```
1
2  /*********************************************************************/
3  /***** Função exec recebe o callback de outras funçoes e recebe ****/
4  /********************** parametros "a" e "b" ************************/
5  /*********************************************************************/
6  console.log("**** Função callback ****")
7  function exec(fn,a,b){      // Função com 2 parametros
8      return fn(a,b)          // retorna a e b
9  }
10
11 const somaNoTerminal = (x,y) => console.log(x+y)
12 const subtrairNoTerminal = (w,z) => console.log(w-z)
13 const subtrair = (w,z) => w - z
14
15 exec(somaNoTerminal, 56,38)
16 exec(subtrairNoTerminal, 182,27)
17
18 const r = exec(subtrair, 50, 25)
19 console.log(r)
20
21
22
23 /********* Trabalhando com intervalos de tempo *********/
24 //const cb = () => console.log('Exec...')
25 //setInterval(cb, 1000);
26
27 //setInterval(function() {        // neste exemplo vai executar depois de 3s
28 //    console.log('Exec...')
29 //}, 3000);
30
31
32
33
34 /*************************************************************************/
35 /***** callback - Faz a leitura de um arquivo e retorna com resultado ****/
36 /*************************************************************************/
37 const fs = require('fs')        // Bibliotecas de arquivos do node js
38 const path = require('path')
39
40 const caminho = path.join(__dirname, 'dados.txt')
41
42 // console.log(__dirname)
43
44 function exibirConteudo(err, conteudo){
45     console.log(conteudo.toString())
46 }
47
48
49 console.log("inicio")
50 //fs.readFile(caminho, exibirConteudo)
51 fs.readFile(caminho, (_, conteudo) => console.log(conteudo.toString()))
52 console.log("fim")
53 console.log("")
54
55 console.log("Inicio Sync")
56 const conteudo = fs.readFileSync(caminho)
57 console.log(conteudo.toString())
58 console.log("Fim Sync")
59
```

```
60
61
62
63 /*****************************************************************/
64 /********************** Callback 3 - Usando MAP ******************/
65 /*****************************************************************/
66 const nums = [1,2,3,4,5]
67 const dobro = (n, i, a) => n * 2 + i + a.length
68 console.log(nums.map(dobro))
69
70 const nomes = ['Ana', 'Bia', 'Gui', 'Lia', 'Rafa']
71 const primeiraLetra = texto => texto[0]
72 const letras = (nomes.map(primeiraLetra))
73 console.log(nomes, letras)
74
75
76
77 /***********************************************/
78 /**** DESAFIO: Mostrar o nome de cada item *******/
79 /********* Multiplicar a qtde * preço ************/
80 /***********************************************/
81 const carrinho = [
82     { nome: 'caneta', qtde: 10, preco: 7.99 },
83     { nome: 'impressora', qtde: 0, preco: 649.50 },
84     { nome: 'caderno', qtde: 4, preco: 27.10 },
85     { nome: 'lapis', qtde: 3, preco: 5.82},
86     { nome: 'tesoura', qtde: 1, preco: 19.20 },
87 ]
88
89 const getNome = item => item.nome
90 console.log(carrinho.map(getNome))
91
92 const getTotal = item => item.qtde * item.preco
93 const totais = carrinho.map(getTotal)
94 console.log(totais)
95
96
97
98
99 /*********************************************************/
100 /******* Implementando nossa propria versao de MAP *******/
101 /*********************************************************/
102 Array.prototype.meuMap = function(fn){
103     const novoArray = []
104     for(let i=0; i < this.length; i++){
105         const resultado = fn(this[i], i, this)
106         novoArray.push(` ==> ${resultado}`)
107     }
108     return novoArray
109 }
110
111 const getNome2 = item => item.nome
112 console.log(carrinho.meuMap(getNome2))
113
114 const getTotal2 = item => item.qtde * item.preco
115 const totais2 = carrinho.meuMap(getTotal2)
116 console.log(totais2)
117
118
119
```

```
120
121 /*****************************************************************/
122 /******************* Callback 4 - Usando FILTER ******************/
123 /*****************************************************************/
124 const carrinho2 = [
125     { nome: 'caneta', qtde: 10, preco: 7.99 },
126     { nome: 'impressora', qtde: 0, preco: 649.50 },
127     { nome: 'caderno', qtde: 4, preco: 27.10 },
128     { nome: 'lapis', qtde: 3, preco: 5.82},
129     { nome: 'tesoura', qtde: 1, preco: 19.20 },
130 ]
131
132 const getNome3 = item => item.nome
133 const qtdeMaiorQueZero2 = item => item.qtde > 0
134 const qtdeMaiorIgualZero2 = item => item.qtde >= 0
135 const qtdeMuitoGrande2 = item => item.qtde > 0
136
137 const nomeItensValidos = carrinho2
138     .filter(qtdeMaiorIgualZero2)
139     .map(getNome3)
140 console.log(nomeItensValidos)
141
142
143
144
145 /*****************************************************************/
146 /** DESAFIO: Pega quantidade maior que zero e nomes dos itens **/
147 /*****************************************************************/
148 Array.prototype.meuFilter = function(fn){
149     const novoArray = []
150
151     for(let i=0; i < this.length; i++ ){
152         if(fn(this[i], i, this)){       // Testa se funçao é true (fn(this[i], i,
    this))
153             novoArray.push (this[i])
154         }
155     }
156     return novoArray
157 }
158 const nomeItensValidos2 = carrinho2
159     .meuFilter(qtdeMaiorIgualZero2)
160     .map(getNome3)
161 console.log(nomeItensValidos2)
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
```

```
/*********************************************************************/
/******************** Callback 5 - Usando REDUCE *********************/
/*********************************************************************/
const { get } = require("http")

const carrinho3 = [
    { nome: 'caneta', qtde: 10, preco: 7.99 },
    { nome: 'impressora', qtde: 0, preco: 649.50 },
    { nome: 'caderno', qtde: 4, preco: 27.10 },
    { nome: 'lapis', qtde: 3, preco: 5.82},
    { nome: 'tesoura', qtde: 1, preco: 19.20 },
]

const getTotal3 = item => item.qtde * item.preco
const somar3 = (acc,el) => acc + el

//const totais = carrinho.map(totalGeral)

//console.log(totais)

const totalGeral3 = carrinho
    .map(getTotal3)
    .reduce(somar3)

    console.log(totalGeral3)



/*********************************************************/
/************* Implementação de meu reduce **************/
/*********************************************************/

    Array.prototype.meuReduce = function(fn, inicial){
        let acc = inicial
        for(let i = 0; i < this.length; i++){
            if(!acc && i === 0){            // para verificar se acumulador tem valor inicial
                acc = this[i]               // se não for definido, primeiro elemento será valor inicial
                continue
            }
            acc = fn(acc, this[i], i, this)
        }
        return acc
    }

    const totalGeral2 = carrinho3
    .map(getTotal3)
    .meuReduce(somar3)

    console.log(totalGeral2)
```