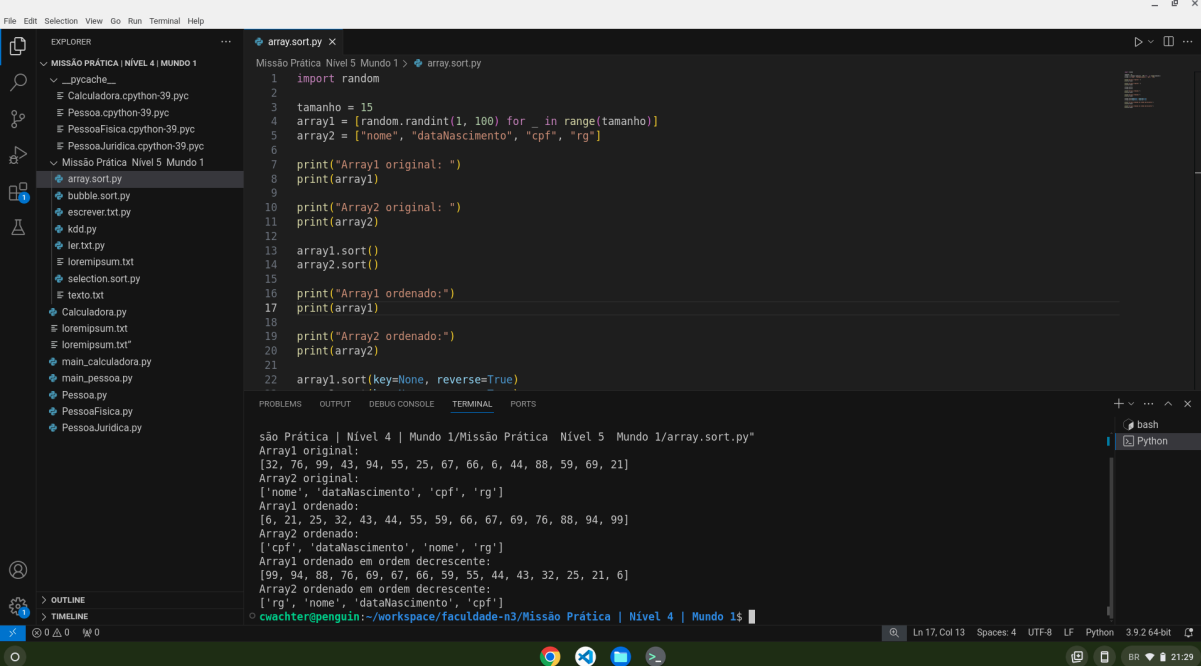


Missão Prática Nível 5 Mundo

Microatividade 1



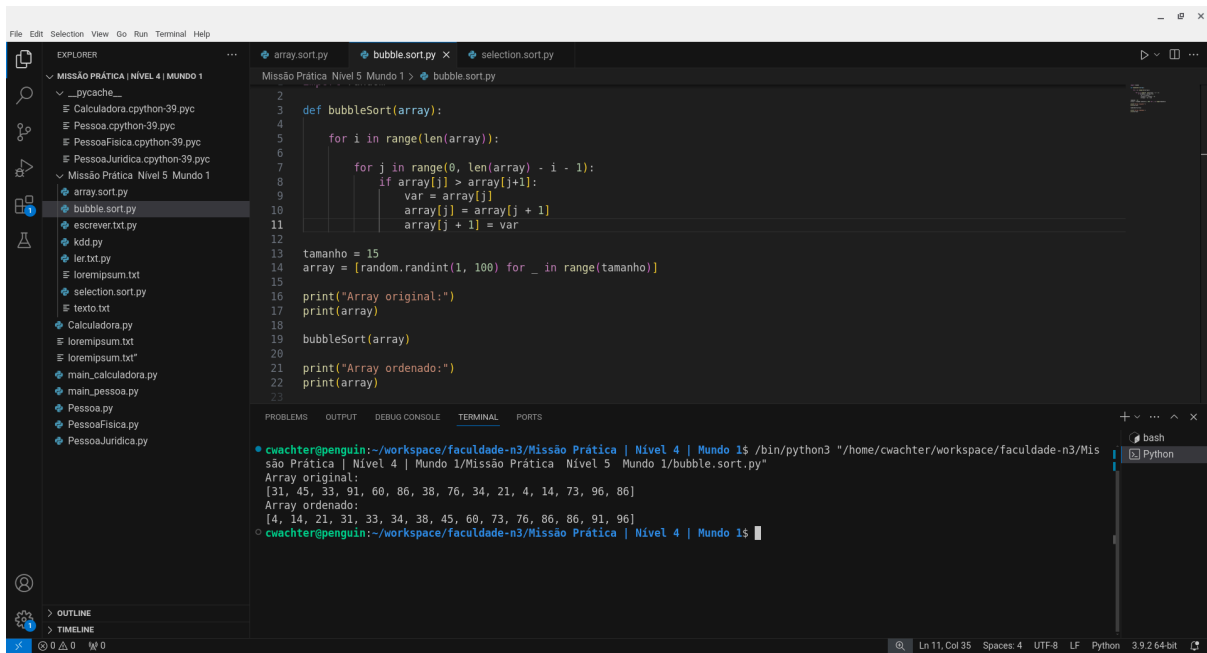
The screenshot shows a Visual Studio Code editor window with a Python file named `array.sort.py` open. The file is located in the `Missão Prática | Nível 5 | Mundo 1` directory. The code in the file is as follows:

```
1 import random
2
3 tamanho = 15
4 array1 = [random.randint(1, 100) for _ in range(tamanho)]
5 array2 = ['nome', 'dataNascimento', 'cpf', 'rg']
6
7 print("Array1 original: ")
8 print(array1)
9
10 print("Array2 original: ")
11 print(array2)
12
13 array1.sort()
14 array2.sort()
15
16 print("Array1 ordenado:")
17 print(array1)
18
19 print("Array2 ordenado:")
20 print(array2)
21
22 array1.sort(key=None, reverse=True)
```

The terminal output shows the execution of the script, displaying the original and sorted arrays:

```
são Prática | Nível 4 | Mundo 1/Missão Prática | Nível 5 | Mundo 1/array.sort.py"
Array1 original:
[32, 76, 99, 43, 94, 55, 25, 67, 66, 6, 44, 88, 59, 69, 21]
Array2 original:
['nome', 'dataNascimento', 'cpf', 'rg']
Array1 ordenado:
[6, 21, 25, 32, 43, 44, 55, 59, 66, 67, 69, 76, 88, 94, 99]
Array2 ordenado:
['cpf', 'dataNascimento', 'nome', 'rg']
Array1 ordenado em ordem decrescente:
[99, 94, 88, 76, 69, 67, 66, 59, 55, 44, 43, 32, 25, 21, 6]
Array2 ordenado em ordem decrescente:
['rg', 'nome', 'dataNascimento', 'cpf']
cwachter@penguin: ~/workspace/faculdade-n3/Missão Prática | Nível 4 | Mundo 1$
```

