

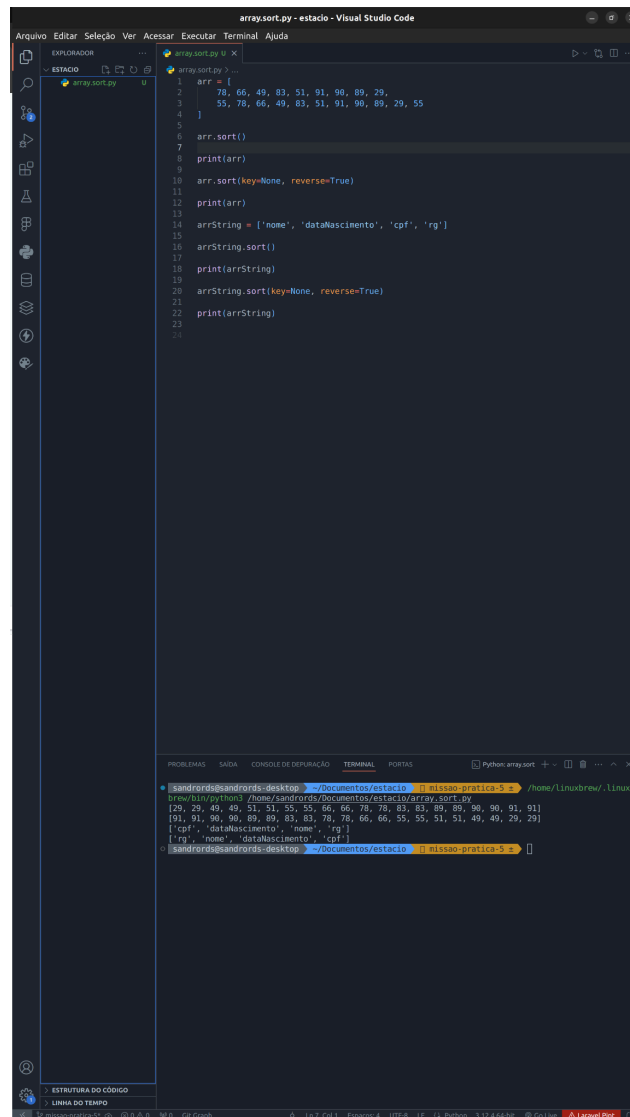
Missão Prática 5

Execução do Projeto

Configurações da Máquina

A atividade foi executada através de um desktop com o sistema operacional Ubuntu 22.04 LTS.

