

**Pruebas de software y aseguramiento de la calidad**

**4.2 Ejercicio de programación 1**

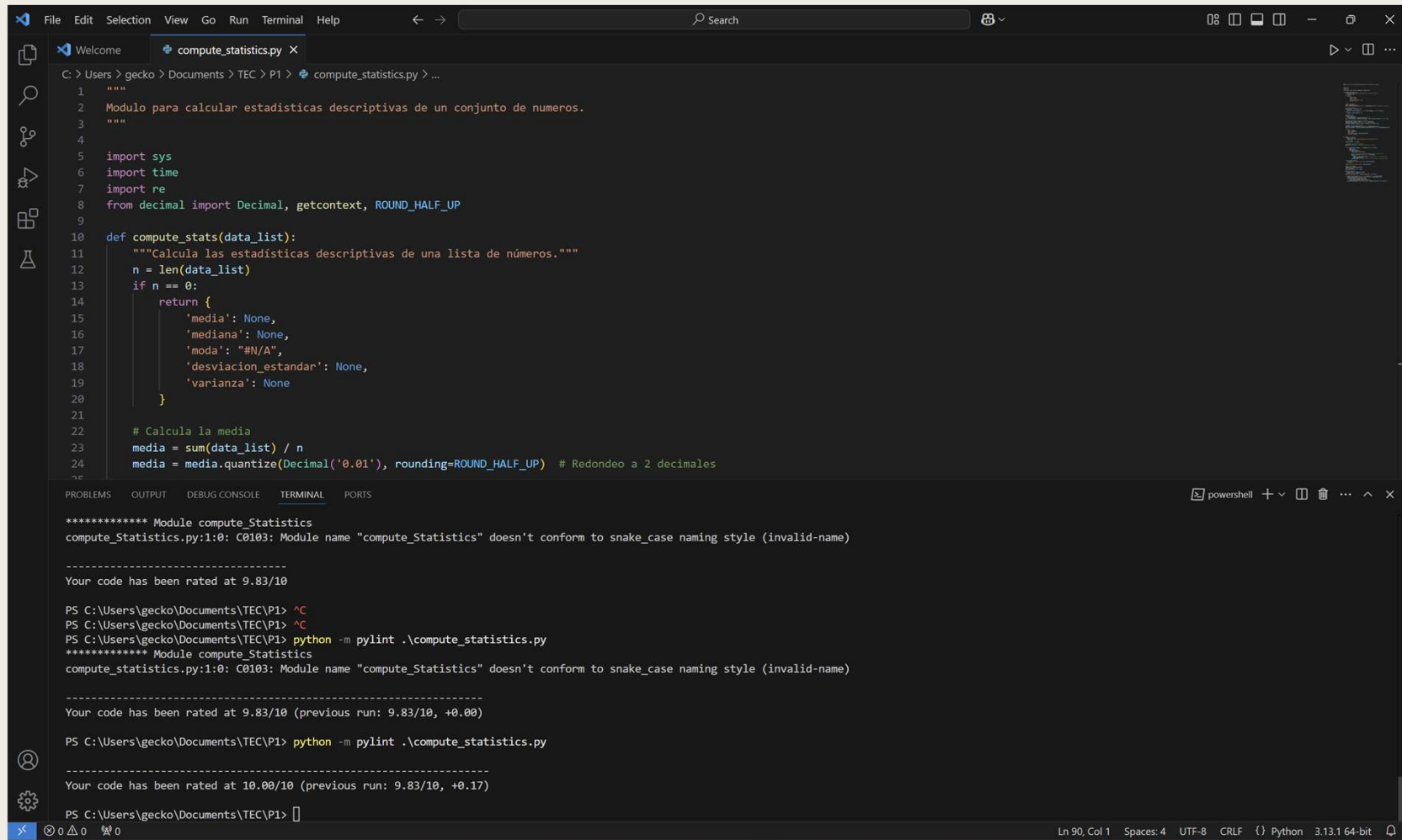
**Profesor:** Dr. Gerardo Padilla Zárate