Microatividade 2

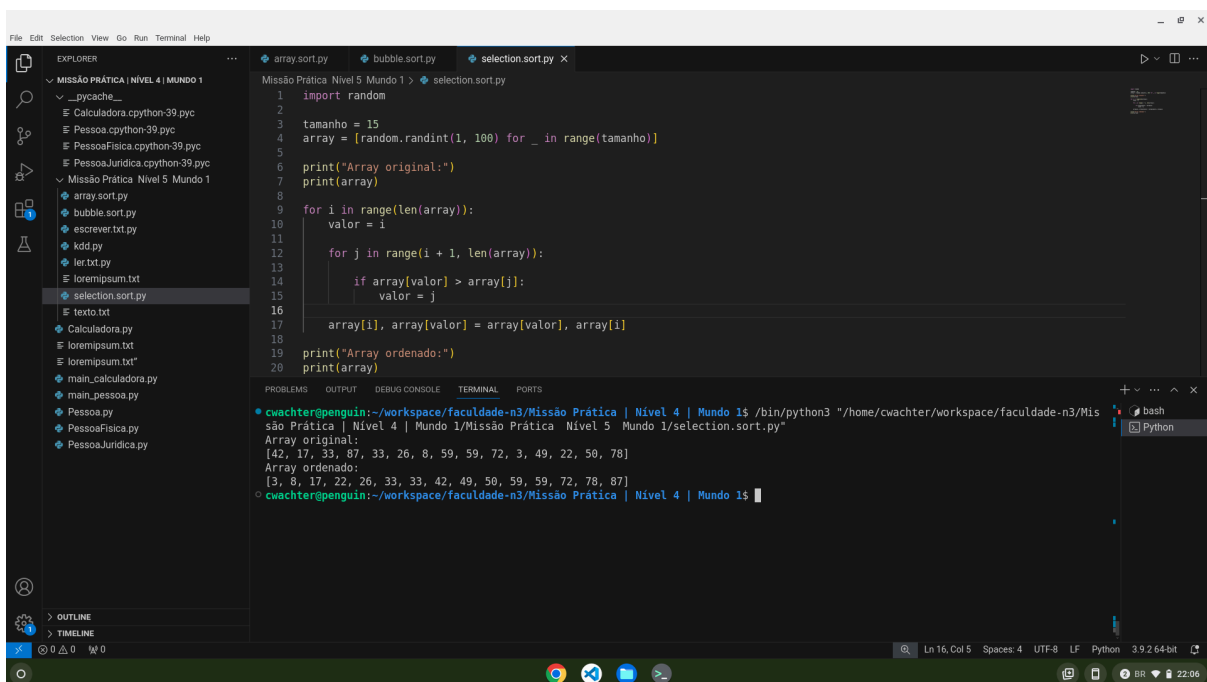


The screenshot shows a VS Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'MISSÃO PRÁTICA | NÍVEL 4 | MUNDO 1' with several files, including 'bubble.sort.py'. The main editor window displays the code for 'bubble.sort.py', which implements a bubble sort algorithm. The code includes a function 'bubbleSort(array)' that iterates through the array, comparing adjacent elements and swapping them if they are in the wrong order. The array is generated using 'random.randint(1, 100)' and the sorted array is printed.

```
2
3 def bubbleSort(array):
4     for i in range(len(array)):
5         for j in range(0, len(array) - i - 1):
6             if array[j] > array[j+1]:
7                 var = array[j]
8                 array[j] = array[j + 1]
9                 array[j + 1] = var
10
11 tamanho = 15
12 array = [random.randint(1, 100) for _ in range(tamanho)]
13
14 print("Array original:")
15 print(array)
16
17 bubbleSort(array)
18
19 print("Array ordenado:")
20 print(array)
```

The terminal output shows the execution of the script, displaying the original array [31, 45, 33, 91, 60, 86, 38, 76, 34, 21, 4, 14, 73, 96, 86] and the sorted array [4, 14, 21, 31, 33, 34, 38, 45, 60, 73, 76, 86, 86, 91, 96].

