

ESCOLA SUPERIOR DE TECNOLOGIAS E GESTÃO

Ano Letivo 2023/2024

Curso Técnico Superior Profissional em: Tecnologias e Programção de Sist. Inf. 1º Ano/2 º Semestre

Unidade Curricular: Desenvolvimento Back-End Data Entrega: \_\_\_/\_\_\_/\_\_\_\_\_

Docente: David Jardim Época: \_\_\_\_\_\_\_\_

Relatório escrito do trabalho (Projeto 1)

API EM REST PARA UM SITE DE GESTÃO DE CONSULTAS MÉDICAS

[CTESP em Tecnologias e Programação de Sistemas de Informação](https://www.uma.pt/ensino/ctesp/ctesp-em-tecnologias-e-programacao-de-sistemas-de-informacao)

**Projeto criado em sql e javascript usando Postman, Visual Studio Code, phpMyAdmin + XAMPP e Mysql Worbench**

Índice

[**Introdução** 3](#_Toc164034703)

[**Objetivos Gerais e Endpoints:** 3](#_Toc164034704)

[**Implementação da Base de Dados em MySQL:** 5](#_Toc164034705)

[**Implementação do Servidor Express.js** 6](#_Toc164034706)

[**Linguagem de Programação JavaScript** 8](#_Toc164034707)

[**Conclusão** 9](#_Toc164034708)

# **Introdução**

No âmbito da unidade curricular de Desenvolvimento Back-End no curso Técnico Superior Profissional em Tecnologias e Programação de Sistemas de Informação foi proposto um protejo sobre exercícios de programação destinados a praticar o desenvolvimento de uma API com Node.js e MySQL.

## **Objetivos Gerais e Endpoints:**

**Listar todas as consultas existentes na tabela:**

* + Método HTTP: GET
  + Endpoint: /appointments
  + Descrição: Retorna todas as consultas presentes na tabela.

**Adicionar uma nova consulta à base de dados:**

* + Método HTTP: POST
  + Endpoint: / appointments
  + Descrição: Adiciona uma nova consulta ao banco de dados e retorna o ID da consulta adicionada.

**Selecionar todos as consultas de um determinado médico**

* + Método HTTP: GET
  + Endpoint: /appointments/doctor\_id/:doctor\_id
  + Descrição: Retorna todas as consultas associadas a um médico específico.

**Modificar o horário da consulta:**

* + Método HTTP: PUT
  + Endpoint: /appointments/start\_time/:id
  + Descrição: Modifica o horário de uma consulta, somando ou subtraindo uma determinada quantidade de tempo, e retorna a entrada atualizada.

**Listar todas as consultas anteriores a uma hora:**

* + Método HTTP: GET
  + Endpoint: /appointments/Hanterior
  + Descrição: Retorna todas as consultas que ocorreram antes de uma hora específica.

**Selecionar apenas uma consulta pelo seu ID:**

* + Método HTTP: GET
  + Endpoint: /appointments/:id
  + Descrição: Retorna uma consulta específica com base no seu ID.

**Apagar uma consulta existente:**

* + Método HTTP: DELETE
  + Endpoint: /appointments/:id
  + Descrição: Apaga uma consulta existente com base no seu ID e retorna mensagem de erro em caso de falha ou mensagem de sucesso se a consulta for apagada com sucesso.

**Selecionar todas as consultas de uma determinada clínica:**

* + Método HTTP: GET
  + Endpoint: / appointments /clinic/:clinic
  + Descrição: Retorna todas as consultas associadas a uma clínica específica.

**Adicionar um comentário a uma determinada consulta:**

* + Método HTTP: PUT
  + Endpoint: / appointments /:id/comments
  + Descrição: Adiciona um comentário a uma consulta específica, mantendo os comentários anteriores, e retorna a entrada atualizada.

**Listar todas as consultas ordenadas por ordem crescente de duração:**

* + Método HTTP: GET
  + Endpoint: / appointments /ordered
  + Descrição: Retorna todas as consultas ordenadas por ordem crescente de duração, com a ordenação realizada em JavaScript.

## **Implementação da Base de Dados em MySQL:**

Foi criado a tabela Appointments conforme especificado no enunciado, incluindo a definição das colunas e a inserção manual de algumas entradas.

Uma imagem com texto, captura de ecrã, software

Descrição gerada automaticamente

## **Implementação do Servidor Express.js**

Servidor em Express.js foi configurado e iniciado, incluindo a importação do módulo **mysql** para acesso ao banco de dados MySQL em Node.js.



**“O nosso cronograma”**

1º - Organização de ideias

2º - Distribuição de Tarefas

3º - Criar Tabela ‘ appointments ‘ MySQL

4º - Fazer os endpoints da Parte A primeiro e testá-los no Postman

5º - Fazer os endpoints da Parte B e testá-los no Postaman

6º - Acrescentar alguns comentários e ideias que surgiram

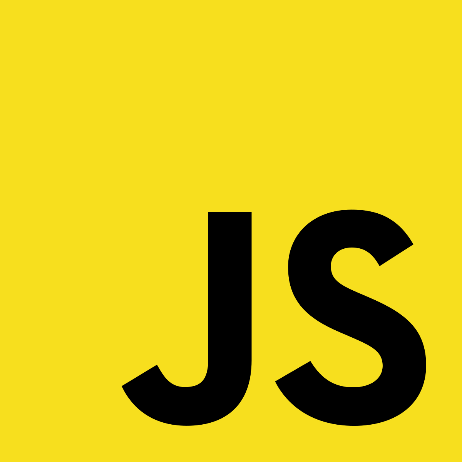
7º - Voltar a testar

8º - Relatório

## **Linguagem de Programação JavaScript**

JavaScript é uma linguagem de programação popular para criar aplicações web interativas. Ela foi originalmente desenvolvida para ser uma linguagem de script do lado do cliente, ou seja, para uso no navegador do usuário, mas agora também é amplamente utilizada no desenvolvimento de aplicativos do lado do servidor e em outros ambientes.

Uma imagem com texto, captura de ecrã, Tipo de letra

Descrição gerada automaticamente

## **Conclusão**

Apesar dos contratempos que enfrentamos com o programa MySQL inicialmente planeado, levando-nos a mudar para o PHPMyAdmin, a conclusão deste projeto evidencia a integração bem-sucedida entre uma base de dados MySQL e um servidor em Express.js. A nossa capacidade de adaptação e resolução de problemas ficou patente durante esta transição. Foi criado um sistema de gestão de consultas médicas funcional e fiável através da implementação cuidadosa dos endpoints REST e do tratamento de erros. A colaboração entre os parceiros revelou-se fundamental para alcançar os objetivos do projeto e superar os obstáculos.