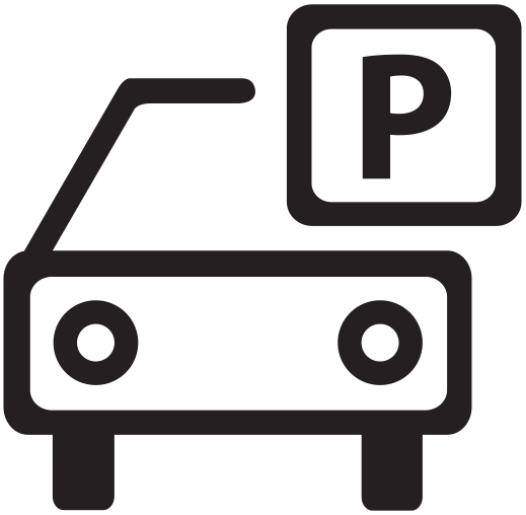


CITY-PARK-APP

UNIVERSIDADE LUSÓFONA DO PORTO

LUIS EMILIO NÚÑEZ MORALES - ALICIA SAMBADE MATA

3º ENGENHARIA INFORMÁTICA
PROGRAMAÇÃO DE SISTEMAS DISTRIBUIDOS

CITY-PARK-APP

1. Conteúdo

Neste documento está o trabalho realizado até o momento da primeira entrega (8 Maio 2022) pelo grupo 01 formado por Luis Morales e Alicia Mata.

No projeto está incluído um arquivo .JSON com a collection realizada em postman com os pedidos feitos, mas é acessível também no link a seguir: *https://www.getpostman.com/collections/98eb76a5435799ca00fa*

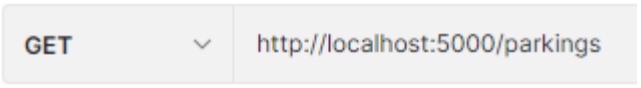
No relatório são incluídas as imagens dos testes feitos no postman para ser mostrado visualmente e ter melhor compreensão.

Quanto às funcionalidades implementadas, estão: mostrar todos os parkings da base de dados, criá-los e modificá-los e adicionar e apagar carros dos parkings.

Estas funcionalidades são empregadas nas duas interfaces implementadas: uma onde se mostram todos os parkings com a sua informação (necessário mostrar, criar e modificar parkings) e outra para que o cliente possa registrar a entrada e saída do seu carro do parking (necessário criar e apagar entradas de carros).

2. Funcionalidades

- **"/parkings"**
  - pedido **GET**, mostra a informação guardada de todos os parkings.

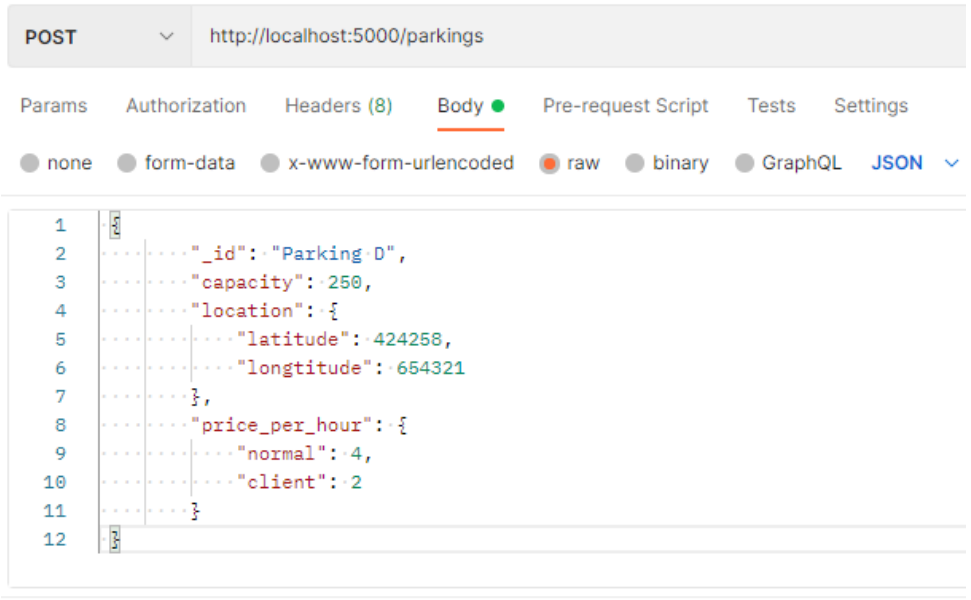


Pedido GET em Postman


```
"_id": "Parking A",
"capacity": 200,
"location": {
    "latitude": 123456,
    "longtitude": 654321,
    "_id": "6276892f27d27902df11b0f3"
},
"price_per_hour": {
    "normal": 5,
    "client": 2,
    "_id": "6276892f27d27902df11b0f4"
},
"cars_stored": [
    {
        "_id": "9687DWT",
        "entrance_date": "2022-05-07T17:01:03.000Z"
    }
],
```


Resultado em Postman do pedido GET em modo pretty (JSON)

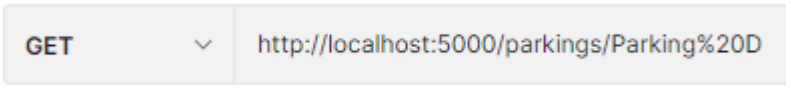
➢ pedido **POST**, adiciona um novo parking.



