

## Documentação – Banco corrigido.

```
var dados = require('./broken-database.json');
/*criamos um const dados que recebe um require, que tem o intuito de trazer as informações do arquivo json ligando com o javascript*/
console.log(dados);

function database(Listadados){
    console.log('listadados'); //vai mostrar no terminal o que vc pedir no parâmetro

    var array = []; //variavel que recebe um array por onde entrará as listas de dados
    for(var count = 0; count <= listadados.length; count++){ /*0 for cria um loop que trará os dados do arquivo json, o length tras todos eles, porém só para após o loop acabar*/
        array.push(listadados[count]); /*0 push vai adicionar os elementos no array da variavel 'array', tudo isso será trago no parametro da função que levará em conta os valores do array*/
    }
    return array; //retorna para função os dados do array
}

console.log(database(dados));
/*mostra no terminal a função que fizemos com base no que está no arquivo 'broken-database.json'*/

//FUNÇÃO PARA CORRIGIR CARACTERES INCORRETOS
function convert(carac){ //criação de uma função para converter os caracteres
    carac.forEach(i => { //Para cada item do array criado "i", ele busca o caracter que vc chama nas linhas abaixo
        var stri = i.name; //variável stri recebe a função acima e pega valores da linha que contém a chave name
        stri = stri.replace(/ø+/ig, "o") //Trazendo a variável stri e usando o método replace que pega os caracteres bugados e troca por novos
            .replace(/æ+/ig, "a")
            .replace(/ç+/ig, "c")
            .replace(/ß+/ig, "b");

        i.name = stri;
        /*console.log(stri);*/ //Mostra os caracteres convertidos da variável stri
    });

    return carac;
}
```

```
    dados = convert(dados); //criamos uma variável que pega os resultados anteriores das funções anteriores e guarda atualizado para quando iniciar uma função nova os valores anteriores
```

```
    //FUNÇÃO PARA CONVERTER O PREÇO DE STRING PARA FLOAT
```

```
function float(type){  
    type.forEach(i => { //Usando o método ForEach (para cada item do array criado "i")
```

```
        var num = i.price; //Criar uma variável que recebe o array da chave price do Json
```

```
        if(num === "" + num){ //O if compara se esta variável é idêntica ela mesma que já está no tipo string
```

```
            num = parseFloat(num); //se não, fará a conversão, usando o método parseFloat que converterá para tipo float  
        }
```

```
        i.price = num;  
    });
```

```
    return type;  
}
```

```
    dados = float(dados);  
console.log(dados); //mostra a função funcionando
```

```
    //FUNÇÃO QUE TRÁS A QUANTIDADE QUE FOR IGUAL A 0
```

```
function quantity(qnt){ //criando a função quantity
```

```
    qnt.forEach(i => { //usando o método ForEach (para cada item do array criado "i")
```

```
        if(!("quantity" in i)){ //usando o operador "in", verificamos se existe no objeto a chave quantidade, com o "!" fazemos com que a execução passe de false para true por essa linha usando o if.
```

```
            i.quantity = 0; //Caso não obter esta linha no array, mostrar "quantiti=0"
```

```
        }  
    });
```

```
    return qnt;  
}
```

```
    dados = quantity(dados);  
console.log(dados);
```

```
//CRIANDO UM NOVO ARQUIVO JSON COM O BANCO CORRIGIDO
var banco_corrigido = JSON.stringify(dados); //criando uma variável que r
ecebe um objeto JSON com um método que cria automaticamente um novo arqui
vo json

var file = require('fs');
file.writeFile("saida.json", banco_corrigido, (err, result) => { //O méto
do writeFile exporta o conteúdo para o novo arquivo em json.
    if(err) console.log(err); //Este if compara se caso tiver algum erro
no arquivo novo, caso tiver, o console.log mostrará o erro.
});
```

Obs: Acabei encontrando dificuldades para finalizar o teste, por isso consegui até aqui. Gostaria de agradecer pela oportunidade que me foi dada apesar de ainda estar no primeiro semestre da faculdade, espero de verdade atender as expectativas mesmo não tendo concluído o teste.