



FULL DIGITAL PERFORMANCE

## Documentação resolução JS

Processo seletivo - Web Development

# Funcionalidades

## 1. Ler arquivo JSON

Utilizado a funcionalidade *'require'* para importar o arquivo JSON para JS, o motivo para utilizar o *require* ao invés de *'import'* foi porque o ambiente de teste será utilizando *node.js*.

## 2. fixName

A função *fixName* utiliza a funcionalidade *'map'* e cria uma nova *'array'* com os itens da variável *'brokenDatabase'* alterando conforme a *'callback function'*, dentro da função criado um loop utilizando o *'for'* para substituição das letras alteradas em cada propriedade *'name'* dentro do objeto.

## 3. fixPrice

A função *fixPrice* utiliza a funcionalidade *'map'* e cria uma nova *'array'* com os itens da variável *'brokenDatabase'* alterando conforme a *'callback function'*, dentro da função é atribuído um novo valor de propriedade de objeto com o tipo *number*

## 4. fixQuantity

A função *fixQuantity* utiliza a funcionalidade *'map'* e cria uma nova *'array'* com os itens da variável *'brokenDatabase'* alterando conforme a *'callback function'*, dentro da função é atribuído à propriedade *'quantity'* o próprio valor dela, feito uma estrutura condicional para verificar se o valor for abaixo de 0 exiba uma mensagem de erro no console, caso o valor seja maior que 0 ele atribui o próprio valor caso contrário o valor atribuído é 0.

## 5. exportJsonFile

A função *'exportJsonFile'* converte o objeto para *string* e cria e escreve em um arquivo chamado *saída.json*, caso ocorra um erro ele apresenta um erro no console.

## 6. listProducts

A função *'listProducts'* cria uma *array* com todos os objetos corrigidos e ordena por ID

em ordem crescente e por categoria em ordem alfabética. Na ordenação por categoria é colocado um método para não haver problemas com caixa alta.

## 7. warehouseValue

A função *'warehouseValue'* cria uma variável total que recebe o valor de 0 inicialmente, criado um loop onde é atribuído a variável total a soma do valor total com a multiplicação do preço com quantidade de cada item.

## JavaScript

A escolha do JavaScript como linguagem para desenvolvimento foi porque é uma linguagem leve, simples e com bastante documentação para consulta e facilidade de entendimento das funcionalidades.

## Tratamentos e considerações

Tratamento de ocorrência / prevenção de erros na execução das funções *'fixQuantity'*, *'exportJsonFile'* e *'listProducts'*.

Na função *fixName* foi verificado a possibilidade do item estar sem codificação utf-8 mas não era o caso portanto prossegui com a substituição de caracteres.