

# DESENVOLVIMENTO FULL STACK

## NÍVEL 5: Colocando Tudo Em Ordem e Guardando

---

### Microatividade-1:Descrever a ordenação de um array utilizando Python

#### M 1.1

```
1 import random
2
3 # Criando uma lista vazia para armazenar os números aleatórios
4 numeros_aleatorios = []
5
6 # Gerando 15 números aleatórios entre 1 e 100 e adicionando à lista
7 for _ in range(15):
8     numero = random.randint(1, 100)
9     numeros_aleatorios.append(numero)
10
11 # Ordenando a lista em ordem crescente
12 numeros_aleatorios.sort()
13
14 print(numeros_aleatorios)
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nivel5> & C:/Users/PC/OneDrive/Documentos/Códigos_VSCode/Estácio/Microatividade_nivel5/array.sort.py
[2, 60, 92, 98, 47, 76, 72, 11, 40, 50, 100, 96, 55, 17, 5]
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nivel5> & C:/Users/PC/OneDrive/Documentos/Códigos_VSCode/Estácio/Microatividade_nivel5/array.sort.py
[7, 9, 16, 18, 21, 23, 24, 33, 50, 62, 69, 69, 71, 78, 100]
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nivel5>
```

#### M 1.2

```
1 import random
2
3 # Criando uma lista vazia para armazenar os números aleatórios
4 numeros_aleatorios = []
5
6 # Gerando 15 números aleatórios entre 1 e 100 e adicionando à lista
7 for _ in range(15):
8     numero = random.randint(1, 100)
9     numeros_aleatorios.append(numero)
10
11 # Ordenando a lista em ordem decrescente
12 numeros_aleatorios.sort(reverse=True)
13
14 print(numeros_aleatorios)
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nível5>& C:/Users/PC/AppData
/PC/OneDrive/Documentos/Códigos_VSCode/Estácio/Microatividade_nível5/array.sort.py
[100, 95, 93, 89, 79, 77, 30, 25, 22, 20, 5, 5, 4, 4, 3]
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nível5>
```

## M1.3

```
1 # Criando o array de strings
2 dados = [
3     {
4         "Nome: Ana Clara",
5         "dataNascimento: 20/01/1990",
6         "CPF: 088.888.888-88",
7         "RG: 9.584.147"
8     },
9     {
10        "Nome: Marcelo Alves",
11        "dataNascimento: 20/06/1998",
12        "CPF: 456.845.785-00",
13        "RG: 7.547.156"
14    },
15 },
16 {
17     "Nome: Bruna Silva",
18     "dataNascimento: 07/07/1995",
19     "CPF: 123.874.987-02",
20     "RG: 2.514.789"
21 }
22 ]
23
24 # Ordenando de forma decrescente
25 dados.sort(key=None, reverse=True)
26
27 # Exibindo o resultado
28 print(dados)
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS Python + ...
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nível5> & C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/Users
/PC/OneDrive/Documentos/Códigos_VSCode/Estácio/Microatividade_nível5/array.sort.py
[{'CPF: 088.888.888-88', 'dataNascimento: 20/01/1990', 'RG: 9.584.147', 'Nome: Ana Clara'}, {'RG: 7.547.156', 'dataNascimento: 20/06/1998', 'Nome: Marcelo
Alves', 'CPF: 456.845.785-00'}, {'dataNascimento: 07/07/1995', 'CPF: 123.874.987-02', 'RG: 2.514.789', 'Nome: Bruna Silva'}]
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nível5>
```

## Microatividade-2: Descrever a utilização do algoritmo de ordenação “Buble Sort” em Python

### M 2.1

```
1  def bubbleSort(array):
2      # Itera sobre todos os elementos do array
3      for i in range(len(array)):
4          # Itera sobre os elementos restantes
5          for j in range(0, len(array) - i - 1):
6              # Compara os elementos adjacentes
7              if array[j] > array[j + 1]:
8                  # Troca os elementos se estiverem na ordem errada
9                  temp = array[j]
10                 array[j] = array[j + 1]
11                 array[j + 1] = temp
12
13     # Declara um array com 15 números
14     numeros = [64, 34, 25, 12, 22, 11, 90, 42, 56, 78, 23, 45, 67, 89, 10]
15
16     # Aplica o método bubbleSort
17     bubbleSort(numeros)
18
19     # Imprime o array ordenado
20     print("Array ordenado:")
21     print(numeros)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nivel5> & C:/Users/PC/AppData/Local/Microsoft/Windows/PC/OneDrive/Documentos/Códigos_VSCode/Estácio/Microatividade_nivel5/bubble.sort.py
Array ordenado:
[10, 11, 12, 22, 23, 25, 34, 42, 45, 56, 64, 67, 78, 89, 90]
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nivel5>
```

