



FULL DIGITAL PERFORMANCE

## Projeto Classificatório

Processo seletivo - Web Analytics

Candidato: Stefanie de Oliveira Siqueira Ernane.

Documentação: Teste prático-projeto classificatório Rocky

## Funcionalidade

O projeto tinha como objetivo corrigir todos os dados corrompido do 'Broken-database.json', erros localizados em 'name', 'price' e 'quantity'. Além de criar funções onde os arrumasse, tem também função para ler o arquivo json e para exportar o mesmo, porém, atualizado.

Para a validação do banco de dados, foi desenvolvido outra função que o ordenassem em ordem alfabética através das 'category' e em seguida, ordem crescente através do 'id' de cada produto.

Explicando melhor cada função:

objJson- lê e transforma o json inicial "broken-database.json" em objeto, para que o mesmo possa ser manipulado. Uso do JSON.parse.

Para percorrer e acessar cada objeto foi utilizado o "for(var i=0; i<data.length; i++).

subsName- função responsável por substituir os erros "æ", "ç", "ø", "ß" por "a", "c", "o" e "b". através do split que divide a string dando fácil acesso ao erro que pretendemos arrumar e join responsável por juntar a string novamente, após que o erro foi corrigido.

numberPrice- responsável por alterar o typeof dos "price" que de inicio estava em string, para number, com a ajuda do parseFloat.

insereQuantity-foi usado o método hasOwnProperty para verificar se o objeto tinha ou não o "quantity" inserido, se o caso fosse negativo, a função inseria quantity=0 no objeto que de início tinha ausência.

exportarJson- diferente do objJson foi utilizado o JSON.stringify que transforma objeto em um arquivo .json, sendo assim exportando o arquivo com auxílio da variável guardandoDados, que recebia objJson(jsonInicial) que a partir do if atualizou todo o arquivo tirando seus erros.

Seguindo para a segunda parte do projeto:

Ordenando- fez a comparação de dois em dois para facilitar, tendo retorno de três valores -1,0 e 1. Organizando em categoria em ordem alfabética e em seguida em ordem crescente através dos valores do id.

qtdCategoria- como antes dessa função as categorias foram ordenadas não foi preciso fazer um loop para identificar se a categoria existe. A array quantityTotal armazena a quantidade de cada categoria através do .push. o guardandoCategory se inicia com ' ', e vai guardando a categoria anterior ajudando o loop fazer todas as somas.

Como foi realizar o projeto:

Foi um desafio, onde tive que aprender Javascript uma vez em que não era uma linguagem de meu conhecimento e ao mesmo momento realizar um projeto. Gostei bastante de realizar, pesquisar e adquirir uma nova linguagem na minha carreira profissional.