



FULL DIGITAL PERFORMANCE

Projeto Classificatório

Processo seletivo - Web Development

Lucas Fernandes Tolotto

Sorocaba/SP
2021

1 ESCOLHA DA LINGUAGEM

A linguagem escolhida foi o JavaScript, a escolha foi feita seguindo a recomendação descrita no documento do teste prático, estudo e pratico essa linguagem há 3 meses.

2 FUNCIONALIDADES DA RESOLUÇÃO

2.1 LEITURA JSON

Para fazer a leitura do JSON foi desenvolvida uma função que utiliza o módulo require para consumir o arquivo, este módulo por sua vez utiliza o caminho do JSON para funcionar, logo, foi usado na função a variável “__dirname” do node.js para pegar o diretório atual da pasta em conjunto com o nome do arquivo do json, desta maneira, o programa deve funcionar em qualquer computador que baixar o arquivo, pois o caminho do arquivo não é estático, sendo sempre definido e atualizado pela variável __dirname para cada computador que rodar o código. Entretanto, o require só aceita caminhos com barra normal, e o __dirname retorna um caminho com barra invertida, então foi usado o método replace para substituir as barras invertidas, concluindo a seguinte função:

```
function readjson(jsonfilename){
    var caminhorequire = __dirname.replace(/\\/g, '/') + jsonfilename
    var dados = require(caminhorequire)
    return dados
}
var json = readjson("/broken-database.json")
```

2.2 CORREÇÃO DOS NOMES

A função desenvolvida para corrigir os nomes faz uso dos métodos “JSON.stringify” e “replace”, o primeiro método foi aplicado para trocar o tipo da variável “object” devolvida pela função de leitura do JSON em tipo “string”, após isso, foi usado o replace, que funciona apenas em strings, para trocar os caracteres corrompidos, concluindo a seguinte a função:

```
function cnomes(dados){
    dados = JSON.stringify(dados)
    dados = dados.replace(/æ/g, "a")
    dados = dados.replace(/¢/g, "c")
    dados = dados.replace(/ø/g, "o")
    dados = dados.replace(/ß/g, "b")
    return dados
}
```

2.3 CORREÇÃO DO TIPO DOS PREÇOS

A correção do tipo dos preços foi aplicada através de uma simples função que percorre os dados do JSON e faz o casting do atributo “price” para float, utilizando a função “parseFloat()”, para realizar isso foi necessário trocar o tipo da variável que continha o JSON de string para object novamente, através do método JSON.parse(), concluindo a seguinte função:

```
function cprice(dados){
  dados = JSON.parse(json)
  for(i in dados)
  {
    dados[i]["price"] = parseFloat(dados[i]["price"])
  }
  return dados
}
```

2.4 CORREÇÃO DAS QUANTIDADES NULL

A função desenvolvida para corrigir as quantidades que não aparecem foi feita de maneira que a variável que contém o JSON é percorrida com um FOR verificando através de um IF se a quantidade atual é igual a “null”, caso verdadeiro a quantidade atual é definida para 0, concluindo a seguinte função:

```
function cquantity(dados){
  for(i in dados)
  {
    if(dados[i]["quantity"] == null)
    {
      dados[i]["quantity"] = 0;
    }
  }
  return dados
}
```

2.5 EXPORTAR ARQUIVO JSON CORRIGIDO

A exportação do arquivo JSON foi feita através do node.JS, para isso foi necessário importar o módulo de sistema do node para manipulação de arquivos, esse módulo é responsável pela leitura e criação de arquivos, o módulo é incluído através do seguinte comando:

```
const fs = require('fs') // fs = FILE SYSTEM
```

Após isso, foi usado o método “.writeFile” para criar o arquivo de saída, esse método recebeu 3 parâmetros, sendo eles:

1. Nome do arquivo que vai ser criado (“saida.json”)
2. Variável que contém o arquivo JSON.

Para isso foi usado o JSON.stringify novamente, passando dois argumentos extras para deixar o arquivo JSON mais confortável de ser lido, com uma informação em cada linha.

```
const file = JSON.stringify(jsonData, null, 2)
```

3. Uma função para ser chamada quando o método write file é executado.

Essa função por convenção é algo que é usado para retornar um erro caso a construção do arquivo falhe.

```
const finished = (error) => {  
  if(error){  
    console.error(error)  
    return;  
  }  
}
```

Dessa maneira, foi concluída a seguinte função:

```
const fs = require('fs')  
function exportToJsonFile(jsonData) {  
  const finished = (error) => {  
    if(error){  
      console.error(error)  
      return;  
    }  
  }  
  const file = JSON.stringify(jsonData, null, 2)  
  fs.writeFile("saida.json", file, finished)  
}
```

2.6 IMPRIMIR LISTA ORDENADA

O desafio de printar a lista de nomes dos produtos ordenados por categoria em ordem alfabética e ID em ordem crescente foi resolvido pelos seguintes passos:

1. Fazer uma array com todas as categorias em ordem alfabética
2. Fazer uma array da quantidade de ocorrências de cada categoria no JSON, seguindo a ordem alfabética.
3. Fazer uma array auxiliar que contém os ID's dos produtos da categoria na ordem crescente, essa array está em um loop e quando sua variável quantidade atinge a quantidade de ocorrências da categoria a variável com o JSON é percorrida para printar os nomes dos produtos com os ID's da array auxiliar, que já estão em ordem crescente.