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

### M 3.1

```
1 import random
2
3 array = [random.randint(1, 100) for _ in range(15)]
4
5 for i in range(len(array)):
6     c = i
7
8     for j in range(i + 1, len(array)):
9         if array[c] > array[j]:
10             c = j
11
12     if c != i: # Verifica se houve troca
13         array[i], array[c] = array[c], array[i]
14
15 print(array)
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nivel5>
& C:/Users/PC/AppData/Local/Microsoft/WindowsApps/python3.12.exe c:/Users/PC/OneDrive/Documentos/Códigos_VSCode/Estácio/M
ort.py
[9, 14, 20, 27, 33, 37, 53, 57, 66, 70, 77, 80, 89, 93, 94]
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nivel5> & C:/Users/PC/AppData/Local/Microsoft/Wi
s/PC/OneDrive/Documentos/Códigos_VSCode/Estácio/Microatividade_nivel5/selection.sort.py
[4, 9, 21, 27, 33, 43, 48, 69, 69, 72, 77, 81, 89, 91, 96]
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nivel5> & C:/Users/PC/AppData/Local/Microsoft/Wi
s/PC/OneDrive/Documentos/Códigos_VSCode/Estácio/Microatividade_nivel5/selection.sort.py
[12, 12, 24, 30, 38, 44, 46, 54, 60, 67, 72, 78, 88, 89, 94]
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nivel5> & C:/Users/PC/AppData/Local/Microsoft/Wi
s/PC/OneDrive/Documentos/Códigos_VSCode/Estácio/Microatividade_nivel5/selection.sort.py
[25, 28, 35, 37, 41, 51, 58, 59, 61, 68, 68, 76, 86, 87, 89]
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nivel5>
```

## Microatividade 4: Descreva a leitura de dados a partir de um arquivo externo em Python

### M 4.1

```
1 with open("loremipsum.txt", "r") as arquivo:
2     conteudo = arquivo.read()
3
4 # Imprime na tela todo o conteúdo da variável
5 print("Conteúdo do arquivo:")
6 print(conteudo)
7
8 # Imprime na tela apenas a primeira linha do arquivo txt
9 primeira_linha = conteudo.splitlines()[0]
10 print("\nPrimeira linha do arquivo:")
11 print(primeira_linha)
12
13 # Imprime na tela apenas os 3 primeiros caracteres do arquivo txt
14 tres_primeiros_caracteres = conteudo[:3]
15 print("\nTrês primeiros caracteres do arquivo:")
16 print(tres_primeiros_caracteres)
17
18 # Abrir o arquivo text novamente
19 with open("loremipsum.txt", "r") as arquivo:
20     conteudo_novo = arquivo.read()
21     print("\nConteúdo do arquivo usando 'with':")
22     print(conteudo_novo)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nível5> C:/Users/PC/AppData/Local/
/PC/OneDrive/Documentos/Códigos_VSCode/Estácio/Microatividade_nível5/ler.txt.py
Conteúdo do arquivo:
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi
ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum.

Primeira linha do arquivo:
Lorem ipsum dolor sit amet, consectetur adipiscing elit,

Três primeiros caracteres do arquivo:
Lor

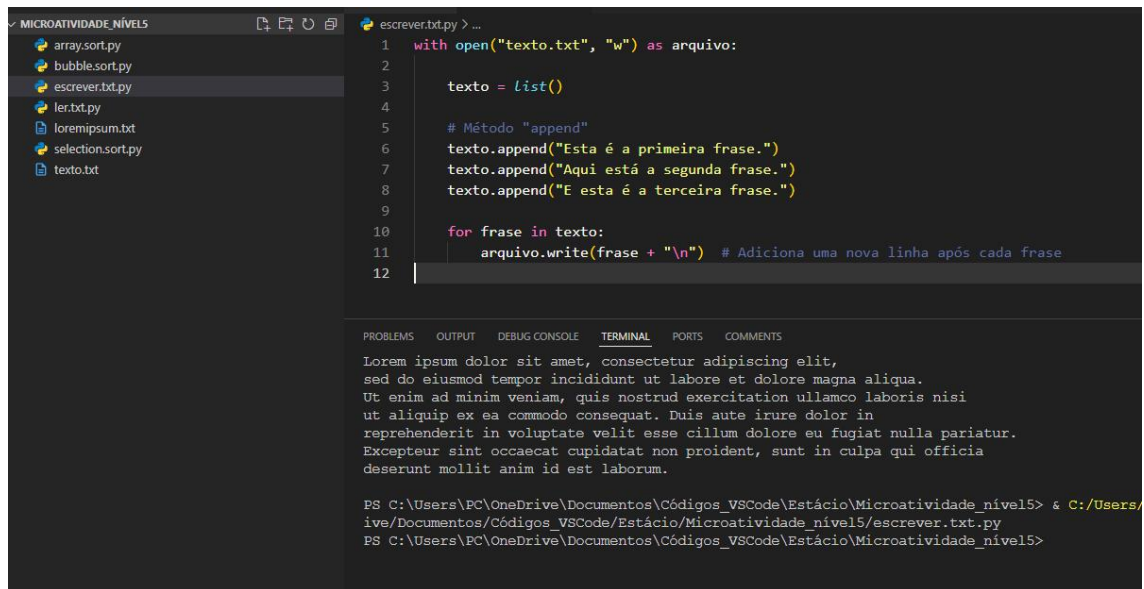
Conteúdo do arquivo usando 'with':
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi
ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum.

PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nível5>
```

