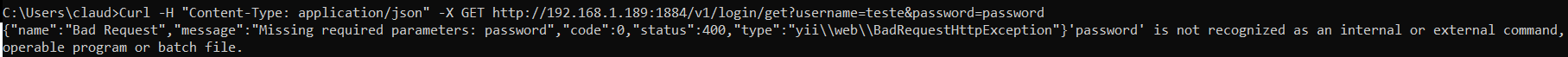


Figura 10 - Login, Método Crud Get Custom

### Products

**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>

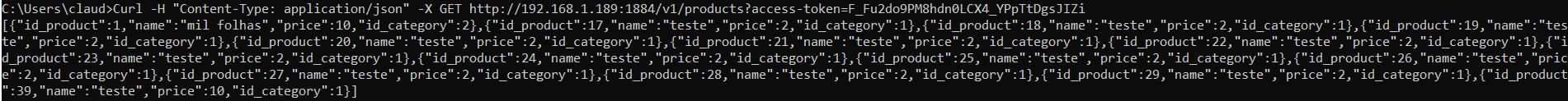


Figura 11 - Products, Método Crud Get

**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>



Figura 12 - Products, Método Crud Post

**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>



Figura 13 - Products, Método Crud Put

**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>



Figura 14 - Products, Método Crud Delete

### Purchases

**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>



Figura 15 - Purchases, Método Crud Get

**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>

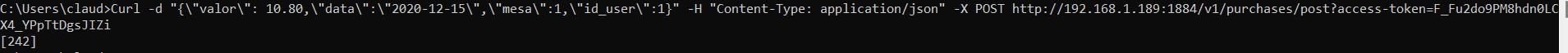


Figura 16 - Purchases, Método Crud Post

**PUT**

**O método PUT designa-se a alterar na plataforma, o registo da compra e é feito pelo administrador:**

**Curl -d “{\"valor\": 10.80,\”data\”:\"2020-12-15\",\”mesa\”:1,\”id\_user\”:1}” -H "Content-Type: application/json" -X PUT** <http://192.168.1.189:1884/v1/purchases/put/1?access-token=F_Fu2do9PM8hdn0LCX4_YPpTtDgsJIZi>

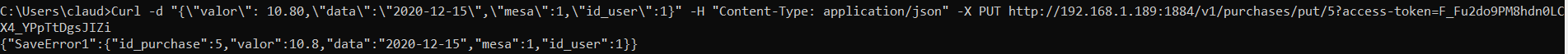
****

Figura 17 - Purchases, Método Crud Put

**DELETE**

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

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



Figura 18 - Purchases, Método Crud Delete

**GET - Custom**

**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>



Figura 19 - Purchases, Método Crud Get Custom

### Signup

**POST - Custom**

**Neste método não é obrigatório o Access Token e serve para 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>

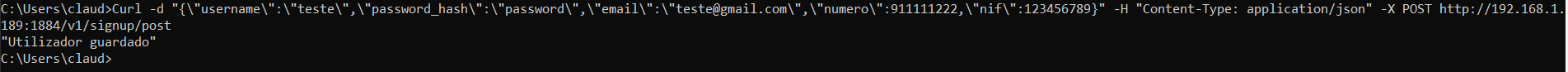


Figura 20 - Signup, Método Crud Post Custom

### User

**GET - Custom**

**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>

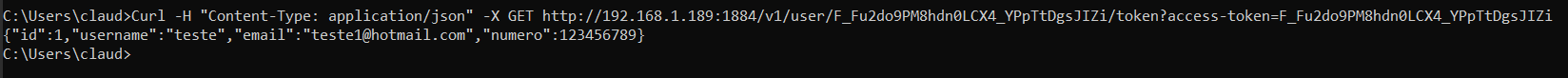


Figura 21 - User, Método Crud Get Custom

**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>

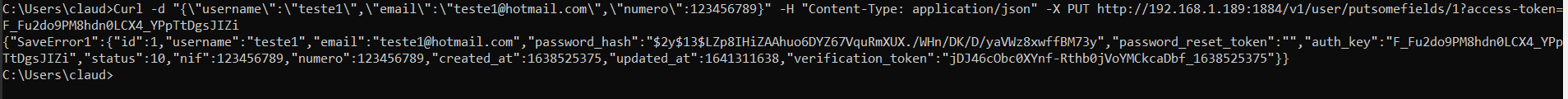


Figura 22 - User, Método Crud Put Custom

# 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.

## Aplicação na Prática

Em contexto do nosso projeto, a aplicação Mosquitto foi feita no modelo Purchases, através da função aftersave, de modo em que se faz a ligação da gravação do modelo do Purchases.

Como observamos na figura apresentada de seguida, o objetivo desta função é publicar a compra efetuada pelo cliente e de seguida mostrar a mesma ao colaborador.

Uma imagem com texto

Descrição gerada automaticamente

Figura 23 - Mosquitto, Função aftersafe

Após a função afterSave, temos a função fazPublish, que é o local onde se coloca o nome do canal e a respetiva mensagem. Posto isto, podemos verificar na figura abaixo a função fazpublish, que vai publicar a mensagem definida anteriormente, no canal referente ao mesmo.

Uma imagem com texto

Descrição gerada automaticamente

Figura 24 - Mosquitto, FazPublish

## Interface Gráfica

Como referido anteriormente, a interface gráfica foi desenvolvida em linguagem Java através do IDE NetBeans.

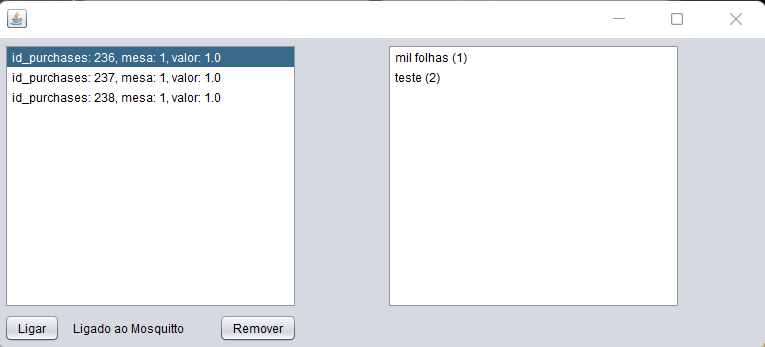


Figura 25 - Interface Gráfica Mosquitto

Posto isto na imagem acima podemos observar o desenvolvido no contexto do projeto.

Relativamente às funcionalidades temos o botão *Ligar*, que serve para fazer a 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

Com a elaboração deste projeto, cujo objetivo foi o desenvolvimento de uma API REST com fim a haver comunicação, consegui alcançar os requisitos propostos e os meus objetivos.

Apesar de algumas dificuldades iniciais, sinto que com o decorrer das aulas consegui trabalhar de acordo o esperado e por fim finalizar o projeto.

Consegui adquirir os conhecimentos suficientes para a criação de uma API REST e para a implementação do Mosquitto e consequentemente consolidei estes mesmos conhecimentos ao desenvolver o projeto.