Para desenvolver o primeiro passo foram utilizados os métodos de array "include", "push", "sort", basicamente o arquivo é percorrido e caso a categoria já esteja na array nada acontece, caso contrário a categoria é colocada na array através do push. Por fim, a array é colocada em ordem crescente com o método sort.

```
for(i in dados[i]["category"])
{
    if(array.includes(dados[i]["category"]))
        quantidade = quantidade; // não fazer nada
    else
        array.push(dados[i]["category"])
}
array.sort(); //ordem alfabética
```

O segundo passo foi feito através de um for aninhado na array das categorias, para cada categoria na array o JSON é varrido e quando existe uma ocorrência da categoria atual ser igual à da array de categorias, uma quantidade é incrementada, ao resetar o loop essa quantidade é colocada na array das quantidades, dessa maneira são pegadas as quantidades de ocorrência de cada categoria já em ordem alfabética.

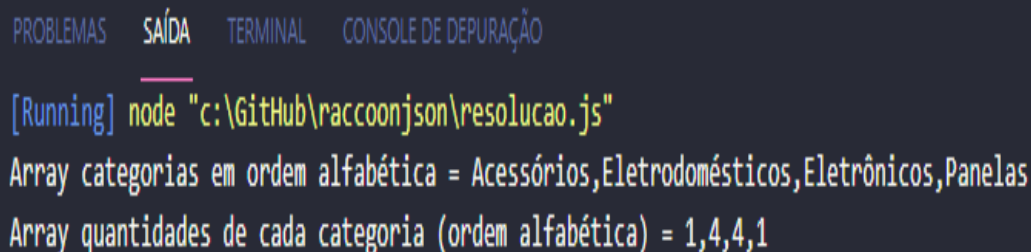
```

    for(i in array)
    {
        if(i==0) // primeira ocorrência
            quantidade = quantidade; // não fazer nada
        else
            arrayquantidades.push(quantidade)
        quantidade = 0;
        for(j in dados)
        {
            if(dados[j]["category"] == array[i])
            {
                quantidade++;
            } // if
        } // for j in dados
    } // for i in array

    arrayquantidades.push(quantidade) // for acaba, então para a última
    quantidade é necessário mais um comando

```

Dessa maneira já temos os 2 primeiros passos concluídos:



```

PROBLEMAS  SAÍDA  TERMINAL  CONSOLE DE DEPURAÇÃO
[Running] node "c:\GitHub\raccoonjson\resolucao.js"
Array categorias em ordem alfabética = Acessórios,Eletrônicos,Eletrônicos,Panelas
Array quantidades de cada categoria (ordem alfabética) = 1,4,4,1

```

Após isso, o terceiro passo segue o mesmo raciocínio de loop de repetição aninhado do segundo passo, para cada categoria na array o JSON é varrido e quando existe uma ocorrência da categoria atual ser igual à da array de categorias uma variavel quantidade é incrementada e o id do produto atual é passado para uma nova array auxiliar, que por sua vez também é colocada em ordem crescente com o método sort a cada interação. Quando a quantidade de ocorrências da categoria atual é igual à quantidade de ocorrências na array das quantidades, o JSON é varrido novamente para printar todos os produtos com ID's que estão na array auxiliar, dessa maneira o desafio de printar a lista de nomes dos produtos ordenados por categoria em ordem alfabética e ID em ordem crescente é resolvido, obtendo o seguinte resultado:

Categoria = Acessórios

ID = 1911864 Mouse Gamer Predator cestus 510 Fox Preto

Categoria = Eletrodomésticos

ID = 1316334 Refrigerador bottom Freezer Electrolux de 02 Portas Frost Free com 598 Litros

ID = 6502394 Fogão de Piso Electrolux de 04 bocas, Mesa de Vidro Prata

ID = 9576720 Forno Micro-ondas Panasonic com capacidade de 21 Litros branco

ID = 9628920 Lava & Seca 10,2 Kg Samsung Eco bubble branca com 09 Programas de Lavagem

Categoria = Eletrônicos

ID = 2162952 Kit Gamer acer - Notebook + Headset + Mouse

ID = 3500957 Monitor 29 LG FHD Ultrawide com 1000:1 de contraste

ID = 8875900 Smart TV 4K Sony LED 65" 4K X-Reality Pro, UpScalling, Motionflow XR 240 e Wi-F

