

# Nível 5: Colocando Tudo em Ordem e Guardando

## Microatividades

### Missão prática - Código final

```
# Missão Prática | Colocando tudo em
# ordem e guardando 🟩

import time

def bubbleSort(array):
    lista = len(array)
    for i in range(lista):
        for j in range(0, lista - i - 1):
            if array[j] > array[j + 1]:
                array[j], array[j + 1] = array[j + 1], array[j]

def selectionSort(array):
    for i in range(len(array)):
        menor_index = i
        for j in range(i + 1, len(array)):
            if array[j] < array[menor_index]:
                menor_index = j
        array[i], array[menor_index] = array[menor_index], array[i]

def processamento_arquivo(arquivo):
    palavras = []
    with open(arquivo, 'r', encoding='utf-8') as file:
        for linha in file:
            palavras.extend(linha.split())
    return palavras

entrada_arquivo = 'trabalho.txt'

palavras = processamento_arquivo(entrada_arquivo)

palavras_sort = palavras.copy()
start_time = time.time()
palavras_sort.sort()
end_time = time.time()
print("Método de ordenação sort():")
print(palavras_sort)
print(f"Tempo da execução: {end_time - start_time:.6f} segundos")
print()

novo_arquivo = 'palavras_ordenadas.txt'
with open(novo_arquivo, 'w', encoding='utf-8') as file:
    for palavra in palavras_sort:
        file.write(palavra + '\n')

print(f"O arquivo com as palavras em ordem foi criado com sucesso: {novo_arquivo}")
```

### Missão prática - código completo

```
# Missão Prática | Colocando tudo em
# ordem e guardando 🟩

import time

def bubbleSort(array):
    lista = len(array)
    for i in range(lista):
        for j in range(0, lista - i - 1):
            if array[j] > array[j + 1]:
                array[j], array[j + 1] = array[j + 1], array[j]

def selectionSort(array):
    for i in range(len(array)):
        menor_index = i
        for j in range(i + 1, len(array)):
            if array[j] < array[menor_index]:
                menor_index = j
        array[i], array[menor_index] = array[menor_index], array[i]

def processamento_arquivo(arquivo):
    palavras = []
    with open(arquivo, 'r', encoding='utf-8') as file:
        for linha in file:
            palavras.extend(linha.split())
    return palavras

entrada_arquivo = 'trabalho.txt'

palavras = processamento_arquivo(entrada_arquivo)

palavras_bubble_sort = palavras.copy()
start_time = time.time()
bubbleSort(palavras_bubble_sort)
end_time = time.time()
print("Método de ordenação Bubble Sort:")
print(palavras_bubble_sort)
print(f"Tempo de execução: {end_time - start_time:.6f} segundos")
print()

palavras_selection_sort = palavras.copy()
start_time = time.time()
selectionSort(palavras_selection_sort)
end_time = time.time()
print("Método de ordenação Selection Sort:")
print(palavras_selection_sort)
print(f"Tempo da execução: {end_time - start_time:.6f} segundos")
print()

palavras_sort = palavras.copy()
start_time = time.time()
palavras_sort.sort()
end_time = time.time()
print("Método de ordenação sort():")
print(palavras_sort)
print(f"Tempo da execução: {end_time - start_time:.6f} segundos")
print()

novo_arquivo = 'palavras_ordenadas.txt'
with open(novo_arquivo, 'w', encoding='utf-8') as file:
    for palavra in palavras_selection_sort:
        file.write(palavra + '\n')

print(f"O arquivo com as palavras em ordem foi criado com sucesso: {novo_arquivo}")
```