1. Microatividade 1

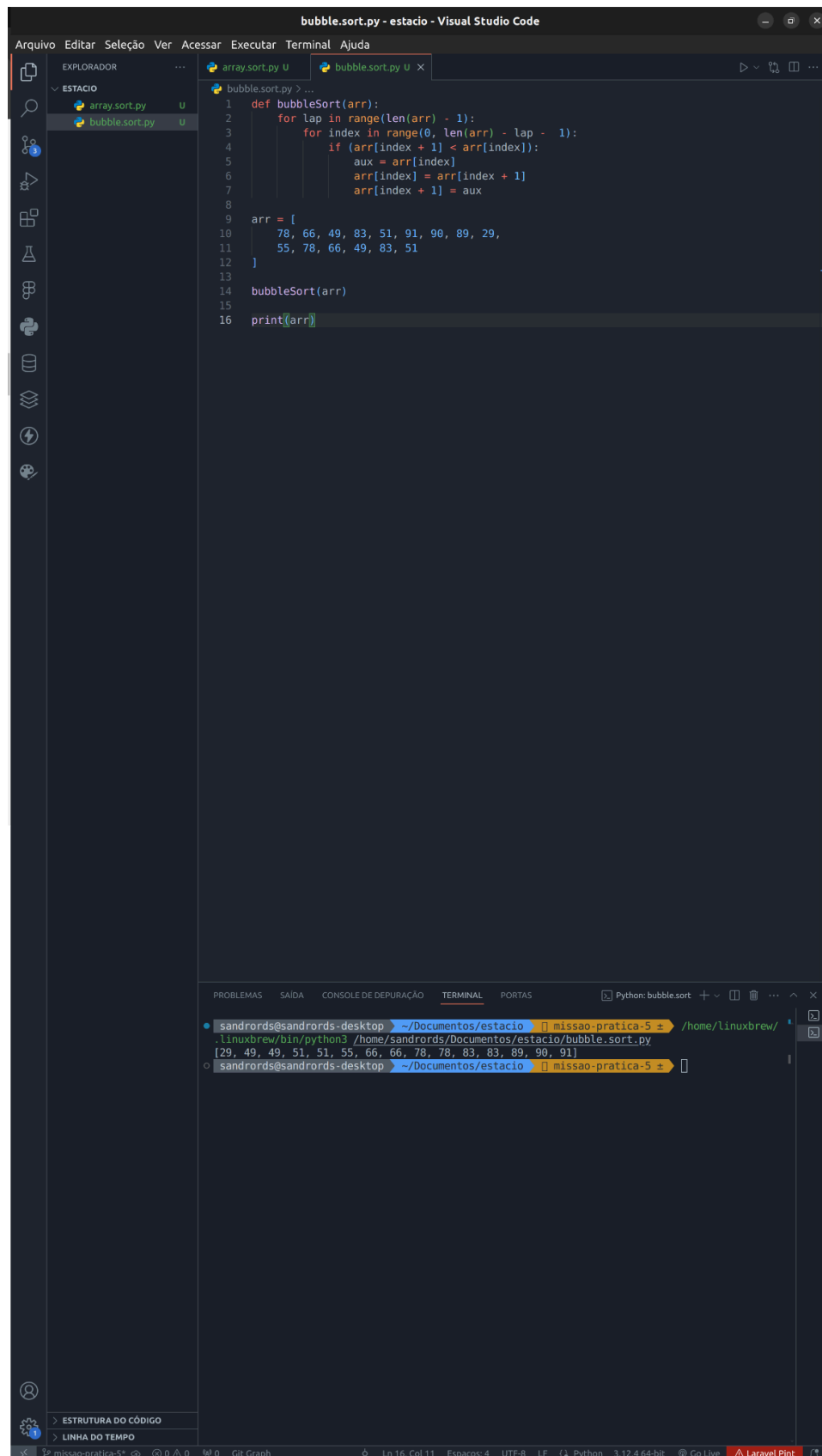


The screenshot shows the Visual Studio Code editor with a file named `array.sort.py` open. The code defines an array, sorts it, and prints the result. It also demonstrates sorting a dictionary-like structure. The terminal at the bottom shows the command `python3 array.sort.py` being executed, resulting in the following output:

```
[29, 20, 40, 49, 51, 51, 55, 55, 66, 66, 78, 78, 83, 83, 89, 89, 90, 91, 91]
{'cpf': 'dataNascimento', 'nome': 'rg'}
{'rg': 'nome', 'dataNascimento': 'cpf'}
```

Ordenando array com o método Sort.

2. Microatividade 2



The image shows a Visual Studio Code editor window titled "bubble.sort.py - estacio - Visual Studio Code". The editor is open to a file named "bubble.sort.py" which contains the following Python code:

```
1 def bubbleSort(arr):
2     for lap in range(len(arr) - 1):
3         for index in range(0, len(arr) - lap - 1):
4             if (arr[index + 1] < arr[index]):
5                 aux = arr[index]
6                 arr[index] = arr[index + 1]
7                 arr[index + 1] = aux
8
9 arr = [
10     78, 66, 49, 83, 51, 91, 90, 89, 29,
11     55, 78, 66, 49, 83, 51
12 ]
13
14 bubbleSort(arr)
15
16 print(arr)
```

The code implements a bubble sort algorithm. It defines a function `bubbleSort` that takes an array `arr` and sorts it in ascending order. The array is then printed.

