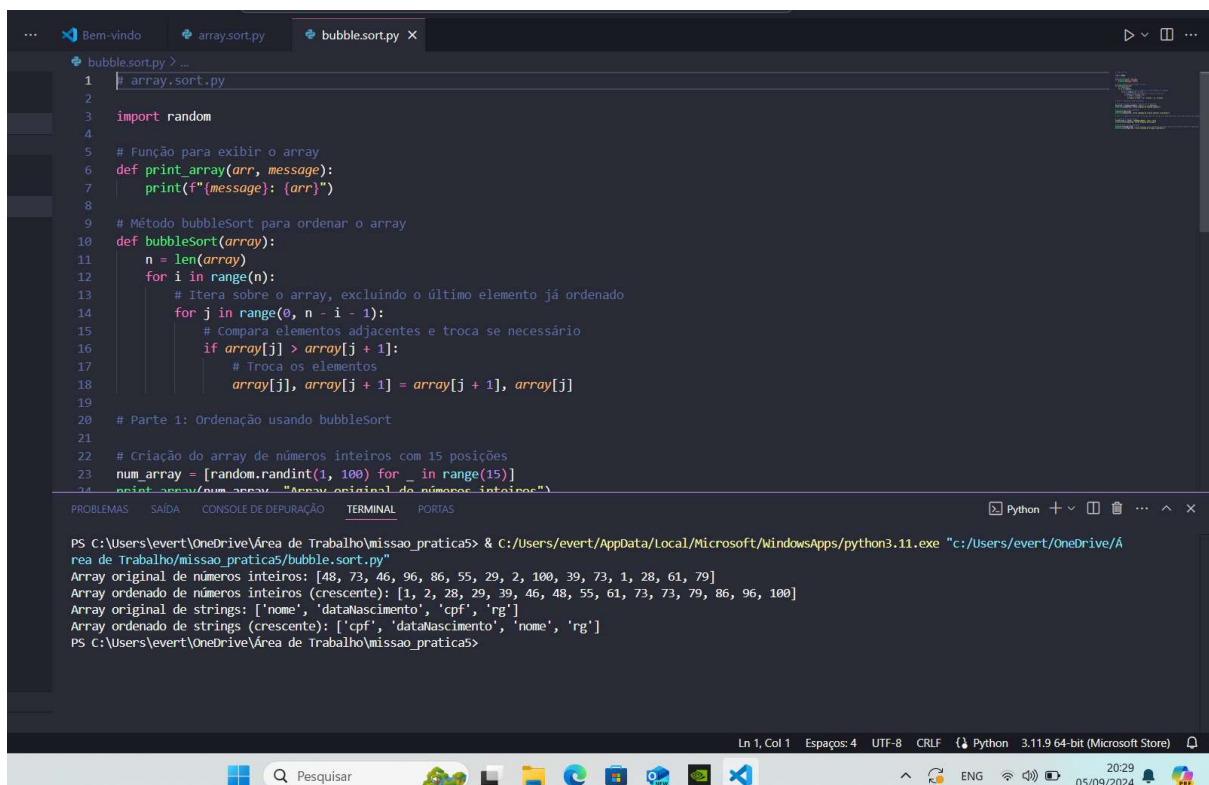


# Relatório Missão Prática

Colocando tudo em ordem e guardando:

O resultado dessa micro atividade é demonstrar como ordenar os valores de um array usando um método próprio da linguagem Python.



```
1 array.sort.py
2
3 import random
4
5 # Função para exibir o array
6 def print_array(arr, message):
7     print(f"{message}: {arr}")
8
9 # Método bubbleSort para ordenar o array
10 def bubbleSort(array):
11     n = len(array)
12     for i in range(n):
13         # Itera sobre o array, excluindo o último elemento já ordenado
14         for j in range(0, n - i - 1):
15             # Compara elementos adjacentes e troca se necessário
16             if array[j] > array[j + 1]:
17                 # Troca os elementos
18                 array[j], array[j + 1] = array[j + 1], array[j]
19
20 # Parte 1: Ordenação usando bubbleSort
21
22 # Criação do array de números inteiros com 15 posições
23 num_array = [random.randint(1, 100) for _ in range(15)]
24 print_array(num_array, "Array original de números inteiros")

```

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica5> & C:/Users/event/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho/missao\_pratica5/bubble.sort.py"

