# Documentação da resolução do Projeto Classificatório da Rocky

## Introdução

Este repositório destina-se a alocar os materiais utilizados para desenvolver a resolução do desafio da Rocky. Utilizei o javascript por ter alguma experiência com ela em pequenos projetos pessoais e para faculdade. Tenho certa familiaridade com linguagem C e muitas das sintaxes dessa linguagem têm suporte em javascript, embora nem sempre sejam as mais indicadas.

Para executar o programa é necessário instalar o node (mais informações [aqui](https://nodejs.org/en/download/)), baixar os arquivos do repositório em uma pasta, acessar essa pasta via prompt de comando e executar o comando “node resolução.js”.

A seguir, detalharei as funcionalidades.

## Funcionalidades

### 2.1. Leitura do arquivo

Para leitura do arquivo JSON, por ser um arquivo estático, criei uma função que recebe o nome do arquivo e executa a função ‘require()’ com esse nome. Para criar esta função e a função de escrita do JSON, adaptei do site:

<https://medium.com/@osiolabs/read-write-json-files-with-node-js-92d03cc82824>

### 2.2. Correção dos nomes

O documento cita que o banco de dados foi corrompido e os caracteres do campo “nome” foram substituídos "a" por "æ", "c" por "¢", "o" por "ø", "b" por "ß". Para resolver criei uma função que recebe três argumentos: a array “objArray” de objeto produtos, uma array “takeOutArray” de caracteres a serem removidos e uma array “putInArray” de caracteres que devem entrar como substitutos. A substituição se dará na mesma posição destas arrays, ou seja, o caracter na primeira posição da array “takeOutArray” será substituído pelo caracter na primeira posição da array “putInArray”. Através de laços de repetição, a propriedade nome de cada objeto é testado com uma expressão regular buscando cada instância de cada caracter do array “takeOutArray”, e substituindo-os pelo caracter da posição correspondente da array “putInArray”.

Após isso, outra expressão regular altera a primeira letra de cada palavra para maiúscula, para manter o padrão visto no arquivo de entrada. Ponto para melhoria fica a adição de exceções a essa operação, onde poderíamos evitar que palavras não significativas como “de” ou “com” fossem alteradas.

A expressão regular para primeira letra de cada palavra foi baseada no site:

<https://www.w3resource.com/javascript-exercises/javascript-string-exercise-9.php>

### 2.3. Correção dos preços

Sobre os preços, o problema é o tipo errado atribuído, onde em alguns objetos o preço está definido como tipo string ao invés um tipo numérico. A função criada recebe a array de objetos e executa o parseFloat na propriedade “price” de cada um dos objetos.

### 2.4. Correção das quantidades

O problema das quantidades refere-se a falta da propriedade “quantity” em alguns objetos. Todos os objetos que tinham essa falha possuíam quantidade igual a “0”. A função criada recebe a array de objetos e testa se cada um possui a propriedade “quantity”. Os que não possuírem recebem a nova propriedade já atribuída com o valor “0”.

### 2.5. Ordenação por Categoria e Id

Para ordenar os produtos por categoria e id, foram criadas duas funções. A sortCategoriaId, que deve ser usada como argumento da função sort(), e realiza a ordenação primeiro por categoria em ordem alfabética, e dentre os itens de mesma categoria, ordena por id em ordem crescente. A segunda função recebe a array de objetos e aplica a função sort usando a função acima como argumento. Depois imprime as colunas “category”, ”id” e ”name” usando console.table().

A função sortCategoriaId foi baseada no site:

<https://www.sitepoint.com/sort-an-array-of-objects-in-javascript/>

### 2.6. Exportação do Arquivo JSON

Criei a função de exportação baseada no mesmo site citado do item 2.1., recebendo a array de objetos, usando stringfy (setando o parâmetro “space” como 1 para melhorar a legibilidade e seguir padrão do arquivo de entrada) e usando o writeFile para exportar.

## Conclusão

Aprendi muito enquanto pesquisava e testava as funções para desenvolver este trabalho. Defini como próximos passos, pesquisar mais sobre call-back functions com arrow functions e as funções nativas do javascript para arrays, como .map() e .forEach().