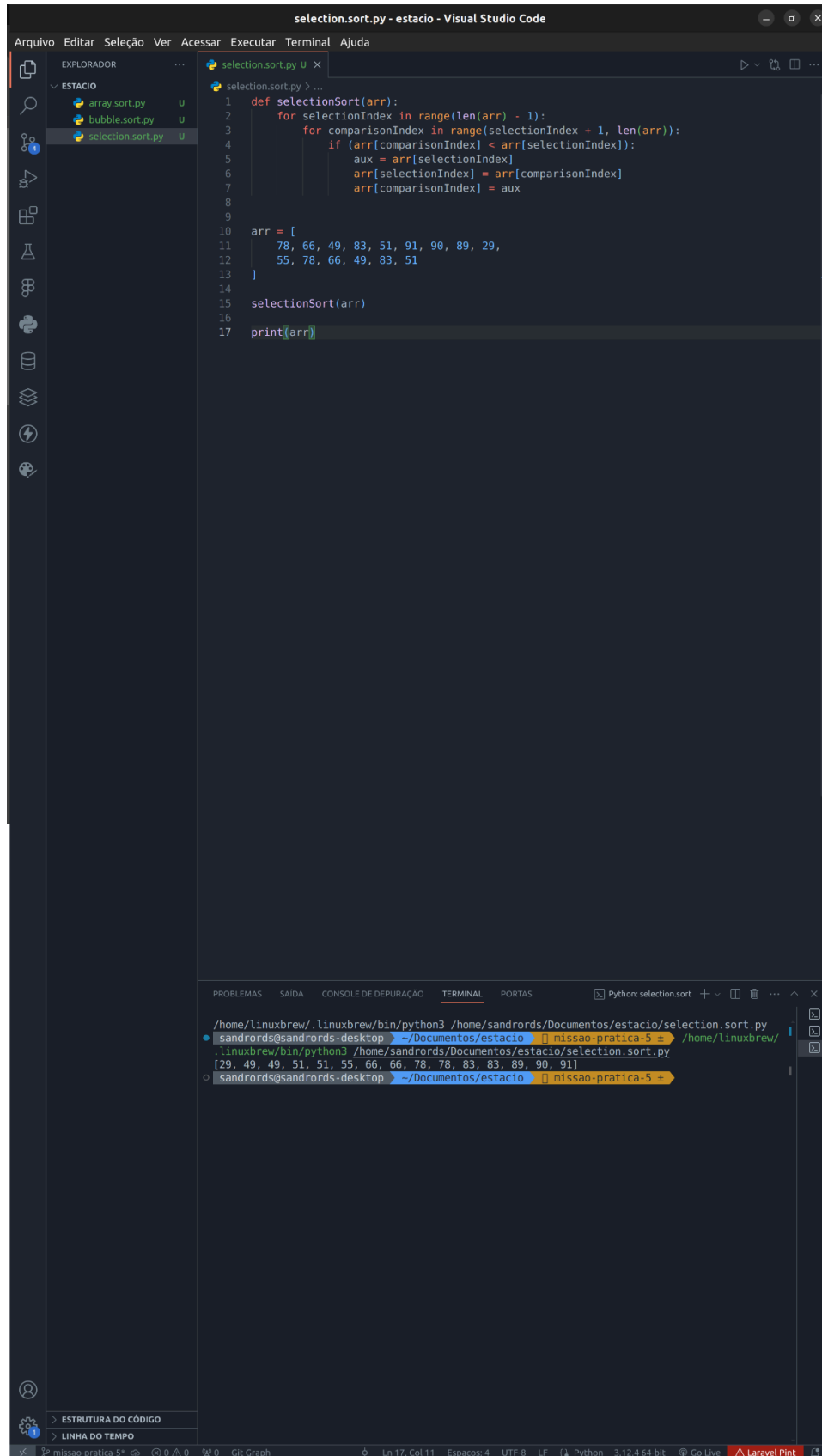
The terminal output shows the execution of the script:

```
sandrords@sandrords-desktop ~/Documentos/estacio missao-pratica-5 /home/linuxbrew/
linuxbrew/bin/python3 /home/sandrords/Documentos/estacio/bubble.sort.py
[29, 49, 49, 51, 51, 55, 66, 66, 78, 78, 83, 83, 89, 90, 91]
```

The terminal output shows the execution of the script, displaying the sorted array: `[29, 49, 49, 51, 51, 55, 66, 66, 78, 78, 83, 83, 89, 90, 91]`.

Implementando algoritmo Bubble Sort.