Array original de números inteiros: [48, 73, 46, 96, 86, 55, 29, 2, 100, 39, 73, 1, 28, 61, 79]

Array ordenado de números inteiros (crescente): [1, 2, 28, 29, 39, 46, 48, 55, 61, 73, 79, 86, 96, 100]

Array original de strings: ['nome', 'dataNascimento', 'cpf', 'rg']

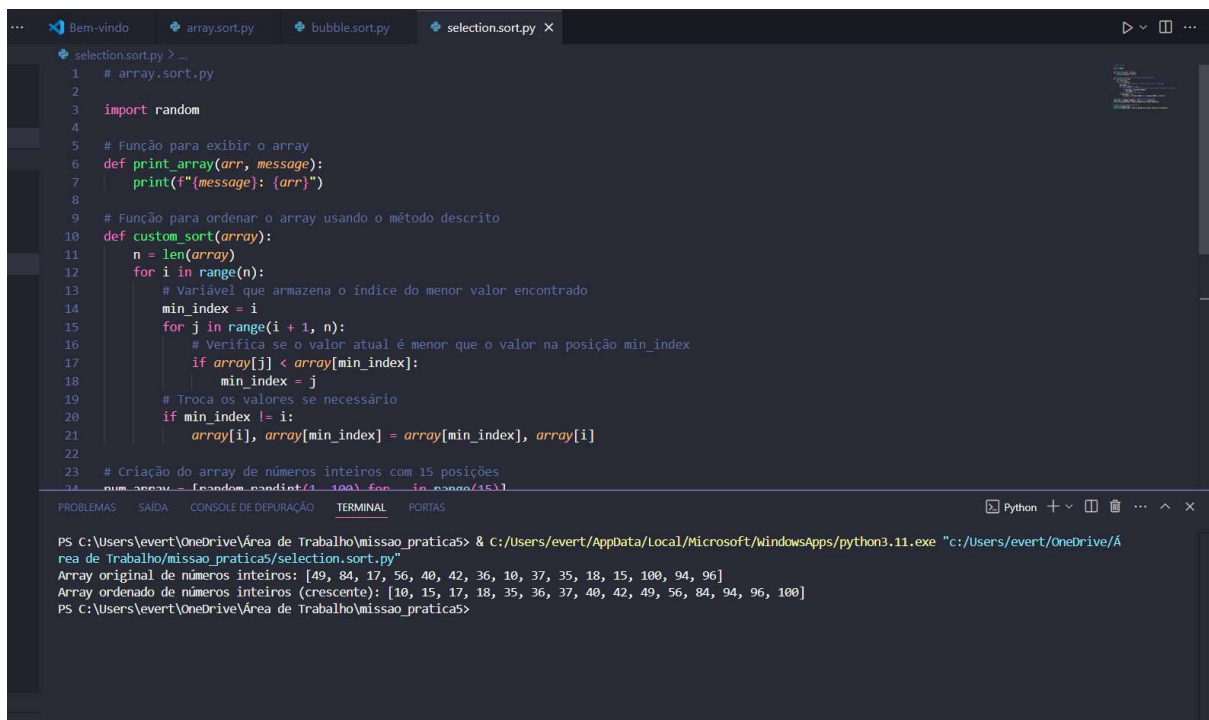
Array ordenado de strings (crescente): ['cpf', 'dataNascimento', 'nome', 'rg']

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica5>

Essa micro atividade verifica os conhecimentos do aluno relacionados a algoritmos de ordenação e permiti ao mesmo aplicar, de forma prática, o algoritmo Bubble Sort.

```
1 array.sort.py
2
3 import random
4
5 # Função para exibir o array
6 def print_array(arr, message):
7     print(f"{message}: {arr}")
8
9 # Método bubbleSort para ordenar o array
10 def bubbleSort(array):
11     n = len(array)
12     for i in range(n):
13         # Itera sobre o array, excluindo o último elemento já ordenado
14         for j in range(0, n - i - 1):
15             # Compara elementos adjacentes e troca se necessário
16             if array[j] > array[j + 1]:
17                 # Troca os elementos
18                 array[j], array[j + 1] = array[j + 1], array[j]
19
20 # Parte 1: Ordenação usando bubbleSort
21
22 # Criação do array de números inteiros com 15 posições
23 num_array = [random.randint(1, 100) for _ in range(15)]
24 print(f"Array original de números inteiros:")
25
26 PS C:\Users\event\OneDrive\Área de Trabalho\missao_pratica5> & C:\Users\event\AppData\Local\Microsoft\WindowsApps\python3.11.exe "C:\Users\event\OneDrive\Área de Trabalho\missao_pratica5\bubble.sort.py"
Array original de números inteiros: [48, 73, 46, 96, 86, 55, 29, 2, 100, 39, 73, 1, 28, 61, 79]
Array ordenado de números inteiros (crescente): [1, 2, 28, 29, 39, 46, 48, 55, 61, 73, 73, 79, 86, 96, 100]
Array original de strings: ['nome', 'dataNascimento', 'cpf', 'rg']
Array ordenado de strings (crescente): ['cpf', 'dataNascimento', 'nome', 'rg']
PS C:\Users\event\OneDrive\Área de Trabalho\missao_pratica5>
```

