

O algoritmo começa com uma função require que atribui para a variável de objeto os dados do arquivo "broken-database.json" fornecido para o teste, depois disso a variável de Objeto "bancoCerto" recebe os dados da variável "dadosQuebrados" para tratamento.

O primeiro código percorre e mapeia a propriedade "name" de todos os objetos para encontrar e substituir as letras estranhas por letras corretas.

Depois mais uma vez o objeto é acessado e percorrido transformando todas as propriedades "price" para o tipo number float.

Então é feito uma checagem em todos os objetos para verificar se todos contem a propriedade "quantity" e para aquele que não tem é feito um tratamento para inserir a propriedade no lugar correto, mantendo o padrão.

Com o banco corrigido é exportado um novo arquivo de nome "saida.json" com o banco certo.

Em seguida são feitos os testes de validação, começando com a organização do banco de acordo com o critério estabelecido primeiro em ordem de categoria e depois em ordem numérica crescente, respeitando a primeira ordem. Na sequência é impresso no terminal os objetos segundo a organização estabelecida anteriormente.

Após isso o código cria um novo objeto de nome "total" e verifica se já existe a propriedade com o nome de "category do banco corrigido" no banco "total" e criar se ainda não existe, e armazenar a soma do valor de "price do banco corrigido" multiplicado pela "quantity do banco corrigido".

E então finaliza imprimindo no terminal o valor total dos produtos separados por estoque.

Notas:

Para evitar Bugs, as estruturas de verificação são baseadas no tamanho da variável de objeto, pois se acrescida mais tarde ainda funcionará.

Principais fontes de estudo para esse teste:

<https://www.alura.com.br/>

<https://developer.mozilla.org/pt-BR/>

<https://www.w3schools.com/>