3. Microatividade 3



The image shows a Visual Studio Code editor window titled "selection.sort.py - estacio - Visual Studio Code". The editor is displaying a Python script for the Selection Sort algorithm. The script is as follows:

```
1 def selectionSort(arr):
2     for selectionIndex in range(len(arr) - 1):
3         for comparisonIndex in range(selectionIndex + 1, len(arr)):
4             if (arr[comparisonIndex] < arr[selectionIndex]):
5                 aux = arr[selectionIndex]
6                 arr[selectionIndex] = arr[comparisonIndex]
7                 arr[comparisonIndex] = aux
8
9
10 arr = [
11     78, 66, 49, 83, 51, 91, 90, 89, 29,
12     55, 78, 66, 49, 83, 51
13 ]
14
15 selectionSort(arr)
16
17 print(arr)
```

The Explorer sidebar on the left shows a folder named "ESTACIO" containing three files: "array.sort.py", "bubble.sort.py", and "selection.sort.py". The "selection.sort.py" file is currently selected.

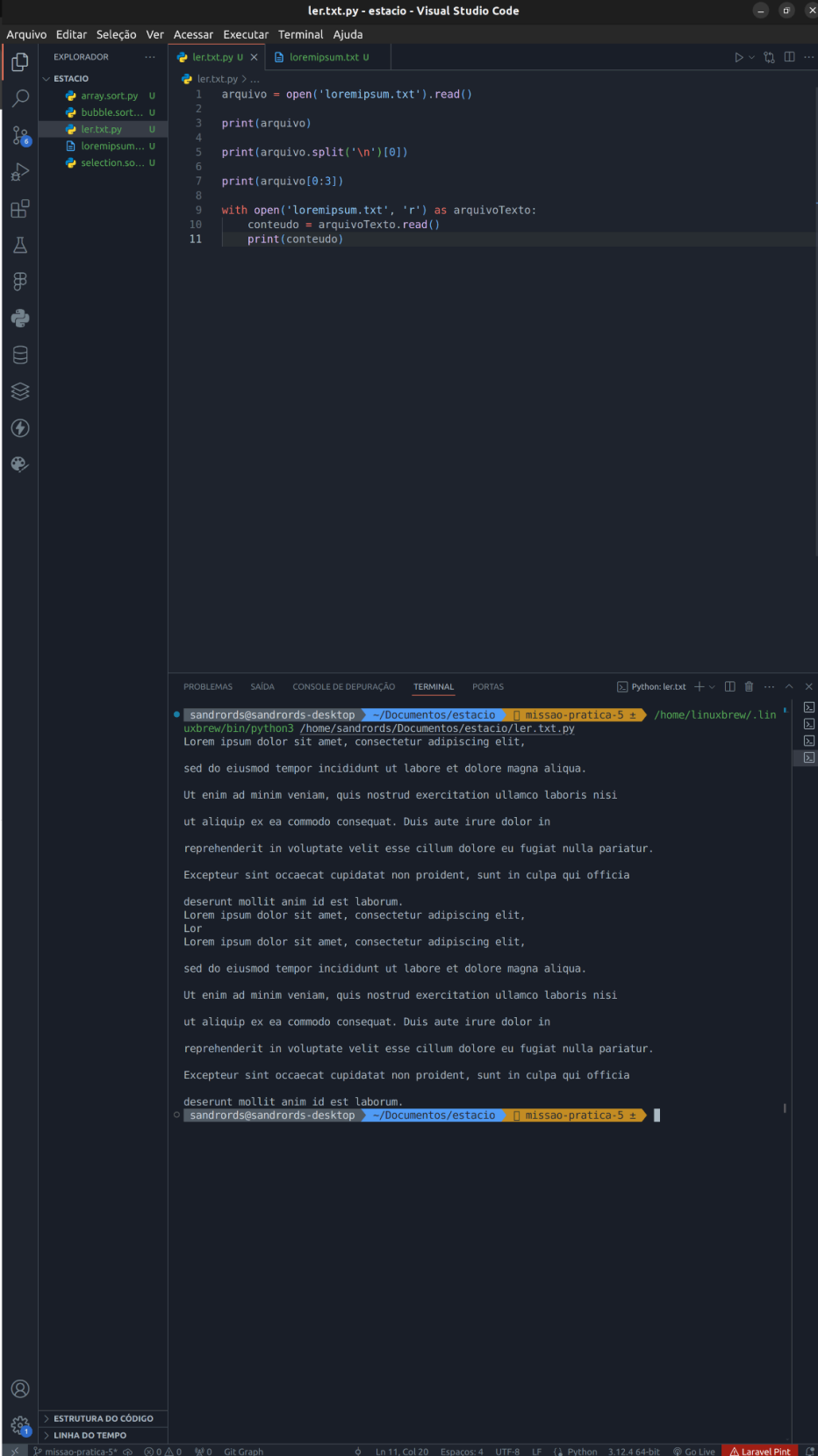
The Terminal panel at the bottom shows the execution of the script. The command prompt is "/home/linuxbrew/.linuxbrew/bin/python3 /home/sandrords/Documents/estacio/selection.sort.py". The output of the script is:

```
[29, 49, 49, 51, 51, 55, 66, 66, 78, 78, 83, 83, 89, 90, 91]
```