O resultado esperado dessa micro atividade é mostrar os algoritmos de ordenação e permitir ao mesmo aplicar, de forma prática, o algoritmo Selection Sort.



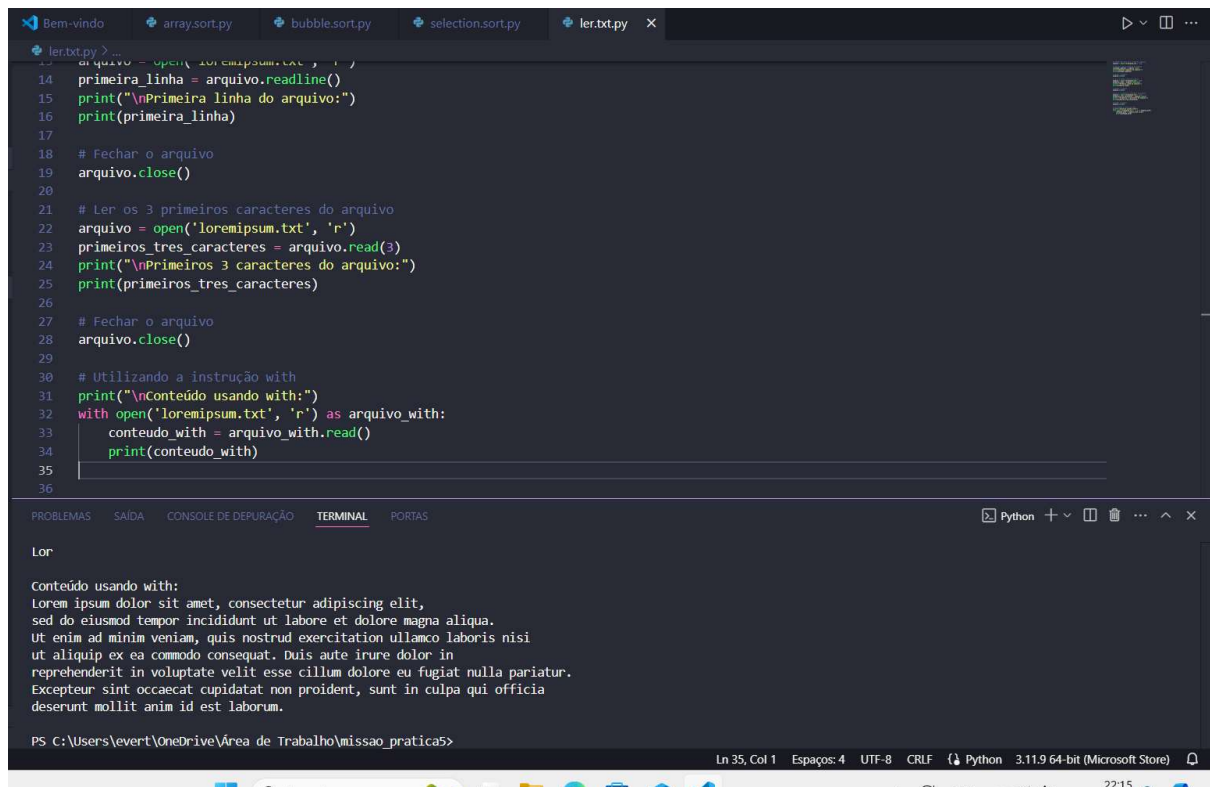
The screenshot shows a Python IDE with a file named `selection.sort.py` open. The code implements a Selection Sort algorithm. It includes a function `print_array` to display the array, a `custom_sort` function that finds the minimum element in the unsorted part of the array and swaps it with the element at the current index, and a main block that creates a random array of 15 integers and sorts it.

```
1 # array.sort.py
2
3 import random
4
5 # Função para exibir o array
6 def print_array(arr, message):
7     print(f"{message}: {arr}")
8
9 # Função para ordenar o array usando o método descrito
10 def custom_sort(array):
11     n = len(array)
12     for i in range(n):
13         # Variável que armazena o índice do menor valor encontrado
14         min_index = i
15         for j in range(i + 1, n):
16             # Verifica se o valor atual é menor que o valor na posição min_index
17             if array[j] < array[min_index]:
18                 min_index = j
19         # Troca os valores se necessário
20         if min_index != i:
21             array[i], array[min_index] = array[min_index], array[i]
22
23 # Criação do array de números inteiros com 15 posições
24 num_array = random.randint(1, 100) for _ in range(15)]
```

