

**Curso Técnico Superior Profissional em:** Tecnologias e Programação de Sistemas de Informação

**1.º Ano/2.º Semestre**

**Unidade Curricular:** Desenvolvimento Web - Back-End

**Data Entrega:** 30/06/2020

**Docente:** David Jardim

**Época:** Normal

---

## RECURSO

---

### Entrega

Entrega da resolução: através da plataforma moodle, na página da disciplina.

Data Limite de Entrega: 30 de junho de 2019, até às 19h30.

Nomenclatura da pasta com ficheiros: a pasta comprimida (zipada) deverá ter a seguinte estrutura quanto ao seu nome **FREQ\_numero\_aluno**

### PARTE 1

1. Implementação de um servidor em Express.js respeitando a arquitetura REST e o padrão de desenho MVC utilizando o ficheiro de configuração **.env** (1 valor)
2. Utilize o ficheiro disponibilizado com o enunciado para importar o schema **“world”** para o seu servidor de MySQL. Efetue a ligação à base de dados utilizando o ORM *Sequelize* criando os modelos necessários (2 valores).
3. Tendo em conta a tabela 1, implemente os seguintes *endpoints* no servidor:
  - a. Listar todos os países existentes na tabela **countries** e devolver a resposta no *body* (1 valor).
  - b. Adicionar um novo país à tabela **countries**. O código do país adicionado deve ser devolvido na resposta (2 valores).
  - c. Apagar um país da tabela **countries** pelo seu código. O número de linhas afetadas deve ser devolvido na resposta. Caso o país a apagar não exista o erro deverá ser tratado de forma adequada (2 valores).
  - d. Selecionar apenas um país pelo seu código e devolver esse mesmo país na resposta. Caso o país a selecionar não exista, o erro deverá ser tratado de forma adequada (2 valores).
  - e. Selecionar uma determinada cidade referente a um determinado país. Caso não exista, o erro deverá ser tratado de forma adequada (2 valores).

URI	Método HTTP	Body do POST	Resultado
/countries	GET	empty	Show list of all the countries.
/countries	PUT	JSON String	Add details of new country.
/countries/:id	DELETE	empty	Delete an existing country.
/countries/:id	GET	empty	Show details of a country.
/countries/:code/cities/:id	GET	empty	Show details of a single city from a single country.
/log	GET	empty	Download log file

Tabela 1 - Lista de endpoints

4. Implemente uma função de *logging* como *middleware* para registar num ficheiro **log.txt** o número de vezes que um determinado endpoint é acedido. Formato: **URI, Método HTTP, CONTAGEM**. (3 valores).
5. Implemente um endpoint que permita efetuar download do ficheiro de log em formato comprimido **log.txt.gz** utilizando o módulo *zlib* (2 valores).

## PARTE 2

1. Crie um ficheiro denominado *algorithm.js*, nesse ficheiro implemente uma função cujo objetivo é rodar o array **K** vezes para a direita e devolver um novo array com a soma de cada elemento. Tenha em conta o seguinte exemplo:

```
var array = [[1,1], [2,3], [3,3], [5,5,5], [5,4]];

var K = 3;
```

Roda o array original da seguinte forma:

```
var rotatedArray = [[3,3], [5,5,5], [5,4], [1,1], [2,3]];
```

E devolve um novo array contendo as somas de cada elemento (3 valores):

```
var summedArray = [6, 15, 9, 2, 5];
```

Column	Type	Default Value	Nulla...
Capital	int		YES
Code	char(3)		NO
Code2	char(2)		NO
Name	char(52)		NO

Figura 1 - Tabela *countries*

Column	Type	Default Value	Nulla...
CountryCode	char(3)		NO
District	char(20)		NO
ID	int		NO
Info	json		YES
Name	char(35)		NO

Cofinanciado por:

Figura 2 - Tabela *cities*

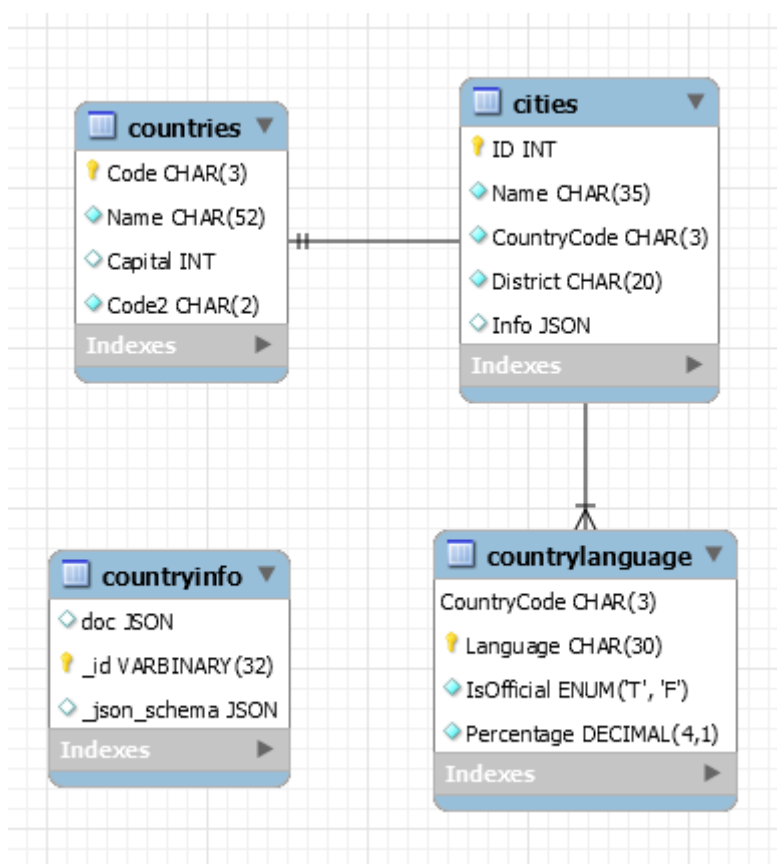


Figura 3 - Schema da Base de Dados

Cofinanciado por: