**Candidato: Marco Antônio da Silva Júnior**

Primeiramente, optei por tratar os livros como objetos, assim ficaria mais fácil de manipulá-los posteriormente.

const livro1 = {

Titulo: 'Java How To Program',

Author: 'Deitel & Deitel',

EditionYear: 2007

}

const livro2 = {

Titulo: 'Patterns of Enterprise Application Architecture',

Author: 'Martin Fowler',

EditionYear: 2002

}

const livro3 = {

Titulo: 'Head First Design Patterns',

Author: 'Elisabeth Freeman',

EditionYear: 2004

}

const livro4 = {

Titulo: 'Internet & World Wide Web: How to Program',

Author: 'Deitel & Deitel',

EditionYear: 2007

Depois fiz as funções de ordenar ascendente e descendente passando o próprio array de objetos, bem como o atributo a ser ordenado, utilizando ternários e a função “.sort()” que facilitou a ordenação.

function ordem\_asc (*array*, *atributo*) {

return array.sort((*elem1*, *elem2*) => {return elem1[atributo] < elem2[atributo] ? -1 : elem1[atributo] > elem2[atributo] ? 1 : 0})

}

function ordem\_desc (*array*, *atributo*) {

return array.sort((*elem1*, *elem2*) => {return elem1[atributo] < elem2[atributo] ? 1 : elem1[atributo] > elem2[atributo] ? -1 : 0})

Então foi cria a função “iguais” para verificar quais valores se repetem no atributo escolhido a fim de ordená-los em seguida.

function iguais(*array*, *atributo*) {

let repete = []

for (i = 0; i < array.length; i++){

let aux = [array[i]]

for (x = i; x < array.length; x++){

if (i != x && array[i][atributo] == array[x][atributo]) aux.push(array[x])

}

repete.push(aux)

}

return repete

}

Após, criei a função principal que se chama “ordenar” e recebe o array de objetos, juntamente com o arquivo de configuração (config.js). Optei por utilizar o localStorage porque como a aplicação é simples e não precisa de uma grande quantidade de dados a solução foi utilizar o necessário para cumprir o demandado.

const config = {

primeira\_ord: 'asc',

primeiro\_atrib: 'Author',

segunda\_ord: 'desc',

segundo\_atrib: 'Titulo',

terceira\_ord: '',

terceiro\_atrib: ''

}

localStorage.setItem('config', JSON.stringify(config))

A ordenação é feita em três etapas: a primeira chamando a função de ordenar correspondente.

A segunda verificando se o segundo atributo é igual e a terceira da mesma forma.

Caso não haja atributos iguais, passa para próxima etapa ou retorna o valor final.

function ordenar(*ordenados*) {

if (config.primeira\_ord = '') return ''

if (ordenados = null) throw 'OrderingException'

let listaIgual = ''

let listaIgual2 = ''

let filtrado = ''

let indice = 0

let primeira = config.primeira\_ord == 'asc' ? ordem\_asc(ordenados, config.primeiro\_atrib) : ordem\_desc(ordenados, config.primeiro\_atrib)

if (config.segunda\_ord == '') return primeira

Utilizei a função “.filter” em algumas ocasiões para extrair valores de arrays, porquanto ela facilita a não precisar de loops para realizar as checagens e criação de novos arrays.

listaIgual = iguais(primeira, config.segundo\_atrib)

filtrado = listaIgual.filter((*valor*) => valor.length >= 2)