The terminal output shows the execution of the script:

```
PS C:\Users\event\OneDrive\Área de Trabalho\missao_pratica5> & C:/Users/event/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho\missao_pratica5/selection.sort.py"
Array original de números inteiros: [49, 84, 17, 56, 40, 42, 36, 10, 37, 35, 18, 15, 100, 94, 96]
Array ordenado de números inteiros (crescente): [10, 15, 17, 18, 35, 36, 37, 40, 42, 49, 56, 84, 94, 96, 100]
PS C:\Users\event\OneDrive\Área de Trabalho\missao_pratica5>
```

O resultado esperado dessa micro atividade é verificar a leitura de dados em arquivos utilizando Python.



```
13 arquivo = open('loremipsum.txt', 'r')
14 primeira_linha = arquivo.readline()
15 print("\nPrimeira linha do arquivo:")
16 print(primeira_linha)
17
18 # Fechar o arquivo
19 arquivo.close()
20
21 # Ler os 3 primeiros caracteres do arquivo
22 arquivo = open('loremipsum.txt', 'r')
23 primeiros_tres_caracteres = arquivo.read(3)
24 print("\nPrimeiros 3 caracteres do arquivo:")
25 print(primeiros_tres_caracteres)
26
27 # Fechar o arquivo
28 arquivo.close()
29
30 # Utilizando a instrução with
31 print("\nConteúdo usando with:")
32 with open('loremipsum.txt', 'r') as arquivo_with:
33     conteudo_with = arquivo_with.read()
34     print(conteudo_with)
35
36
```

PROBLEMAS SAÍDA CONSOLE DE DEPURACÃO **TERMINAL** PORTAS

Python + - [ ] ... ^ x

Lor

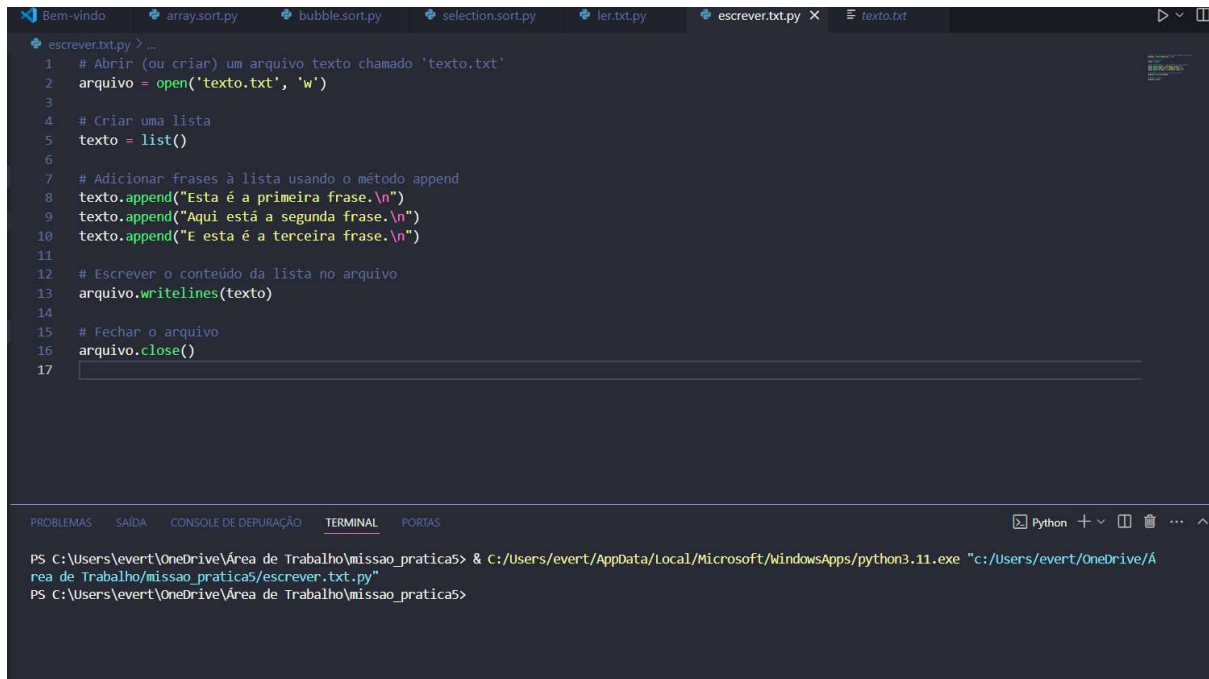
Conteúdo usando with:  
Lorem ipsum dolor sit amet, consectetur adipiscing elit,  
sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.  
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi  
ut aliquip ex ea commodo consequat. Duis aute irure dolor in  
reprehenderit 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\event\OneDrive\Área de Trabalho\missao\_pratica5>