The status bar at the bottom indicates the file is "missao-pratica-5.py", the editor is in "Ln 17, Col 11", the encoding is "UTF-8", the line ending is "LF", the language is "Python", the version is "3.12.4 64-bit", and the environment is "Go Live". The "Laravel Pint" extension is also visible.

Implementando algoritmo Selection Sort.

4. Microatividade 4



The screenshot shows the Visual Studio Code interface with a Python file named `ler.txt.py` open. The file contains the following code:

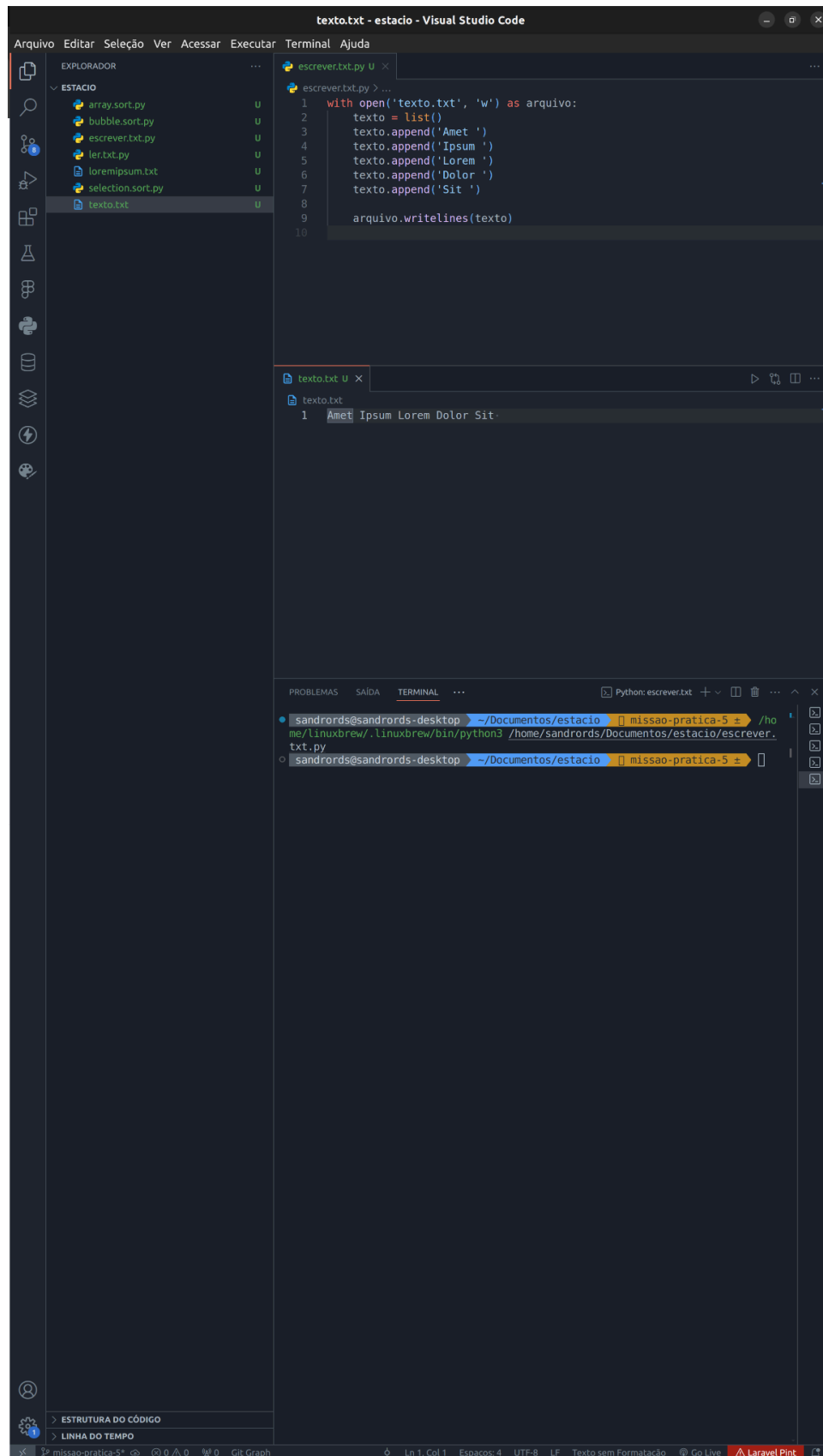
```
1 arquivo = open('loremipsum.txt').read()
2
3 print(arquivo)
4
5 print(arquivo.split('\n')[0])
6
7 print(arquivo[0:3])
8
9 with open('loremipsum.txt', 'r') as arquivoTexto:
10     conteudo = arquivoTexto.read()
11     print(conteudo)
```