Microatividade 3

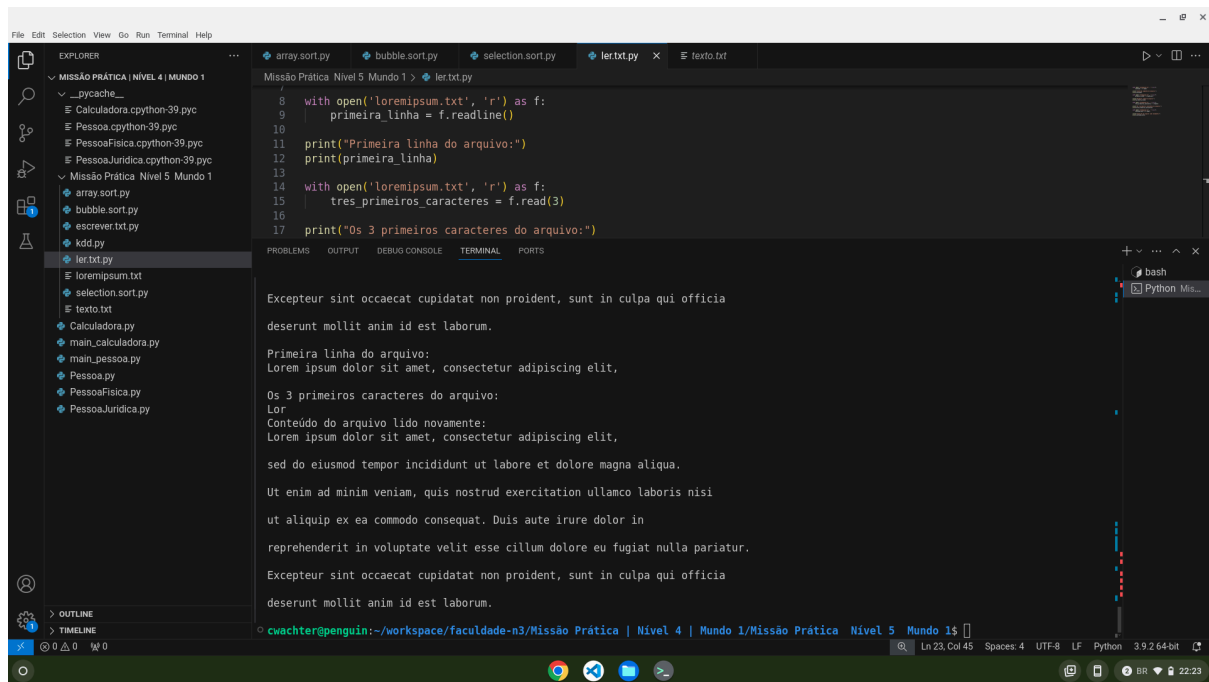


The screenshot shows a VS Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project named 'MISSÃO PRÁTICA | NÍVEL 4 | MUNDO 1' with several files, including 'selection.sort.py'. The main editor window displays the code for 'selection.sort.py', which implements a selection sort algorithm. The code includes a function 'selectionSort(array)' that iterates through the array, finding the minimum element and swapping it with the first element. The array is generated using 'random.randint(1, 100)' and the sorted array is printed.

```
1 import random
2
3 tamanho = 15
4 array = [random.randint(1, 100) for _ in range(tamanho)]
5
6 print("Array original:")
7 print(array)
8
9 for i in range(len(array)):
10     valor = i
11     for j in range(i + 1, len(array)):
12         if array[valor] > array[j]:
13             valor = j
14
15     array[i], array[valor] = array[valor], array[i]
16
17 print("Array ordenado:")
18 print(array)
```

The terminal output shows the execution of the script, displaying the original array [42, 17, 33, 87, 33, 26, 8, 59, 59, 72, 3, 49, 22, 50, 78] and the sorted array [3, 8, 17, 22, 26, 33, 33, 42, 49, 50, 59, 59, 72, 78, 87].

Microatividade 4



The screenshot shows the VS Code editor interface. The Explorer panel on the left displays a file tree for 'MISSÃO PRÁTICA | NÍVEL 4 | MUNDO 1'. The file 'ler.txt.py' is selected. The main editor area shows the code for 'ler.txt.py':

```
1 with open('loremipsum.txt', 'r') as f:
2     primeira_linha = f.readline()
3
4 print("Primeira linha do arquivo:")
5 print(primeira_linha)
6
7 with open('loremipsum.txt', 'r') as f:
8     tres_primeiros_caracteres = f.read(3)
9
10 print("Os 3 primeiros caracteres do arquivo:")
```

