



Licenciatura em Engenharia Informática

Sistemas Distribuídos

Junho 2022

Trabalho Prático Sistemas Distribuídos

Grupo:

Tiago Godinho – a21803223

Tiago Ribeiro – a21805757

Joaquim Martins – a21803222

1. Introdução

Este trabalho tem como objetivo o desenvolvimento de uma aplicação distribuída utilizando a plataforma Node.js, permita funcionar como uma plataforma integradora que permita gerir a utilização de parques de estacionamento numa determinada cidade, tendo por base a disponibilização de uma API RESTful para garantir a integração entre a aplicação servidor e as aplicações que representam cada um dos tipos de aplicação cliente. A aplicação distribuída deverá incluir a utilização de pelo menos as tecnologias de comunicação estudadas ao longo das aulas: sockets, websockets e API de serviços web RESTful. É esperado que sejam desenvolvidos um conjunto de testes de interação com servidores desenvolvidos, quer através de aplicação de testes (Postman, SoapUI, etc), quer através de protótipos de aplicações cliente que facilitem a utilização dos serviços por parte do utilizador final. Além do código desenvolvido, no final deverá ser entregue um relatório acerca do trabalho realizado, e será realizada uma apresentação acerca do trabalho desenvolvido.

2. Arquitetura do sistema

O sistema a ser desenvolvido deverá permitir gerir a utilização dos parques de estacionamento numa determinada cidade, disponibilizando informações sobre os lugares disponíveis em cada parque, possibilitando o registo de entradas e saídas num determinado parque, e ainda o registo dos pagamentos da utilização dos parques. O sistema mantém ainda um conjunto de informação acerca dos clientes que pagam uma avença mensal para a utilização de um parque específico ou de qualquer parque registado no sistema. A aplicação terá ainda como models e routes, 'Entrada', 'Saída', 'Parques' e 'Preços', assim como no controller. Relativamente à base de dados esta será feita utilizando o 'MongoDB' e terá as collections 'EntradaCollection', 'SaidaCollection', 'ParquesCollection', 'PrecosCollection' e ainda 'ClienteCollection'.

3. Participação dos membros do grupo

No processo de desenvolvimento deste sistema foi estabelecida a seguinte divisão entre os membros do grupo:

Membro Tiago Ribeiro:

- Gestão entradas e saídas
- ParkAPI (1-3)
- Especificação OpenAPI de cada API desenvolvida (Swagger)

Membro Tiago Godinho:

- ParkAPI (4-6)
- APIGateway
- HTML

Membro Joaquim Martins:

- ClientAPI
- Pontos E1 e E2 do sistema entradas e saídas

Após o sistema desenvolvido ser dividido entre os 3 membros do grupo, foram encontrados alguns problemas, sendo que a participação dos 3 membros depois desses problemas foi a seguinte:

Membro Tiago Ribeiro:

- Gestão de Entradas e Saídas (Cliente-Servidor e Servidor-Cliente)
- Gestão dos parques (ParkAPI)
- Especificação OpenAPI de cada API desenvolvida (Swagger)
- Gestão dos Clientes (Operação indicar se cliente temavença válida)
- Postman
- Preçário do parque

Membro Tiago Godinho:

- ParkAPI
- APIGateway
- Postman

Membro Joaquim Martins:

- Gestão entradas (Pontos E1 e E2)
- Sistema de update do status do parque.
- Script tabelas de informação.
- Script filtragem informação.
- Script do validação parâmetros.
- ParkAPI
- ClientAPI
- HTML

4. Abordagem seguida

5. Objetivos atingidos

Durante o desenvolvimento do sistema foram estabelecidos certos objetivos para o sistema. De acordo com o desenvolvimento do projeto foi possível estabelecer os seguintes objetivos:

- Gestão entradas e saídas (Cliente-Servidor e Servidor-Cliente)

- Gestão dos parques (ParkAPI) pontos 1-4
- Criação de novos clientes e operação para verificar a sua avença
- Especificação OpenAPI de cada API desenvolvida (Swagger)
- Postman

6. Problemas encontrados

Durante o desenvolvimento do sistema, foram encontrados alguns problemas que dificultaram a resolução de todas as tarefas inicialmente estabelecidas. Alguns desses problemas foram:

- Má divisão de tarefas

Relativamente aos objetivos não resolvidos são estes:

- APIGateway
- Pontos E1 e E2 do sistema de entradas e saídas
- Pontos 4-7 do ParkAPI

7. Pontos a valorizar o trabalho

O desenvolvimento do sistema do grupo permite além de alguns dos pontos estabelecidos no enunciado sendo eles:

- Verificar os parques atualmente ativos.
- Verificar entradas e saídas dos parques atuais.
- Verificar preço do parque.
- Feature de filtragem dos parâmetros nas tabelas
- Design simples, responsivo.
- Verificar os parques atuais.
 - Ação de filtragem por parque.

8. Postman

Além da especificação OpenAPI de cada API desenvolvida (Swagger), foi também utilizado a ferramenta Postman, sendo que a documentação desta mesma é a seguinte:

<https://documenter.getpostman.com/view/20018264/Uz5GowAr>

Através da utilização desta ferramenta foi efetuada uma sequência dos pedidos no Postman sendo possível verificar os pedidos efetuados através do link situado em cima.

Alguns dos pedidos efetuados foram:

Sistema EECS:

Pedido 'Get' de todas as entradas, 'Post' de uma nova entrada e 'Delete' de uma determinada entrada.

Pedido 'Get' de todas as saídas e 'Post' de uma nova saída.

ParkAPI:

Pedido 'Get' dos atuais parques existentes, 'Post' de um novo parque e 'Get' dos parques atualmente ativos.

Preçario:

Pedido 'Get' dos atuais preços estabelecidos e 'Post' de um novo preço.

ClientAPI:

Pedido 'Get' dos atuais clientes registados e 'Post' de um novo cliente.

9. Webgrafia



<https://www.postman.com/>



<https://swagger.io/>

mongoose

elegant **mongodb** object modeling for **node.js**

<https://mongoosejs.com/>



mongoDB®

<https://www.mongodb.com/>