

# Projeto Prático - Arquitetura de Computadores

## Grupo:

Riquelme Gomes      DRE: 120032785

Vitória Serafim      DRE: 120035369

O desenvolvimento desse projeto baseia-se na implementação de três algoritmos de ordenação em Linguagem de programação C, onde “traduziremos” esses códigos para MIPS usando o simulador MARS MIPS simulator - Missouri State University.

## 1º algoritmo: Insertion Sort (Ordenação por inserção)

### Código em C:

```
void insertionSort(int A[], int tamanho){
    int i, j, eleito;

    for(i = 1; i < tamanho; i++){
        eleito = A[i];
        j = i-1;

        while((j >= 0) && (eleito < A[j])){
            A[j+1] = A[j];
            j --;
        }
        A[j+1] = eleito;
    }
}
```

### Mapeamento (MIPS):

Função insertionSort:

End base de A (array)	\$a0
-----------------------	------

tamanho	\$a1
i (índice)	\$t0
j (índice)	\$t1
eleito (variável)	\$t2

## 2º algoritmo: Selection Sort (Ordenação por seleção)

### Código em C:

2 funções: (swap e selectionSort)

```
void swap(int v[], int i, int j) {
    int temp;
    temp = v[i];
    v[i] = v[j];
    v[j] = temp;
}

void selectionSort(int v[], int n) {
    int i, j, indiceMenor;

    for (int i = 0; i < n-1; i++) {
        indiceMenor = i;

        for (int j = i + 1; j < n; j++) {
            if(v[j] < v[indiceMenor]) {
                indiceMenor = j;
            }
        }
        swap(v, i, indiceMenor);
    }
}
```

### Mapeamento (MIPS):

Função Swap:

End base de v	\$a0
i	\$a1
j	\$a2

Função selectionSort:

End base de v (array)	\$a0 / \$s3
Tamanho do array (n)	\$a1 / \$s4
i (índice)	\$s0

j (índice)	\$s1
indiceMenor (variável)	\$s2

## 3º algoritmo: Bubble Sort (Ordenação por flutuação)

### Código em C:

2 funções: (swap e bSort)

```
void swap(int v[], int k) {
    int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}

void bSort(int v[], int n) {
    int i, j;

    for(i = 0; i < n; i++) {
        for(j = i-1; j >= 0 && v[j] > v[j+1]; j--) {
            swap(v, j);
        }
    }
}
```

### Mapeamento (MIPS):

Função Swap:

End base de v (Array)	\$a0
k	\$a1
temp	\$t0

Função BubbleSort:

End base de v (Array)	\$a0 / \$s2
n (tamanho do array)	\$a1 / \$s3
i (índice)	\$s0
j (índice)	\$s1