

Implementação de Hashing com JavaScript

by Caio A'Lima

```
// HASHING
// TAMANHO DA TABELA -> Idealmente não deve ser potencia de 2 e não
// deve ser primeiro de M(Tamanho da Tabela)

// Encadeamento Externo

// função hash
function h(numero, tamanho) {
    return numero % tamanho;
}

class TabelaHash {
    constructor(tamanho) {
        this.tamanho = tamanho;
        this.tabela = new Array(tamanho);
    }

    // Inserir Valor
    inserir(valor) {
        let novoPonteiro = new Ponteiro(valor);
        let index = h(valor, this.tamanho);
        if (!this.tabela[index]) {
            this.tabela[index] = novoPonteiro;
        } else {
            let atual = this.tabela[index];
            let aux = atual;
            while (aux.proximo !== null) {
                aux = aux.proximo;
            }
            aux.proximo = novoPonteiro;
        }
    }

    // Buscar Valor
    buscar(valor) {
        let index = h(valor, this.tamanho);
        let atual = this.tabela[index];
```

```

    if (atual) {
        if (atual.valor == valor) {
            return atual.valor;
        } else {
            // Tratamento de Colisão
            let aux = atual;
            while (aux != null) {
                if (aux.valor == valor) {
                    return aux.valor;
                }
                aux = aux.proximo;
            }
            return null;
        }
    } else {
        return null;
    }
}

remover(valor) {
    let index = h(valor, this.tamanho);
    let atual = this.tabela[index];

    if (atual) {
        if (atual.valor == valor && atual.proximo == null) {
            this.tabela[index] = null;
            return "Removido com sucesso.";
        } else {
            let anterior = atual;
            let aux = atual.proximo;
            while (aux) {
                if (atual.valor == valor) {
                    this.tabela[index] = atual.proximo;
                    return "Removido com sucesso.";
                }
            }

            if (aux.valor == valor) {
                anterior.proximo = aux.proximo;
                return "REMOVIDO COM SUCESSO.";
            }
        }
    }
}

```

```

        }

        anterior = aux;
        aux = aux.proximo;
    }
    return "Não consegui remover";
}
} else {
    return "Não consegui remover";
}
}

imprimir(){
    console.log(" ***** ");
    this.tabela.forEach((lista,i) => {
        let linha = `INDICE ${i}: `;
        let atual = lista;
        while(atual){
            linha += `(${atual.valor}) ➤ `;
            atual = atual.proximo;
        }
        console.log(linha + " \n ");
    })
    console.log(" ***** ");
}
}

class Ponteiro {
    constructor(valor = "", proximo = null) {
        this.valor = valor;
        this.proximo = proximo;
    }
}

console.log(" *** TABELA HASH - ENCADEAMENTO EXTERNO *** ");
console.log("          by CAIO A'LIMA ");

const tabela = new TabelaHash(5);

tabela.inserir(10);
tabela.inserir(11);
tabela.inserir(12);

```

```
tabela.inserir(13);
tabela.inserir(15);
tabela.inserir(20);
tabela.inserir(30);
tabela.inserir(40);

//console.log("BUSCANDO: 40 -> " + tabela.buscar(40));

console.log(tabela.remover(40));

//console.log("BUSCANDO: 40 -> " + tabela.buscar(40));

tabela.imprimir()
```