The terminal output shows the execution of the script, displaying the contents of the file `loremipsum.txt` in three parts: the first line, the first three characters, and the entire file content.

```
sandrords@sandrords-desktop ~/Documentos/estacio [missao-pratica-5] /home/linuxbrew/.lin
uxbrew/bin/python3 /home/sandrords/Documentos/estacio/ler.txt.py
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.
Lorem ipsum dolor sit amet, consectetur adipiscing elit,
Lor
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.
```

Leitura de arquivos.

5. Microatividade 5



The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORADOR (Explorer):** Lists files in the 'ESTACIO' directory: `array.sort.py`, `bubble.sort.py`, `escrever.txt.py`, `ler.txt.py`, `loremipsum.txt`, `selection.sort.py`, and `texto.txt`.
- EDITOR:** Displays the `escrever.txt.py` file with the following Python code:

```
1 with open('texto.txt', 'w') as arquivo:
2     texto = list()
3     texto.append('Amet ')
4     texto.append('Ipsum ')
5     texto.append('Lorem ')
6     texto.append('Dolor ')
7     texto.append('Sit ')
8
9     arquivo.writelines(texto)
10
```
- TERMINAL:** Shows the command prompt output for running the script:

```
sandrords@sandrords-desktop ~/Documentos/estacio missao-pratica-5 ➤ /home/linuxbrew/.linuxbrew/bin/python3 /home/sandrords/Documentos/estacio/escrever.txt.py
sandrords@sandrords-desktop ~/Documentos/estacio missao-pratica-5 ➤
```
- texto.txt:** A separate editor window showing the output of the script:

```
1 Amet Ipsum Lorem Dolor Sit
```

Escrita de arquivos.

Missão Prática

The image shows a Visual Studio Code editor window titled "kdd.py - estacio - Visual Studio Code". The editor displays a Python script named "kdd.py" with the following code:

```
1 import time
2
3 def bubbleSort(arr):
4     for lap in range(len(arr) - 1):
5         for index in range(0, len(arr) - lap - 1):
6             if (arr[index + 1] < arr[index]):
7                 aux = arr[index]
8                 arr[index] = arr[index + 1]
9                 arr[index + 1] = aux
10
11 def selectionSort(arr):
12     for selectionIndex in range(len(arr) - 1):
13         for comparisonIndex in range(selectionIndex + 1, len(arr)):
14             if (arr[comparisonIndex] < arr[selectionIndex]):
15                 aux = arr[selectionIndex]
16                 arr[selectionIndex] = arr[comparisonIndex]
17                 arr[comparisonIndex] = aux
18
19
20 with open('farAway.txt', 'r') as arquivo:
21     linhas = arquivo.read().strip()
22     palavras = linhas.split(' ')
23     tempoInicial = time.perf_counter()
24     bubbleSort(palavras)
25     tempoFinal = time.perf_counter()
26
27     print(f'Tempo de Execução do Bubble Sort: {(tempoFinal - tempoInicial):.4f} segundos')
28
29 with open('farAway.txt', 'r') as arquivo:
30     linhas = arquivo.read().strip()
31     palavras = linhas.split(' ')
32     tempoInicial = time.perf_counter()
33     selectionSort(palavras)
34     tempoFinal = time.perf_counter()
35
36     print(f'Tempo de Execução do Selection Sort: {(tempoFinal - tempoInicial):.4f} segundos')
37
38
39 with open('farAway.txt', 'r') as arquivo:
40     linhas = arquivo.read().strip()
41     palavras = linhas.split(' ')
42     tempoInicial = time.perf_counter()
43     palavras.sort()
44     tempoFinal = time.perf_counter()
45
46     print(f'Tempo de Execução do Método Nativo Sort: {(tempoFinal - tempoInicial):.4f} segundos')
47
```