ID = 9746439 Home Theater LG com blu-ray 3D, 5.1 canais e 1000W

Categoria = Panelas

ID = 5677240 Conjunto de Panelas antiaderentes com 05 Peças Paris

Através do seguinte código (terceiro passo):

```
for(i in array)
{
    quantidade = 0;
    aux = [];
    for(j in dados)
    {
        if(dados[j]["category"] == array[i])
        {
            aux.push(dados[j]["id"])
            aux.sort();
            quantidade++;
        }

        if(quantidade == arrayquantidades[i])
        {
            console.log("\nCategoria = " + dados[j]["category"])
            quantidade = 0; // quantidade resetada para não printar em todo for
            //console.log("bateu")
            for(l in aux)
            {
                for(k in dados)
                {
                    if(dados[k]["id"] == aux[l])
                    {
                        console.log("ID = " + dados[k]["id"] + "      " + dados[k]["name"])
                    }
                }
            }
        } // if
    } // for j in dados
} //for i in array
```

2.7 IMPRIMIR VALOR TOTAL POR CATEGORIA

O desenvolvimento da função que imprime o valor total por categoria também usa a array de categorias em ordem crescente, é feito um FOR nessa array de categorias e em cada interação o JSON é varrido, então, existe uma verificação para quando os dados da categoria do JSON forem iguais aos da categoria atual do FOR, seguindo o mesmo raciocínio das funções anteriores. Quando essa verificação é verdadeira é armazenado em uma variável de soma a quantidade do produto atual multiplicado pelo preço, essa variável é adicionada a uma array soma, que é tem seu valor total somado e printado na tela quando o loop volta ao primeiro for, ou seja, quando a categoria é trocada, após isso essa array soma é zerada e o loop passa para a próxima categoria, dessa maneira, conseguimos printar o valor total de estoque de cada categoria, através da seguinte função:

```
for(i in arraycategoria)
{
    if(i==0)
        arraycategoria = arraycategoria; // não fazer nada
    else
    {
        soma = 0;
        for(l in arraysomas)
            soma += arraysomas[l]
        console.log("Soma total do valor de estoque da categoria = R$" + soma)
        arraysomas = [];
    }
    quantidade = 0;
    console.log("\n" + arraycategoria[i])
    for(j in dados)
    {
        if(dados[j]["category"] == arraycategoria[i])
        {
            soma = dados[j]["quantity"] * dados[j]["price"]
            arraysomas.push(soma)
        }
    }
}
// repetir o processo para a ultima categoria pois o for acaba antes do if
soma = 0;
for(x in arraysomas)
    soma += arraysomas[x]
console.log("Soma total de estoque da categoria = " + soma)
```

PROBLEMAS SAÍDA TERMINAL CONSOLE DE DEPURAÇÃO

Preço do valor do total de estoque por categoria

Acessórios

Soma total do valor de estoque da categoria = R\$0

Eletrodomésticos

Soma total do valor de estoque da categoria = R\$315752.67000000004

Eletrônicos

Soma total do valor de estoque da categoria = R\$203989.2

Panelas

Soma total de estoque da categoria = 4049.64

[Done] exited with code=0 in 0.09 seconds

3 RESULTADO FINAL

Através do comando node resolucao.js a solução se demonstrou funcional e o resultado no terminal foi:

```
PS C:\GitHub\raccoonjson> node resolucao.js
Array categorias em ordem alfabética = Acessórios,Eletrodomésticos,Eletrônicos,Panelas
Array quantidades de cada categoria (ordem alfabética) = 1,4,4,1
-----

Nomes dos produtos ordenados por categoria em ordem alfabética e ID em ordem crescente:

Categoria = Acessórios
ID = 1911864      Mouse Gamer Predator cestus 510 Fox Preto

Categoria = Eletrodomésticos
ID = 1316334      Refrigerador bottom Freezer Electrolux de 02 Portas Frost Free com 598 Litros
ID = 6502394      Fogão de Piso Electrolux de 04 bocas, Mesa de Vidro Prata
ID = 9576720      Forno Micro-ondas Panasonic com capacidade de 21 Litros branco
ID = 9628920      Lava & Seca 10,2 Kg Samsung Eco bubble branca com 09 Programas de Lavagem

Categoria = Eletrônicos
ID = 2162952      Kit Gamer acer - Notebook + Headset + Mouse
ID = 3500957      Monitor 29 LG FHD Ultrawide com 1000:1 de contraste
ID = 8875900      Smart TV 4K Sony LED 65" 4K X-Reality Pro, UpScalling, Motionflow XR 240 e Wi-F
ID = 9746439      Home Theater LG com blu-ray 3D, 5.1 canais e 1000W

Categoria = Panelas
ID = 5677240      Conjunto de Panelas antiaderentes com 05 Peças Paris
-----

Preço do valor do total de estoque por categoria

Acessórios
Soma total do valor de estoque da categoria = R$0

Eletrodomésticos
Soma total do valor de estoque da categoria = R$315752.670000000004

Eletrônicos
Soma total do valor de estoque da categoria = R$203989.2

Panelas
Soma total do valor de estoque da categoria = 4049.64
```