O algoritmo tem como objetivo corrigir os problemas encontrados no Banco de Dados corrompido, utilizando as seguintes funções:

Função lerArquivo()

Esta função recebe o endereço do arquivo JSON com o Banco de Dados corrompido e retorna seu conteúdo em um formato que pode ser manipulado pela aplicação.

Função corrigirNomes()

Esta função recebe os produtos cadastrados no Banco de Dados, realiza alterações no nome de cada produto e retorna os produtos cadastrados, com os nomes corrigidos. A função substitui o caractere "æ" por "a", "¢" por "c", "ø" por "o" e "ß" por "b".

Função corrigirPrecos()

Esta função recebe os produtos cadastrados no Banco de Dados, realiza alterações no preço de cada produto e retorna os produtos cadastrados, com os preços corrigidos. A função converte os preços de cada produto para o tipo Number.

Função corrigirQuantidades()

Esta função recebe os produtos cadastrados no Banco de Dados, realiza alterações na quantidade de cada produto e retorna os produtos cadastrados, com as quantidades corrigidas. A função verifica se a propriedade "quantity" existe e, caso o resultado seja falso, a adiciona ao produto com o valor "0".

Função exportarArquivo()

Esta função recebe o endereço do arquivo JSON com o Banco de Dados corrompido e cria um arquivo JSON com os dados corrigidos. É esta função que determina o arquivo JSON a ser alterado, pois é a partir dela que as funções lerArquivo(), corrigirNomes(), corrigirPrecos() e corrigirQuantidades() são executadas.

Função imprimirListaProdutos()

Esta função recebe o endereço do arquivo JSON com o Banco de Dados corrigido e exibe na tela os nomes de cada produto, ordenados por categoria em ordem alfabética e por id em ordem crescente.

Função calcularValorEstoque()

Esta função recebe o endereço do arquivo JSON com o Banco de Dados corrigido e exibe na tela o valor total do estoque de cada categoria. As categorias são ordenadas pela ordem em que aparecem no arquivo JSON.

Para a implementação deste algoritmo, foi utilizada a linguagem de programação JavaScript. Como a aplicação utiliza um Banco de Dados NoSQL orientado a documentos, a utilização do ambiente de execução Node.js possibilita que o back-end da aplicação possa se comunicar com o front-end sem a necessidade de tratamento ou conversão de dados, pois os documentos e a linguagem de programação utilizam o mesmo modelo de dados.