Ln 35, Col 1 Espaços: 4 UTF-8 CRLF Python 3.11.9 64-bit (Microsoft Store)

22:15

O resultado esperado dessa micro atividade é verificar se o conhecimentos sobre a manipulação de dados em arquivos utilizando Python.

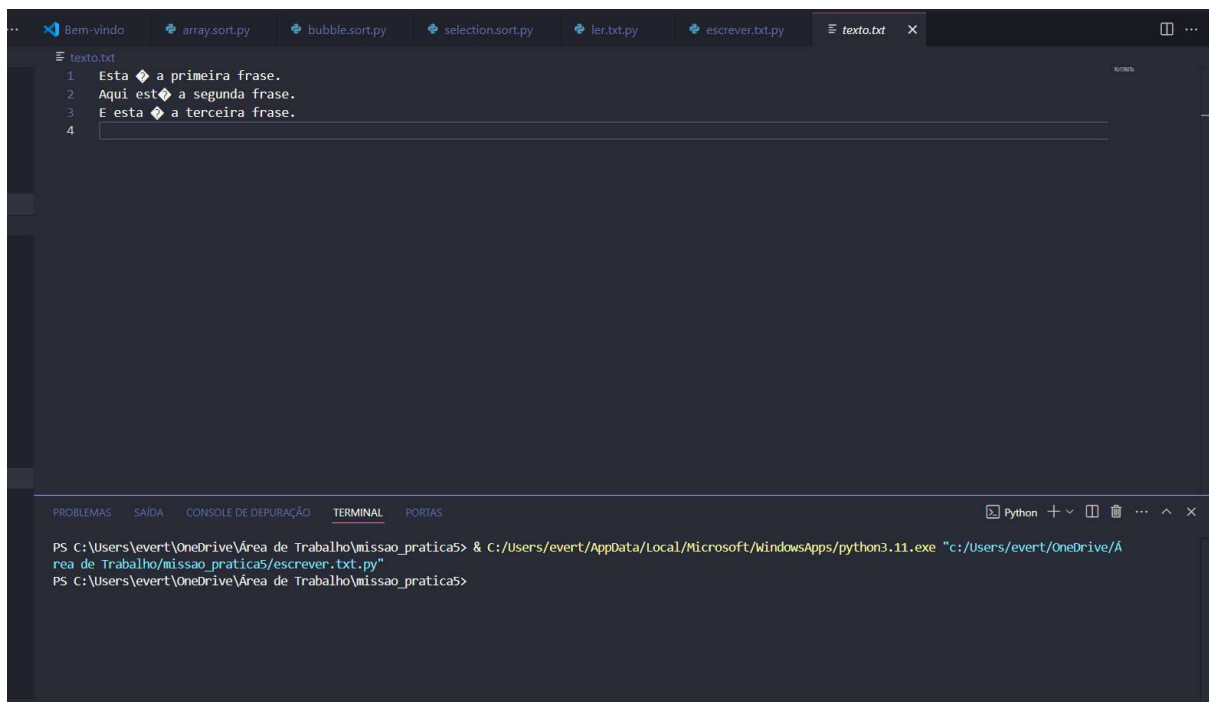


The screenshot shows a Python IDE with a file named `escrever.txt.py` open. The code in the editor is as follows:

```
1 # Abrir (ou criar) um arquivo texto chamado 'texto.txt'
2 arquivo = open('texto.txt', 'w')
3
4 # Criar uma lista
5 texto = list()
6
7 # Adicionar frases à lista usando o método append
8 texto.append("Esta é a primeira frase.\n")
9 texto.append("Aqui está a segunda frase.\n")
10 texto.append("E esta é a terceira frase.\n")
11
