

## Bootcamp: Desenvolvedor(a) Node.js

### Trabalho Prático

#### Módulo 1: Desenvolvimento Back End com JavaScript

#### Objetivos de Ensino

Exercitar os seguintes conceitos trabalhados no Módulo:

1. Criação de um projeto Node.js.
2. Manipulação de arquivos.
3. Manipulação de objetos JSON.
4. Criação de endpoints com Express.

#### Enunciado

O aluno deverá criar um projeto Node.js para realizar a criação de alguns métodos e processamento de arquivos JSON. Além disso, também será preciso criar alguns endpoints utilizando o Express, que irão chamar os métodos criados.

#### Atividades

O aluno deverá utilizar o arquivo “car-list.json”, fornecido junto com este enunciado. Esse arquivo foi obtido do seguinte link (<https://github.com/matthlavacka/car-list>). O aluno deverá utilizar o arquivo fornecido, e não realizar um novo download do link, evitando assim que possíveis alterações no repositório impactem nas respostas corretas do trabalho. O aluno

também não deve alterar o arquivo JSON, evitando divergências nas respostas corretas do trabalho.

Esse arquivo JSON consiste em uma lista de marcas de carros (brand) e seus modelos (models). O arquivo tem em sua raiz um array de marcas e dentro de cada marca há um array de modelos daquela marca.

Os alunos deverão desempenhar as seguintes atividades:

1. Criar uma função que retorne o nome da marca que mais possui modelos.
2. Criar uma função que retorne o nome da marca que menos possui modelos.

Observação: para os itens 1 e 2, em caso de empate em número de modelos, retornar uma lista com o nome das marcas que empataram. Exemplo de retorno em caso de empate: ["Marca 1", "Marca 2"]. Exemplo de retorno caso não tenha empate: "Marca 1".

3. Criar uma função que receba como parâmetro um número X e retorne as X marcas que mais possuem modelos, seguidos da quantidade, em ordem decrescente. Exemplo de retorno caso o parâmetro informado seja 5:

["Marca A - 10", "Marca B - 9", "Marca C - 8", "Marca D - 7", "Marca D - 6"].

4. Criar uma função que receba como parâmetro um número X, retorne as X marcas que menos possuem modelos, seguidos da quantidade, em ordem crescente. Exemplo de retorno caso o parâmetro informado seja 5:

["Marca E - 1", "Marca F - 2", "Marca G - 3", "Marca H - 3", "Marca I - 4"].

Observação: para os itens 3 e 4, em caso de empate, você deve considerar a ordem alfabética do nome das marcas como critério de desempate. Exemplo, caso as marcas "Audi" e "Renault" empatem, a marca "Audi" viria na frente da "Renault", pois ao ordená-los em ordem alfabética, ela é retornada primeiro.

5. Criar uma função que receba como parâmetro o nome de uma marca e retorne a lista de seus modelos. Caso o nome da marca informada não exista no arquivo JSON, você deve retornar um array vazio. A busca deve desconsiderar caracteres maiúsculos e minúsculos. Exemplo: um parâmetro enviado como “HONDA”, deve encontrar a marca “Honda”. Exemplo de retorno caso não encontre a marca: []. Exemplo de retorno caso encontre a marca:

["Modelo 1", "Modelo 2", "Modelo 3"].

6. Utilizar o Express.js para expor 5 endpoints que farão o consumo dos métodos criados anteriormente. Os itens 1 a 4 devem receber as requisições a partir do método HTTP GET. O item 5 deve ser uma requisição POST, recebendo o parâmetro “nomeMarca” a partir de um objeto JSON no body da requisição, exemplo: {"nomeMarca": "Marca A"}. As rotas devem seguir a seguinte nomeação:

Item 1 - /marcas/maisModelos

Item 2 - /marcas/menosModelos

Item 3 - /marcas/listaMaisModelos/X

Item 4 - /marcas/listaMenosModelos/X

Item 5 - /marcas/listaModelos

Observação: nos exemplos de rotas dos itens 3 e 4, o X quer dizer o parâmetro que será recebido pela rota.

## Respostas Finais

Os alunos deverão desenvolver a prática e depois, responder às questões objetivas.



Atentar-se para o utilizar o mesmo arquivo JSON fornecido, sem realizar alterações nele, evitando assim inconsistências nas respostas corretas.