

Teste Prático Processo Seletivo Analista em Tecnologia

Por Kevin Soares (devkevinsoares@gmail.com)

Uma breve explicação de cada funcionalidade

Comecei importando os arquivos json para o script

var **json** importa o banco de dados corrompido

var **jsonProducts** foi uma solução que encontrei para criar um banco de dados para o estoque

fs foi para fazer a exportação dos arquivos json quando terminar de corrigir os problemas

```
var json = require("./broken-database.json");  
  
var jsonProducts = require("./produtos.json");  
  
var fs = require("fs");
```

Essa função corrige o problema dos preços estarem em formato de String, convertendo eles para o tipo Inteiro

```
function setPriceInt(key, value) {  
  
    if (key === "price") {  
  
        return parseInt(value);  
  
    }  
  
    return value;  
  
}  
  
var stringified = JSON.stringify(json, setPriceInt);
```

Peguei o banco de dados em formato de string e foi corrigido todos os erros de string substituindo eles usando *.replace*

```
stringified = stringified

.replace(/æ/g, "a")

.replace(/&/g, "e")

.replace(/ç/g, "c")

.replace(/ø/g, "o")

.replace(/ß/g, "b");
```

Transformei em json novamente para fazer um *forEach* e peguei a quantidade pela data e verifiquei se não possuía esse elemento no objeto e então eu coloquei novamente a *quantity* corrigindo o problema de ter sumido as quantidades.

```
var jsonObject = JSON.parse(stringified);

jsonObject.forEach(function (data, index) {

    if (!data.quantity) {

        jsonObject[index].quantity = 0;

    }

});
```

Logo depois exportei o banco de dados corrigido **resolucao.json** terminando a primeira questão: Recuperação dos dados originais do banco de dados

```
fs.writeFile("resolucao.json", JSON.stringify(jsonObject), function
    (err) {

        if (err) throw err;

        console.log("Resolução JSON Criada com Sucesso!");

    });
```

Comecei a questão 2 Validação do banco de dados corrigido.

Coloquei o banco de dados em ordem alfabética baseado pela categoria

```
keys = ["category"];

jsonObject.sort(function (a, b) {

    var ret = 0,

    i = 0;

    while (ret == 0 && i < keys.length) {

ret = a[keys[i]] > b[keys[i]] ? 1 : a[keys[i]] < b[keys[i]] ? -1 :

0;

i++;

    }

    return ret;
```

Logo depois eu ordenei cada um dos elementos baseado no index mapeando o json

```
const indexedJson = jsonObject.map((item, index) =>

    Object.assign(item, { index })

);
```

Terminando assim a Questão 2 a) exportando o banco de dados ordenado em ordem alfabética e numerado com seu index criando o arquivo **saída.json**

```
function ImprimirProdutos() {

    console.log(indexedJson);

fs.writeFile("saida.json", JSON.stringify(indexedJson), function (err)

    {

        if (err) throw err;

        console.log("Saida JSON Criada com Sucesso!");

    });

}

ImprimirProdutos();
```

E pra finalizar a questão 2 b) calculei o valor do estoque criando um novo arquivo json **produtos.json** para salvar o preço total do estoque verificando em um *forEach* se a categoria bate-se adicionava o preço no **totalPrice** nessa categoria criando assim um preço total do estoque que está no arquivo **saida-produtos-estoque.json**

```
function CalcularValorEstoque() {

    indexedJson.forEach(function (data) {

        jsonProducts.forEach(function (product) {

            if (data.category === product.category) {

                product.totalPrice += data.price * data.quantity;

            }

        });

    });

    fs.writeFile(

        "saida-produtos-estoque.json",

        JSON.stringify(jsonProducts),

        function (err) {

            if (err) throw err;

            console.log("Produtos JSON Atualizado com Sucesso!");

        }

    );

}

CalcularValorEstoque();
```

O porquê da escolha da linguagem

Familiarizado com javascript, venho estudando a anos junto com html, css, javascript e também o react e verifiquei que seria a melhor maneira de fazer esse teste com eficácia e elegância.

Tratamentos feitos no código para evitar bug

Explicado no tópico de explicação das funcionalidades fiz cada função pensando no melhor desempenho criando e exportando o banco de dados corrigido por etapas e exportando o arquivo de saída somente quando estiver tudo concluído.