

# Micro Atividade 3: Descrever a utilização do algoritmo de ordenação “Selection Sort” em Python

//Disciplina: Nivel5: Colocando tudo em ordem e guardando

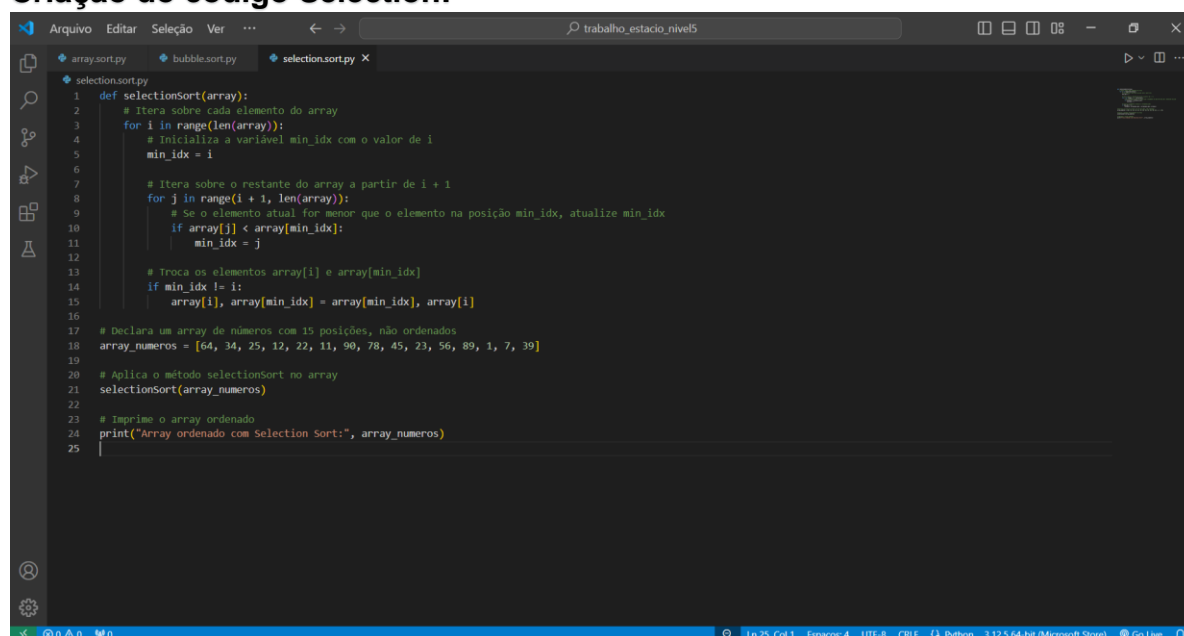
//Professor: Robson Lorbieski

//Aluno: Rafael Lima de Medeiros

//Turma: 2024.2

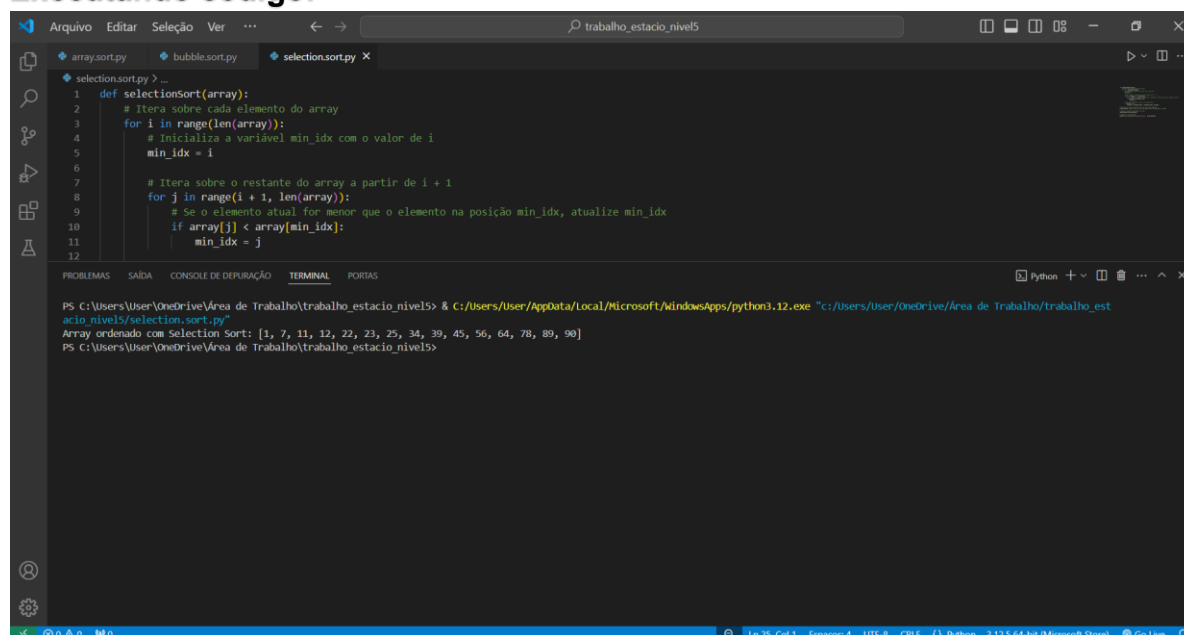
//Data atual: 25/08/2024

## Criação do código Selection:



```
1 def selectionSort(array):
2     # Itera sobre cada elemento do array
3     for i in range(len(array)):
4         # Inicializa a variável min_idx com o valor de i
5         min_idx = i
6
7         # Itera sobre o restante do array a partir de i + 1
8         for j in range(i + 1, len(array)):
9             # Se o elemento atual for menor que o elemento na posição min_idx, atualize min_idx
10            if array[j] < array[min_idx]:
11                min_idx = j
12
13            # Troca os elementos array[i] e array[min_idx]
14            if min_idx != i:
15                array[i], array[min_idx] = array[min_idx], array[i]
16
17 # Declara um array de números com 15 posições, não ordenados
18 array_numeros = [64, 34, 25, 12, 22, 11, 90, 78, 45, 23, 56, 89, 1, 7, 39]
19
20 # Aplica o método selectionSort no array
21 selectionSort(array_numeros)
22
23 # Imprime o array ordenado
24 print("Array ordenado com Selection Sort:", array_numeros)
25
```

## Executando código:



```
PS C:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel5> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.12.exe "c:/Users/User/OneDrive/Área de Trabalho/trabalho_estacio_nivel5/selection_sort.py"
Array ordenado com Selection Sort: [1, 7, 11, 12, 22, 23, 25, 34, 39, 45, 56, 64, 78, 89, 90]
PS C:\Users\User\OneDrive\Área de Trabalho\trabalho_estacio_nivel5>
```

## Código Selection:

```
def selectionSort(array):
    # Itera sobre cada elemento do array
    for i in range(len(array)):
        # Inicializa a variável min_idx com o valor de i
        min_idx = i

        # Itera sobre o restante do array a partir de i + 1
        for j in range(i + 1, len(array)):
            # Se o elemento atual for menor que o elemento na posição min_idx, atualize min_idx
            if array[j] < array[min_idx]:
                min_idx = j

        # Troca os elementos array[i] e array[min_idx]
        if min_idx != i:
            array[i], array[min_idx] = array[min_idx], array[i]

# Declara um array de números com 15 posições, não ordenados
array_numeros = [64, 34, 25, 12, 22, 11, 90, 78, 45, 23, 56, 89, 1, 7, 39]

# Aplica o método selectionSort no array
selectionSort(array_numeros)

# Imprime o array ordenado
print("Array ordenado com Selection Sort:", array_numeros)
```