## Microatividade 5: Descrevendo a escrita de dados em um arquivo externo em Python

### M5.1

```
1 with open("texto.txt", "w") as arquivo:
2
3     texto = list()
4
5     # Método "append"
6     texto.append("Esta é a primeira frase.")
7     texto.append("Aqui está a segunda frase.")
8     texto.append("E esta é a terceira frase.")
9
10    for frase in texto:
11        arquivo.write(frase + "\n") # Adiciona uma nova linha após cada frase
12
```



The screenshot shows a VS Code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a folder named 'MICROATIVIDADE\_NÍVEL5' containing several files: 'array.sort.py', 'bubble.sort.py', 'escrever.txt.py', 'ler.txt.py', 'loremipsum.txt', 'selection.sort.py', and 'texto.txt'. The code editor shows the same Python script as in the previous block, with line numbers 1 through 12. Below the code editor, there is a terminal window with the following output:

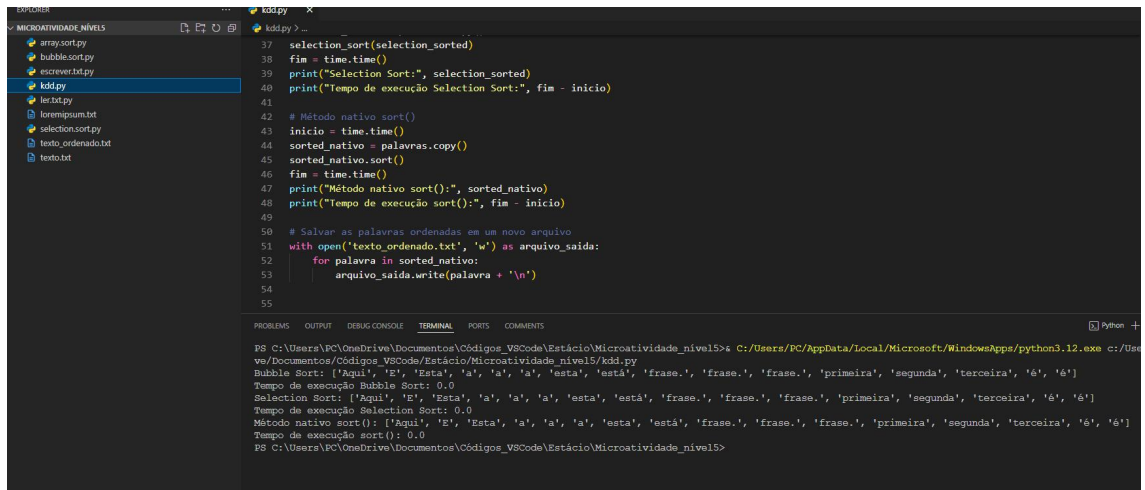
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi
ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum.

PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nível5> & C:/Users/
ive/Documentos/Códigos_VSCode/Estácio/Microatividade_nível5/escrever.txt.py
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nível5>
```



## Missão Prática | Colocando Tudo em Ordem e Guardando

```
1 import time
2
3 palavras = list()
4
5 # Ler arquivo
6 with open('texto.txt', 'r') as arquivo:
7     for linha in arquivo:
8         palavras.extend(linha.split())
9
10 def bubble_sort(lista):
11     n = len(lista)
12     for i in range(n):
13         for j in range(0, n-i-1):
14             if lista[j] > lista[j+1]:
15                 lista[j], lista[j+1] = lista[j+1], lista[j]
16
17 def selection_sort(lista):
18     n = len(lista)
19     for i in range(n):
20         min_idx = i
21         for j in range(i+1, n):
22             if lista[j] < lista[min_idx]:
23                 min_idx = j
24         lista[i], lista[min_idx] = lista[min_idx], lista[i]
25
26 # Usando Bubble Sort
27 inicio = time.time()
28 bubble_sorted = palavras.copy()
29 bubble_sort(bubble_sorted)
30 fim = time.time()
31 print("Bubble Sort:", bubble_sorted)
32 print("Tempo de execução Bubble Sort:", fim - inicio)
33
34 # Selection Sort
35 inicio = time.time()
36 selection_sorted = palavras.copy()
37 selection_sort(selection_sorted)
38 fim = time.time()
39 print("Selection Sort:", selection_sorted)
40 print("Tempo de execução Selection Sort:", fim - inicio)
41
42 # Método nativo sort()
43 inicio = time.time()
44 sorted_nativo = palavras.copy()
45 sorted_nativo.sort()
46 fim = time.time()
47 print("Método nativo sort():", sorted_nativo)
48 print("Tempo de execução sort():", fim - inicio)
49
50 # Salvar as palavras ordenadas em um novo arquivo
51 with open('texto_ordenado.txt', 'w') as arquivo_saida:
52     for palavra in sorted_nativo:
53         arquivo_saida.write(palavra + '\n')
54
```



The image shows a Visual Studio Code editor window with a file explorer on the left and a code editor in the center. The file explorer shows a folder named 'MICROATIVIDADE NÍVELS' containing several files: 'array.sort.py', 'bubbles.sort.py', 'escrever.txt.py', 'kdd.py' (selected), 'ler.txt.py', 'loremipsum.txt', 'selection.sort.py', 'texto\_ordenado.txt', and 'texto.txt'. The code editor displays the content of 'kdd.py', which is a Python script for sorting words. The script includes a function 'selection\_sort' and a main block that reads words from 'texto.txt', sorts them using 'selection\_sort', and writes the sorted words to 'texto\_ordenado.txt'.

```
37 selection_sort(selection_sorted)
38 fim = time.time()
39 print("Selection Sort:", selection_sorted)
40 print("Tempo de execução Selection Sort:", fim - inicio)
41
42 # Método nativo sort()
43 inicio = time.time()
44 sorted_nativo = palavras.copy()
45 sorted_nativo.sort()
46 fim = time.time()
47 print("Método nativo sort():", sorted_nativo)
48 print("Tempo de execução sort():", fim - inicio)
49
50 # Salvar as palavras ordenadas em um novo arquivo
51 with open('texto_ordenado.txt', 'w') as arquivo_saida:
52     for palavra in sorted_nativo:
53         arquivo_saida.write(palavra + '\n')
54
55
```

The terminal window at the bottom shows the output of the script execution. It displays the sorted words and the execution time for both the 'selection\_sort' and 'sort()' methods. The output is as follows:

```
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nível5> python3.12.exe c:/Users/PC/OneDrive/Documentos/Códigos_VSCode/Estácio/Microatividade_nível5/kdd.py
Bubble Sort: ['Aqui', 'E', 'Esta', 'a', 'a', 'a', 'esta', 'está', 'frase.', 'frase.', 'frase.', 'primeira', 'segunda', 'terceira', '6', '6']
Tempo de execução Bubble Sort: 0.0
Selection Sort: ['Aqui', 'E', 'Esta', 'a', 'a', 'a', 'esta', 'está', 'frase.', 'frase.', 'frase.', 'primeira', 'segunda', 'terceira', '6', '6']
Tempo de execução Selection Sort: 0.0
Método nativo sort(): ['Aqui', 'E', 'Esta', 'a', 'a', 'a', 'esta', 'está', 'frase.', 'frase.', 'frase.', 'primeira', 'segunda', 'terceira', '6', '6']
Tempo de execução sort(): 0.0
PS C:\Users\PC\OneDrive\Documentos\Códigos_VSCode\Estácio\Microatividade_nível5>
```