# Explicando cada função

Lembrando que dentro do package.json,existe um script chamado:”resolucao”, o qual executará o arquivo “resolucao.js”.

Dentro do arquivo resolucao.js,temos o total de 5 funções:

-fixedName()
-fixedPrice()
-fixedQuantity()
-organizerObjs()
-calculateToltalPrice()

- **explicando cada função**

**-fixedName() =**

A propriedade replaceAll irá substituir todas as letras que foram modificadas pelas originais,seguindo as instruções.

A propriedade string.split(‘ ‘), irá separar cada palavras do ‘name’ e coloca-la em array,e assim fazer um looping para andar palavra por palavra e se a palavra for igual à ‘com’,’de’ ou ‘e’,retorne a palavra para a array sem fazer nenhuma alteração.Se a palavra não for igual,a palavra vai ficar com sua primeira letra maiúscula,pois no arquivo original,está em modo título,ou seja, ‘Carro de Cor Vermelha’.

Depois usando join(‘ ‘),junte tudo em uma única string e depois coloque tudo na string novemente.

E assim fica o código completo.

```
function fixedName() {

    for(var element in dataJson ) {
        var (local var) nameErr: any
        var nameErr = dataJson[element].name
        nameErr = nameErr.replaceAll('ß','b') // w3 school
        nameErr = nameErr.replaceAll('æ','a')
        nameErr = nameErr.replaceAll('ç','c')
        nameErr = nameErr.replaceAll('ø','o')

        var words =  nameErr.split(' ')
        for(index in words) {
            const word = words[index]
            if(word === 'com' || word === 'de' || word === 'e') {
                words.splice(index,1,word)
            }
            else {
                const capitalizeWord = word.charAt(0).toUpperCase() + word.slice(1)
                words.splice(index,1,capitalizeWord)
            }
        }
        nameErr = words.join(' ')
        dataJson[element].name = nameErr
    }
}
```

**fixedPrice() =**

A propriedade parseFloat,irá passar o que era uma string para um número com vírgula,já que há preços com vírgula.

```
function fixedPrice() {
    for(var element in dataJson) {
        var priceErr = dataJson[element].price
        priceErr = parseFloat(priceErr)  // w3 school

        dataJson[element].price = priceErr
    }
}
```

**fixedQuantity() =**

Essa função criará um looping para andar por cada elemento e os elementos que não tiver a categoria quantity,uma nova categoria quantity será adicionada e após a isso.

Logo após isso,tem uma condicional verificando o stats do arquivo ‘saida.json’,se não existir irá direto criar o arquivo com o dados já organizados,se o arquivo existir irá aparecer um aviso no console avisando que o será reescrito.

```
function fixedQuantity() {
    for(element in dataJson) {

        const quantityItem = dataJson[element].quantity
        if(quantityItem === undefined) {
            const item = dataJson[element]

            const {id,name,price,category} = item

            const correctObj = {
                id:id,
                name:name,
                quantity:0,
                price:price,
                category:category
            }

            dataJson.splice(element,1,correctObj)
        }
    }
    // https://stackoverflow.com/questions/36856232/write-add-data-in-json-file-using-node-js

    fs.stat("./saida.json", (err, stats) => {
        const stringDataJson = JSON.stringify(dataJson,null,2)
        // https://www.geeksforgeeks.org/node-js-fs-stat-method/
        if(stats) {
            console.log('The file saida.json is already exist,so will be rewrite')
        }
        fs.writeFile('./saida.json',stringDataJson,(err)=>{
            if(err) {
                console.log(err)
            }
            else {
                organizerObjs()
                calculateTotalPrice()
            }
        })
    });      You, 2 days ago • finish project without documentation
```

**organizerObjs() =**

Essa é um função assíncrona,pois o arquivo ‘saida.json’ pode não existir na hora da execução do código,portanto daria erro,para evitar o erro usa-se uma promisse.

Aqui usa-se a propriedade sort da array para organizar primeiro por categoria e depois por id,se a categoria for a mesma.

```
async function organizerObjs(){ // https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array/sort
// https://stackoverflow.com/questions/40593875/using-filesystem-in-node-js-with-async-await
    let file
    try {
        fs.readFile('./saida.json',(err,fileDataString)=>{
            if(err) {
                console.log(err)
            }
            else {
                file = JSON.parse(fileDataString)
                file.sort((a,b)=>{
                    const categoryA = a.category
                    const categoryB = b.category

                    const idA = a.id
                    const idB = b.id

                    if(categoryA > categoryB) {
                        return 1
                    }
                    else if(categoryB < categoryB) {
                        return -1
                    }
                    else {
                        if(idA > idB) {
                            return 1        You, 31 minutes ago • Uncommitted changes
                        }
                        else if(idA < idB) {
                            return -1
                        }
                    }

                })
                const fileStringify = JSON.stringify(file,null,2) // https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify
                fs.writeFile('./saida.json',fileStringify,(err)=>{
                    if(err) console.log(err)
                })
            }
        })
    }
    catch {
        console.log(err)
    }
}
```

**calculateTotalPrice() =**

Essa função irá andar por categoria e juntar os seis preços de estoque.

```
function calculateTotalPrice() {

    var priceAcessories = priceElectronics = priceHomeAppliences = pricePan = 0
    // preco dos acessorios = preco do Eletronicos = precos eletrodomesticos = preco das panelas

    fs.readFile('./saida.json',(err,dataFile)=>{
        if(err) console.log(err)
        else {
            file = JSON.parse(dataFile)

            for(var element in file) {
                const priceElement = file[element].price
                const categoryElement = file[element].category

                switch(categoryElement) {       You, an hour ago • Uncommitted changes
                    case 'Acessórios':
                        priceAcessories+=priceElement
                        break
                    case 'Eletrodomésticos':
                        priceHomeAppliences+=priceElement
                        break
                    case 'Eletrônicos':
                        priceElectronics+=priceElement
                        break
                    case 'Panelas' :
                        pricePan +=priceElement
                        break
                    default :
                        console.log('There is nothing category to correspind with this product')
                }
            }
        }
    })
}
```

Usando o método switch,para ficar com menos if, e podendo evitar bugs