The terminal output shows the execution results for three different sorting algorithms:

```
Linuxbrew/bin/python3 /home/sandrords/Documents/estacio/kdd.py
Tempo de Execução do Bubble Sort: 0.0080 segundos
Tempo de Execução do Selection Sort: 0.0055 segundos
Tempo de Execução do Método Nativo Sort: 0.0001 segundos
Linuxbrew/bin/python3 /home/sandrords/Documents/estacio/kdd.py
Tempo de Execução do Bubble Sort: 0.0078 segundos
Tempo de Execução do Selection Sort: 0.0059 segundos
Tempo de Execução do Método Nativo Sort: 0.0001 segundos
Linuxbrew/bin/python3 /home/sandrords/Documents/estacio/kdd.py
Tempo de Execução do Bubble Sort: 0.0079 segundos
Tempo de Execução do Selection Sort: 0.0056 segundos
Tempo de Execução do Método Nativo Sort: 0.0001 segundos
Linuxbrew/bin/python3 /home/sandrords/Documents/estacio/kdd.py
Tempo de Execução do Bubble Sort: 0.0079 segundos
Tempo de Execução do Selection Sort: 0.0055 segundos
Tempo de Execução do Método Nativo Sort: 0.0001 segundos
```

Descobrimos algoritmo com melhor tempo de execução

The screenshot shows the Visual Studio Code interface with the following components:

- Explorer (EXPLORADOR):** Displays a file tree for a project named "ESTACIO". Files include `array.sort.py`, `bubble.sort.py`, `escrever.txt.py`, `farAway.txt`, `farAwayOrdenado.txt`, `kdd.py`, `ler.txt.py`, `loremipsum.txt`, `selection.sort.py`, and `texto.txt`.
- Editor:** The active file is `kdd.py`, containing the following Python code:

```
1 import time
2
3 with open('farAway.txt', 'r') as arquivo:
4     linhas = arquivo.read().strip()
5     palavras = linhas.split(' ')
6     palavras.sort()
7
8 with open('farAwayOrdenado.txt', 'w') as arquivo:
9     arquivo.write(' '.join(palavras))
10
```
- Output View:** The "TERMINAL" tab is active, showing the command prompt output:

```
sandrords@sandrords-desktop ~/Documentos/estacio missao-pratica-5
$ python kdd.py
```
- File Content:** The file `farAwayOrdenado.txt` is open, displaying the sorted words from the input file, such as "and", "A", "Alphabet", "Alphabet", "And", "Big", "Big", "Blind", "Blind", "Blind", "Blind", "Blind", "Bookmarks", "grove", "Bookmarks", "grove", "The", "Grammar", "The", "Ipsum", "Ipsum", "It", "Italic", "Italic", "Lane", "Line", "Little", "Little", "Little", "Little", "Longe", "Lorem", "Lorem", "Ma", "But", "day", "day", "decided", "decided", "devious", "devious", "didn't", "didn't", "didn't", "do", "do", "dragged", "drunk", "everyth", "Far", "herself", "herself", "hills", "hills", "hometown", "hometown", "however", "however", "if", "in", "in", "in", "in", "initial", "in", "Even", "name", "name", "named", "named", "necessary", "necessary", "no", "no", "not", "not", "nothing", "ocean", "ocean", "A", "of", "of", "of", "of", "of", "of", "of", "of", "of", "of", "of", "of", "of", "on", "on", "on", "origin", "over", "own", "own", "It", "rethoric", "return", "rewritten", "rewritten", "right", "right", "river", "river", "road", "roasted", "roasted", "safe", "On", "way", "When", "were", "were", "where", "where", "which", "which", "wild", "wild", "with", "with", "with", "word", "word", "word", "would", "would", "y".

Ordenando com o método Sort(), algoritmo que apresentou melhor performance, e enviando o resultado para o arquivo farAwayOrdenado.txt