Pedido POST em Postman com o body para adicionar Parking D

❖ **"/parkings/:id"**

➢ pedido **GET**, mostra a informação guardada do parking com esse id.

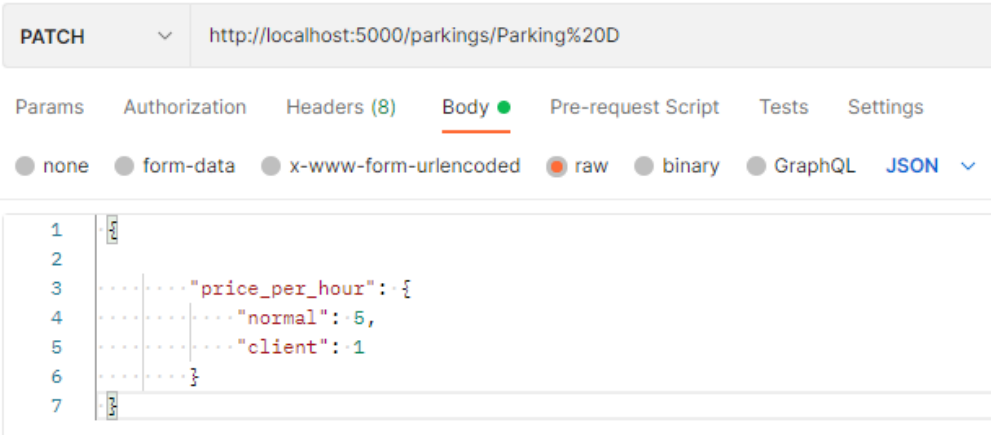


Pedido GET em Postman para Parking D


```
{
    "data": {
        "_id": "Parking D",
        "capacity": 250,
        "location": {
            "latitude": 424258,
            "longtitude": 654321,
            "_id": "6276ed42a7fc20a32189d6e6"
        },
        "price_per_hour": {
            "normal": 4,
            "client": 2,
            "_id": "6276ed42a7fc20a32189d6e7"
        },
        "cars_stored": [],
        "__v": 0
    }
}
```


Resultado em Postman do pedido GET em modo pretty (JSON)

➢ pedido **PATCH**, modifica os dados de um parking.

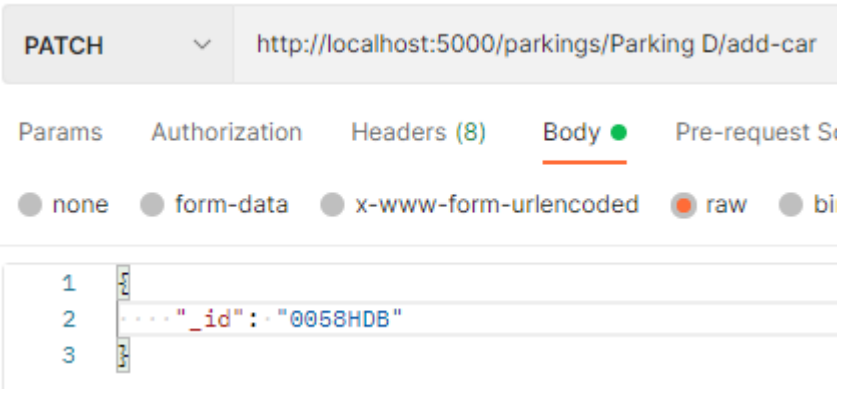


Pedido PATCH em Postman com o body para modificar o dado preço no Parking D

Resultado em Postman antes e depois da modificação com o pedido PATCH do preço

❖ **"parkings/:id/add-car"**

➢ pedido **PATCH**, para adicionar um novo carro (a entrada de um carro).



Pedido PATCH em Postman para registrar a entrada de um carro

```
{
    "data": {
        "_id": "Parking D",
        "capacity": 550,
        "location": {
            "latitude": 123654,
            "longtitude": 456321,
            "_id": "62781264bff854640c2a49de"
        },
        "price_per_hour": {
            "normal": 5,
            "client": 3,
            "_id": "6278138cbff854640c2a49e4"
        },
        "cars_stored": [
            {
                "_id": "1234ADB",
                "entrance_date": "2022-05-08T17:47:33.000Z"
            },
            {
                "_id": "0058HDB",
                "entrance_date": "2022-05-08T19:21:46.000Z"
            }
        ],
        "__v": 1
    }
}
```


Resultado em Postman antes e depois de adicionar um carro

- **"parkings/:id/remove-car/:car_id"**
  - ➢ pedido PATCH, para apagar um carro (a saída de um carro).

