

Situação Problema 1: Recuperação dos dados originais do banco de dados:

1ª Função:

A função anônima **lerArquivoJson**, uma *arrow function*, tem como parâmetro “arquivo” que se trata do local do arquivo JSON que será lido.

Utilizei o bloco *try-catch* para tratar possíveis erros na execução da função.

A leitura é realizada através da função *readFileSync* do módulo *fs* (*file system*) e armazenada na variável “dados”.

O *JSON.parse* é utilizado para converter a *string JSON* que está na variável “dados” em um objeto *JavaScript*.

Caso ocorra algum erro, a função irá retornar a mensagem “Erro ao ler o arquivo” juntamente com os detalhes do erro.

2ª Função:

Uma *arrow function*, que aceita um parâmetro “dado”, que é o *array* de objetos que contém as propriedades *marca* e *nome*. O método *map* percorre cada objeto no *array* dado.

Criei as constantes *newMarca* e *newNome* que verifica se o atributo *marca* e *nome* estão presentes no objeto verificado, estando presente, utiliza o *replace* para substituir as ocorrências de 'æ' por 'a' e 'ø' por 'o' e o novo resultado fica armazenado em *newMarca* e *newNome*, respectivamente.

O *return* dessa função é o novo objeto contendo as propriedades do objeto original (...item), com as devidas substituições de *marca* e *nome*, quando necessárias.

3ª Função:

Semelhante a 2ª função, essa *arrow function* também utiliza o método *map* para percorrer cada objeto no *array* dado.

Para cada objeto, o *parseInt* converte o valor da propriedade *vendas* para um número inteiro e o resultado é armazenado em *newVendas*.

4ª Função:

Função composta chamando as funções `corrigirNomes` e `corrigirVendas` para correção do banco de dados. O resultado da chamada `corrigirNomes` é utilizada como argumento para função `corrigirVendas`. O resultado das alterações é retornado pela função `alterarDatabase`.

5ª Função:

A *arrow function* `escreverJson`, têm como parâmetros: “local” que é o caminho para onde os arquivos JSON ficarão salvos e “dados”, que são os dados que serão convertidos para JSON. O `JSON.stringify` converte os dados fornecidos (dados) em uma *string* JSON e o resultado é armazenado na variável `json`.

A função `fs.writeFileSync` escreve a *string* JSON no arquivo especificado pelo caminho local.

Caso ocorra algum erro durante a escrita do arquivo, o bloco *catch* é acionado e retorna uma mensagem de erro, incluindo detalhes sobre o erro.