12 # Escrever o conteúdo da lista no arquivo
13 arquivo.writelines(texto)
14
15 # Fechar o arquivo
16 arquivo.close()
17
```

Below the editor, the terminal window shows the command to run the script and the resulting output:

```
PS C:\Users\event\OneDrive\Área de Trabalho\missao_pratica5> & C:/Users/event/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho/missao_pratica5/escrever.txt.py"
PS C:\Users\event\OneDrive\Área de Trabalho\missao_pratica5>
```



The screenshot shows the same Python IDE with a file named `texto.txt` open. The file contains the following text:

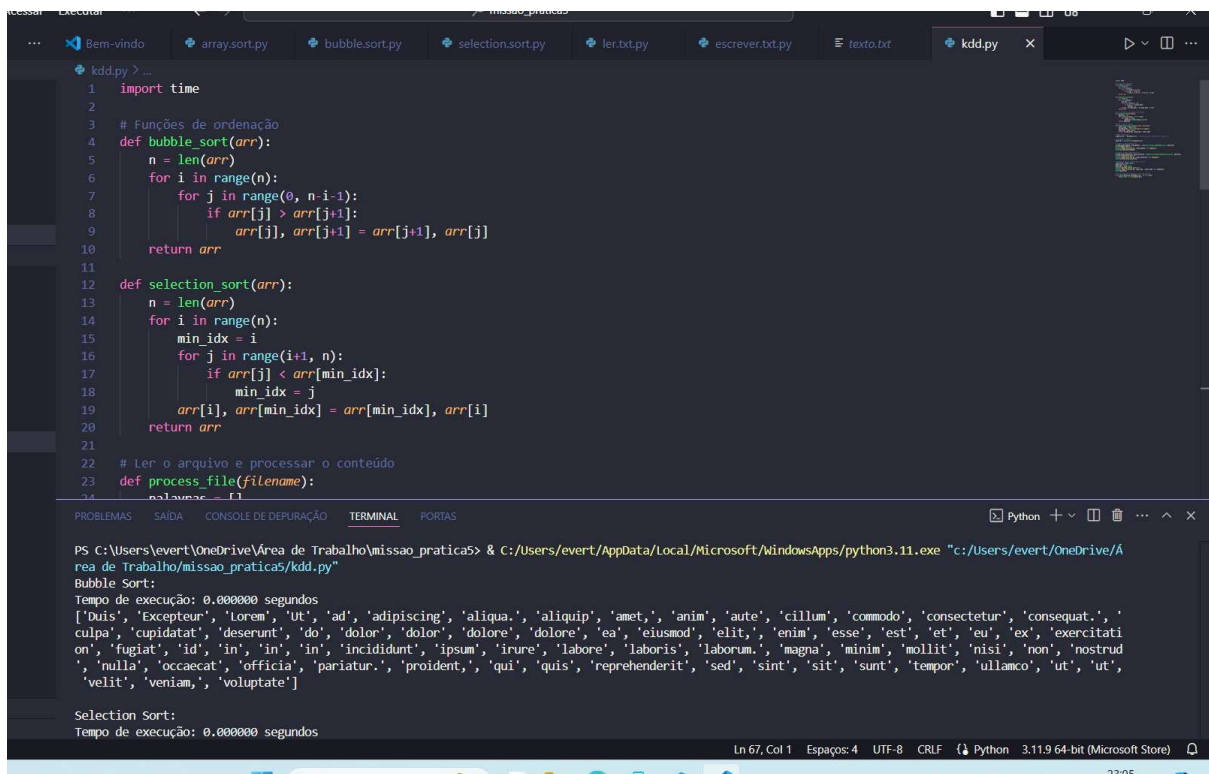
```
1 Esta é a primeira frase.
2 Aqui está a segunda frase.
3 E esta é a terceira frase.
4
```