The TERMINAL panel at the bottom shows the output of the script:

```
Exception: sint occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum.

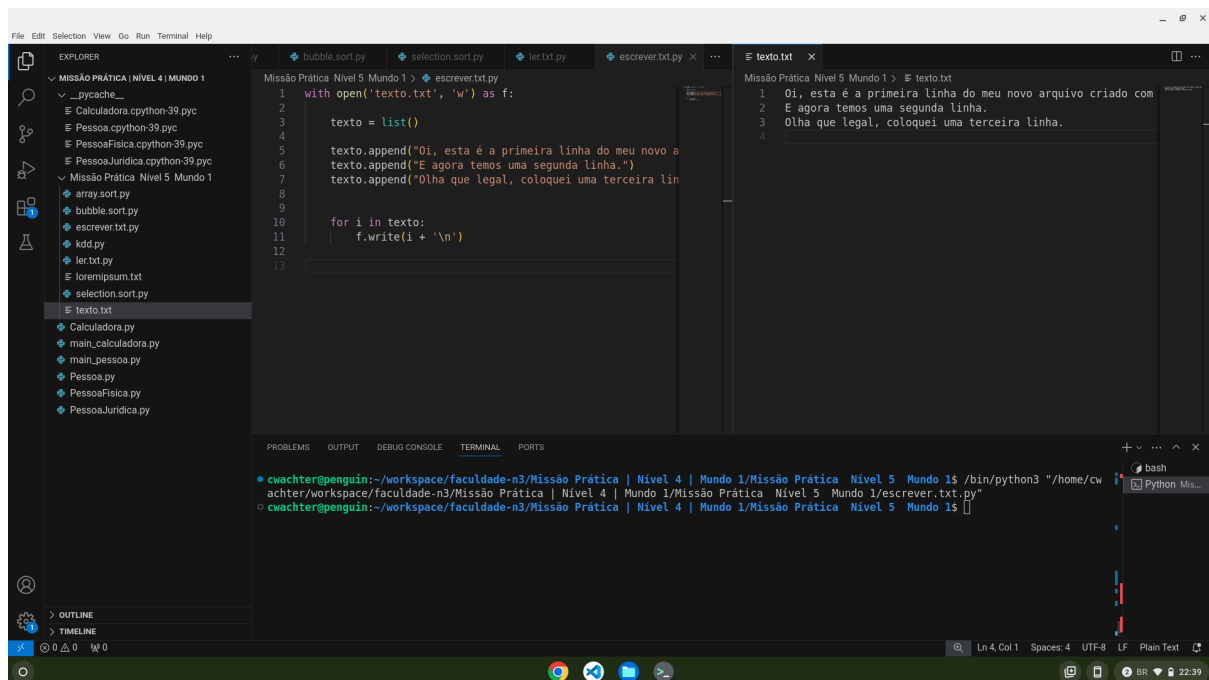
Primeira linha do arquivo:
Lorem ipsum dolor sit amet, consectetur adipiscing elit,

Os 3 primeiros caracteres do arquivo:
Lor
Conteúdo do arquivo lido novamente:
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.
```

Microatividade 5



The screenshot shows the VS Code editor interface. The Explorer panel on the left displays a file tree for 'MISSÃO PRÁTICA | NÍVEL 4 | MUNDO 1'. The file 'escrever.txt.py' is selected. The main editor area shows the code for 'escrever.txt.py':

```
1 with open('texto.txt', 'w') as f:
2
3     texto = list()
4
5     texto.append("Oi, esta é a primeira linha do meu novo a")
6     texto.append("E agora temos uma segunda linha.")
7     texto.append("Olha que legal, coloquei uma terceira lin")
8
9
10 for i in texto:
11     f.write(i + '\n')
```

The TERMINAL panel at the bottom shows the output of the script:

```
1 Oi, esta é a primeira linha do meu novo arquivo criado com
2 E agora temos uma segunda linha.
3 Olha que legal, coloquei uma terceira linha.
```