Pedido PATCH em Postman para registrar a saída de um carro


```
"_id": "Parking B",
"capacity": 800,
"location": {
    "latitude": 123654,
    "longtitude": 456321,
    "_id": "6278039bbff854640c2a4969"
},
"price_per_hour": {
    "normal": 5,
    "client": 1,
    "_id": "6278039bbff854640c2a496a"
},
"cars_stored": [
    {
        "_id": "808AAAA",
        "entrance_date": "2022-05-08T19:21:46.000Z"
    }
],
```


Resultado em Postman depois de adicionar um carro

3. Interfaces

Interface da rota "http://localhost:5000" [pedido GET]

Se se carrega no nome do parking, aparece a sua informação

Interface da rota "http://localhost:5000/entrada [pedido GET]

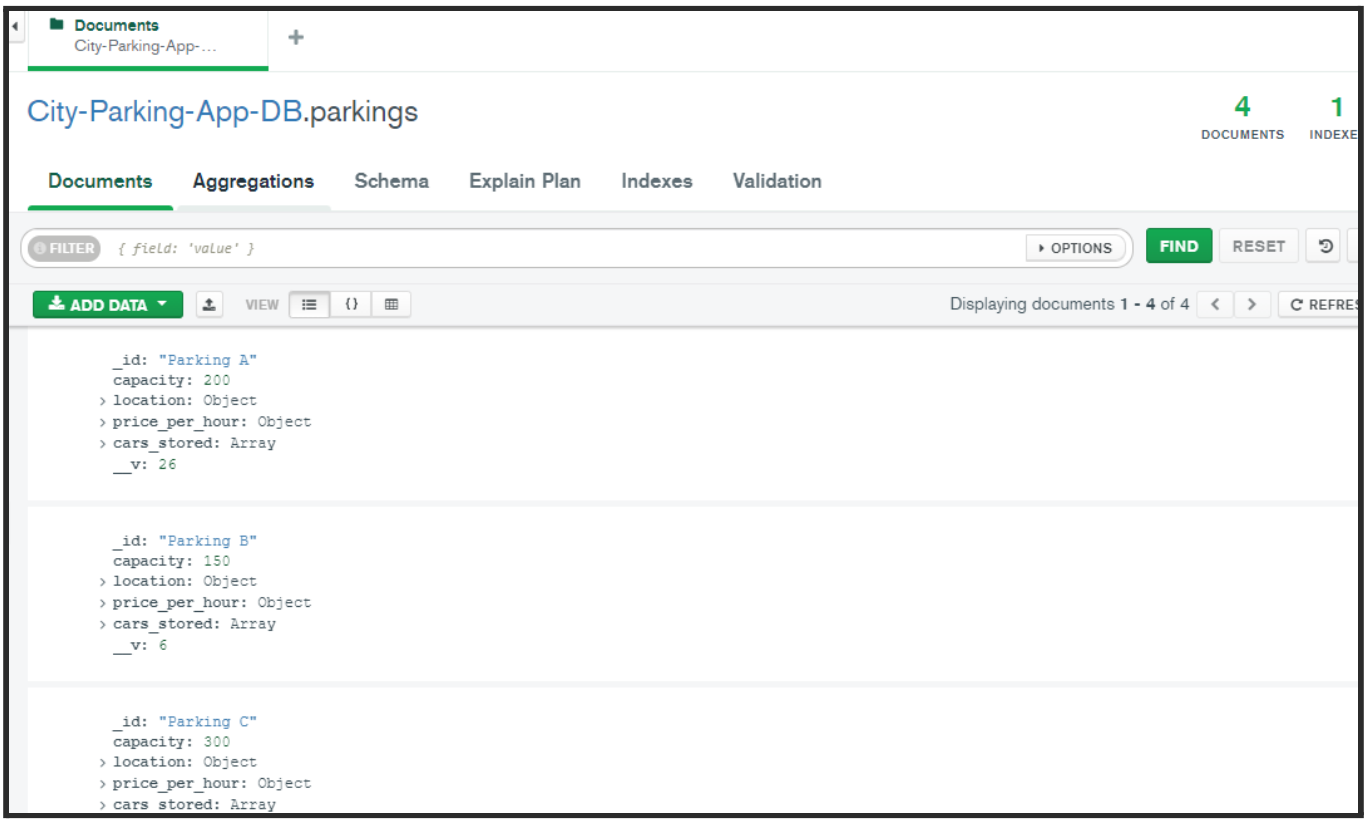
Nesta interface o cliente introduz os dados do estacionamento para registrar entradas e saídas.

Quando é registrada a entrada do carro, mostra-se um aviso ao cliente com a informação introduzida

Quando é registrada a saída do carro, mostra-se um aviso ao cliente com a informação do estacionamento

4. Mongodb

Como sistema de gestão de base de dados empregamos o Mongodb. A seguir, apresenta-se uma captura de ecrã com alguns dos elementos do nosso projeto guardados na base de dados.



O link para o acesso encontra-se no documento .env do projeto e a seguir:
mongodb+srv://invitado:invitado@cluster0.zofw9.mongodb.net/City-Parking-App-DB?retryWrites=true&w=majority