Below the editor, the terminal window shows the command to run the script and the resulting output:

```
PS C:\Users\event\OneDrive\Área de Trabalho\missao_pratica5> & C:/Users/event/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho/missao_pratica5/escrever.txt.py"
PS C:\Users\event\OneDrive\Área de Trabalho\missao_pratica5>
```

# Missão Prática

## Colocando tudo em ordem e guardando



```
1 import time
2
3 # Funções de ordenação
4 def bubble_sort(arr):
5     n = len(arr)
6     for i in range(n):
7         for j in range(0, n-i-1):
8             if arr[j] > arr[j+1]:
9                 arr[j], arr[j+1] = arr[j+1], arr[j]
10    return arr
11
12 def selection_sort(arr):
13     n = len(arr)
14     for i in range(n):
15         min_idx = i
16         for j in range(i+1, n):
17             if arr[j] < arr[min_idx]:
18                 min_idx = j
19         arr[i], arr[min_idx] = arr[min_idx], arr[i]
20    return arr
21
22 # Ler o arquivo e processar o conteúdo
23 def process_file(filename):
24     palavras = []
25
26     with open(filename, 'r') as file:
27         for line in file:
28             words = line.split()
29             palavras.extend(words)
30
31     return palavras
32
33 # Executar o programa
34 if __name__ == '__main__':
35     palavras = process_file('palavras.txt')
36     palavras = bubble_sort(palavras)
37     palavras = selection_sort(palavras)
38     with open('palavras_ordenadas.txt', 'w') as file:
39         for word in palavras:
40             file.write(word + '\n')
```

PROBLEMAS SAÍDA CONSOLE DE DEBURAÇÃO TERMINAL PORTAS

Python + Python 3.11.9 64-bit (Microsoft Store)

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica5> & C:/Users/event/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho/missao\_pratica5/kdd.py"

Bubble Sort:  
Tempo de execução: 0.000000 segundos  
['Duis', 'Excepteur', 'lorem', 'ut', 'ad', 'adipiscing', 'aliqua', 'aliquip', 'amet', 'anim', 'aute', 'cillum', 'commodo', 'consectetur', 'consequat', 'culpa', 'cupidatat', 'deserunt', 'do', 'dolor', 'dolor', 'dolore', 'dolore', 'ea', 'eiusmod', 'elit', 'enim', 'esse', 'est', 'et', 'eu', 'ex', 'exercitatio', 'fugiat', 'id', 'in', 'in', 'in', 'incididunt', 'ipsum', 'irure', 'labore', 'laboris', 'laborum', 'magna', 'minim', 'mollit', 'nisi', 'non', 'nostrud', 'nulla', 'occaecat', 'officia', 'pariatur', 'proident', 'qui', 'quis', 'reprehenderit', 'sed', 'sint', 'sit', 'sunt', 'tempor', 'ullamco', 'ut', 'ut', 'velit', 'veniam', 'voluptate']

Selection Sort:  
Tempo de execução: 0.000000 segundos

Ln 67, Col 1 Espaços: 4 UTF-8 CRLF Python 3.11.9 64-bit (Microsoft Store)

Bem-vindoarray.sort.pybubble.sort.pyselection.sort.pyler.txt.pyescrever.txt.pykdd.py

palavras\_ordenadas.txt

1 Duis

2 Excepteur

3 Lorem

4 Ut

5 ad

6 adipiscing

7 aliqua.

8 aliquip

9 amet,

10 anim

11 aute

12 cillum

13 commodo

14 consectetur

15 consequat.

16 culpa

17 cupidatat

18 deserunt

19 do

20 dolor

21 dolor

22 dolore

23 dolore

24 ea

PROBLEMAS SAÍDA CONSOLE DE DEPURACÃO TERMINAL PORTAS

Python + - [ ] ... ^ x

PS C:\Users\event\OneDrive\Área de Trabalho\missao\_pratica5> & C:/Users/event/AppData/Local/Microsoft/WindowsApps/python3.11.exe "c:/Users/event/OneDrive/Área de Trabalho\missao\_pratica5/kdd.py"

Bubble Sort:

Tempo de execução: 0.000000 segundos

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

Selection Sort:

Tempo de execução: 0.000000 segundos

Ln 1, Col 1 Espaços: 4 UTF-8 CRLF Texto sem Formatação