Missão Prática | Colocando tudo em ordem e guardando

The screenshot shows a VS Code editor with a project named "MISSÃO PRÁTICA | NÍVEL 4 | MUNDO 1". The Explorer sidebar on the left lists files including "_pycache_", "Calculadora.cpython-39.pyc", "Pessoa.cpython-39.pyc", "PessoaFisica.cpython-39.pyc", "PessoaJuridica.cpython-39.pyc", and "Missão Prática | Nível 5 | Mundo 1". Under the mission folder, there are files for "array.sort.py", "bubble.sort.py", "selection.sort.py", "ler.txt.py", "escrever.txt.py", "kdd.py", "ordenado.txt", "ler.txt.py", "loremipsum.txt", "calculadora.py", "main_calculadora.py", "main_pessoa.py", "pessoa.py", "pessoafisica.py", and "pessoajuridica.py".

The main editor window displays the content of "ordenado.txt":

```
1 E
2 Oi,
3 Olha
4 a
5 agora
6 arquivo
7 coloquei
8 com
9 criado
10 do
11 esta
```

The TERMINAL panel at the bottom shows the output of a Python script. It displays the words extracted from the file, the sorting process using Bubble Sort and Selection Sort, and the final sorted words written to the file. The execution times for the sorting algorithms are also shown.

```
Palavras extraídas do arquivo:
['Oi,', 'esta', 'é', 'a', 'primeira', 'linha', 'do', 'meu', 'novo', 'arquivo', 'criado', 'com', 'python!', 'E', 'agora', 'temos', 'um
a', 'segunda', 'linha.', 'Olha', 'que', 'legal,', 'coloquei', 'uma', 'terceira', 'linha.']
Ordenação usando Bubble Sort:
['E', 'Oi,', 'Olha', 'a', 'agora', 'arquivo', 'coloquei', 'com', 'criado', 'do', 'esta', 'legal,', 'linha', 'linha.', 'linha.', 'meu'
, 'novo', 'primeira', 'python!', 'que', 'segunda', 'temos', 'terceira', 'uma', 'uma', 'é']
Tempo de execução (Bubble Sort): 0.000267 segundos

Palavras extraídas do arquivo:
['Oi,', 'esta', 'é', 'a', 'primeira', 'linha', 'do', 'meu', 'novo', 'arquivo', 'criado', 'com', 'python!', 'E', 'agora', 'temos', 'um
a', 'segunda', 'linha.', 'Olha', 'que', 'legal,', 'coloquei', 'uma', 'terceira', 'linha.']
Ordenação usando Selection Sort:
['E', 'Oi,', 'Olha', 'a', 'agora', 'arquivo', 'coloquei', 'com', 'criado', 'do', 'esta', 'legal,', 'linha', 'linha.', 'linha.', 'meu'
, 'novo', 'primeira', 'python!', 'que', 'segunda', 'temos', 'terceira', 'uma', 'uma', 'é']
Tempo de execução (Selection Sort): 0.000105 segundos

Palavras extraídas do arquivo:
['Oi,', 'esta', 'é', 'a', 'primeira', 'linha', 'do', 'meu', 'novo', 'arquivo', 'criado', 'com', 'python!', 'E', 'agora', 'temos', 'um
a', 'segunda', 'linha.', 'Olha', 'que', 'legal,', 'coloquei', 'uma', 'terceira', 'linha.']
Ordenação usando o método sort() nativo:
['E', 'Oi,', 'Olha', 'a', 'agora', 'arquivo', 'coloquei', 'com', 'criado', 'do', 'esta', 'legal,', 'linha', 'linha.', 'linha.', 'meu'
, 'novo', 'primeira', 'python!', 'que', 'segunda', 'temos', 'terceira', 'uma', 'uma', 'é']
Tempo de execução (sort()) nativo: 0.000007 segundos

Palavras ordenadas foram escritas no arquivo 'ordenado.txt'.
```

The status bar at the bottom indicates the current file is "ordenado.txt" at line 8, column 4, with 4 spaces, UTF-8 encoding, and LF line endings.