

# Guião para a terceira semana de aulas (RPCW2023)

## Gestão de indivíduos

Tarefas a realizar na aula:

1. Explicar e instalar o json-server:
  1. Breve introdução ao npm;
  2. Instalar o json-server;
  3. Explicar como funciona;
  4. Explicar a estrutura que uma BD em JSON tem de ter.
2. Colocar o json-server a servir uma das BDs de indivíduos:
  1. Usar o dataset-extra1.json por ser mais pequeno;
  2. `json-server --watch dataset-extra1.json`
  3. Demonstrar no browser o seu funcionamento:
    1. Listar os indivíduos; `http://localhost:3000/pessoas`
    2. Listar apenas um indivíduo (a importância de haver um campo id). Não havendo id, é preciso usar um campo candidato a chave como filtro, neste caso, CC ou BI: `http://localhost:3000/pessoas?CC=48386914-0-BU9`
    3. Filtrar a lista por uma das características: `http://localhost:3000/pessoas?sexo=feminino`
    4. Ordenar a lista: `http://localhost:3000/pessoas?_sort=nome&_order=asc`
    5. Filtrar por campos internos: `http://localhost:3000/pessoas?morada.cidade=Braga`
    6. Filtrar pela posição, do X ao Y: `http://localhost:3000/pessoas?_start=20&_end=30`
    7. Full Text Search: `http://localhost:3000/pessoas?q=Paddle`
    8. Acrescentar um registo à BD (usar o Postman):
      - Configurar o header: `application/json`
      - Configurar o body: `raw`
      - Colocar o JSON no body:

```
{
  "id": "RPCW2023",
  "nome": "Representação e Processamento de Conhecimento na Web",
  "idade": 3,
  "cidade": "Braga"
}
```

9. Confirmar a existência do registo na BD: `http://localhost:3000/pessoas/RPCW2023`
3. Criar um serviço de navegação sobre a BD:
  1. Instalar e explicar o módulo axios;
  2. Criar a primeira versão do serviço: vai buscar a lista de indivíduos e diz quantos recuperou;

```
const axios = require('axios');

axios.get('http://localhost:3000/pessoas')
  .then(function (resp) {
    pessoas = resp.data
    console.log(typeof(pessoas))
    console.dir(pessoas[3].nome)
    console.log("Recuperei " + pessoas.length + " registos.")
  })
  .catch(function (error) {
    console.log(error);
  });
```

3. A seguir, vai buscar um determinado registo e apresenta o nome da pessoa;
4. Sem o id não é prático, criar uma script em Python para acrescentar um id a cada registo;

```
import json

f = open("dataset-extra1.json")
pessoas = json.load(f)
f.close()

for index, p in enumerate(pessoas["pessoas"]):
    p['id'] = "p" + str(index)

f = open("dataset-extra1.json", "w")
json.dump(pessoas,f)
f.close()
print("Adicionados " + str(index) + " identificadores...")
```

5. Acrescentar ao `get_request` o pedido de recuperar a informação do indivíduo "p999";

6. Criar a segunda versão do serviço: vai buscar a lista de indivíduos e apresenta numa tabela HTML uma projeção sobre 4 ou 5 campos: id, nome, idade, sexo, cidade;

```
// pessoas_server.js
// RPCW2023: 2023-02-25
// by jcr

var http = require('http')
var url = require('url')
var mypages = require('./mypages')
var axios = require('axios')

http.createServer(function (req, res) {
  var dicURL = url.parse(req.url, true)

  if(dicURL.pathname == "/"){
    axios.get('http://localhost:3000/pessoas')
      .then(function (resp) {
        var pessoas = resp.data
        res.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'})
        res.end(mypages.genMainPage(pessoas))
      })
      .catch(function (error) {
        console.log(error);
        res.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'})
        res.end("Erro: " + error)
      });
  }
  else if(dicURL.pathname == "/ordenada"){
    axios.get('http://localhost:3000/pessoas')
      .then(function (resp) {
        var pessoas = resp.data
        let pessoasOrdenadas = pessoas.sort(
          (p1, p2) => (p1.nome < p2.nome) ? -1 : 1);
        res.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'})
        res.end(mypages.genMainPage(pessoasOrdenadas))
      })
      .catch(function (error) {
        console.log(error);
        res.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'})
        res.end("Erro: " + error)
      });
  }
  else{
    res.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'})
    res.end("Erro: Operação não suportada...")
  }
}).listen(7777)

console.log("Servidor à escuta na porta 7777...")
```

6. Acrescente uma nova rota onde se obtem a mesma página com as pessoas ordenadas alfabeticamente por nome.

4. Adicionar CSS à geração da página:

1. Introdução à framework w3.css;
2. Como servir recursos estáticos.

5. Criar a terceira versão do serviço: um clique numa linha da tabela resulta num pedido ao serviço que irá servir a página do indivíduo com toda a sua informação, nessa página haverá um link para voltar a página inicial.
6. Criar uma script JS que faz um POST na BD de um indivíduo.
7. Criar uma nova script JS que lê um dos outros datasets e faz POST de todos os registos para a BD.
8. Discutir a integração destas novas funcionalidades no serviço inicial.