**Profesora Asistente:** Mtra. María Mylen Treviño E.

**Alumno:** Daniel Acevedo Sainos - **A01795496**

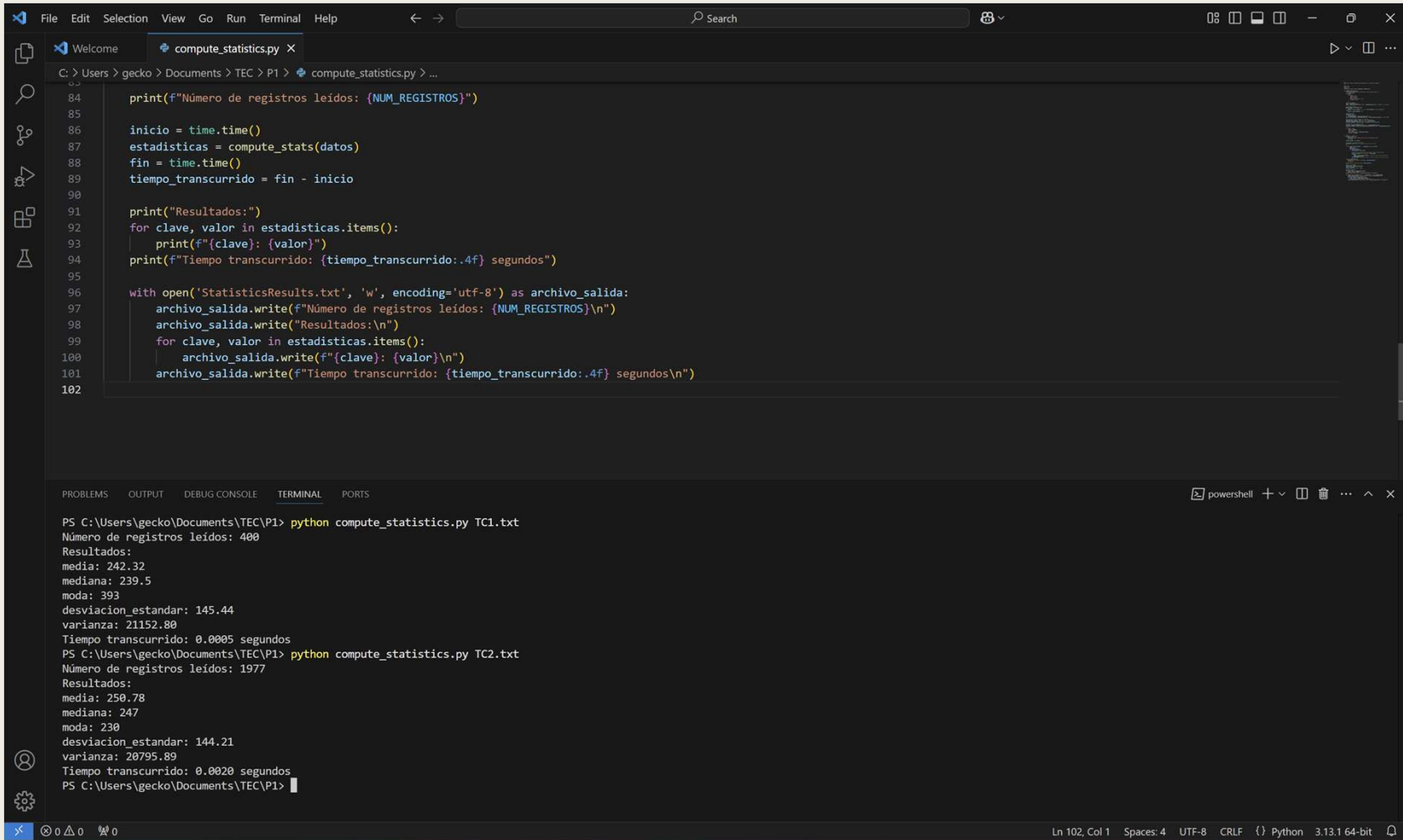
02 de febrero de 2025

## Comprobación del ejercicio 1 “Compute statistics” al 100% con Pylint



```
File Edit Selection View Go Run Terminal Help
compute_statistics.py x
C: > Users > gecko > Documents > TEC > P1 > compute_statistics.py > ...
1 """
2 Modulo para calcular estadísticas descriptivas de un conjunto de numeros.
3 """
4
5 import sys
6 import time
7 import re
8 from decimal import Decimal, getcontext, ROUND_HALF_UP
9
10 def compute_stats(data_list):
11     """Calcula las estadísticas descriptivas de una lista de números."""
12     n = len(data_list)
13     if n == 0:
14         return {
15             'media': None,
16             'mediana': None,
17             'moda': "#N/A",
18             'desviacion_estandar': None,
19             'varianza': None
20         }
21
22     # Calcula la media
23     media = sum(data_list) / n
24     media = media.quantize(Decimal('0.01'), rounding=ROUND_HALF_UP) # Redondeo a 2 decimales
25
26 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
27 ***** Module compute_Statistics
28 compute_Statistics.py:1:0: C0103: Module name "compute_Statistics" doesn't conform to snake_case naming style (invalid-name)
29
30 -----
31 Your code has been rated at 9.83/10
32
33 PS C:\Users\gecko\Documents\TEC\P1> ^C
34 PS C:\Users\gecko\Documents\TEC\P1> ^C
35 PS C:\Users\gecko\Documents\TEC\P1> python -m pylint .\compute_statistics.py
36 ***** Module compute_Statistics
37 compute_statistics.py:1:0: C0103: Module name "compute_Statistics" doesn't conform to snake_case naming style (invalid-name)
38
39 -----
40 Your code has been rated at 9.83/10 (previous run: 9.83/10, +0.00)
41
42 PS C:\Users\gecko\Documents\TEC\P1> python -m pylint .\compute_statistics.py
43
44 -----
45 Your code has been rated at 10.00/10 (previous run: 9.83/10, +0.17)
46
47 PS C:\Users\gecko\Documents\TEC\P1>
48
49 Ln 90, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.13.1 64-bit
```

## Problema 1 corregido con Pylint: Ejecución ejercicio TC1 y TC2



The image shows a Visual Studio Code editor window with a Python file named `compute_statistics.py` open. The file path is `C:\Users\gecko\Documents\TEC\P1> compute_statistics.py`. The code in the file is as follows:

```
84 print(f"Número de registros leídos: {NUM_REGISTROS}")
85
86 inicio = time.time()
87 estadisticas = compute_stats(datos)
88 fin = time.time()
89 tiempo_transcurrido = fin - inicio
90
91 print("Resultados:")
92 for clave, valor in estadisticas.items():
93     print(f"{clave}: {valor}")
94 print(f"Tiempo transcurrido: {tiempo_transcurrido:.4f} segundos")
95
96 with open('StatisticsResults.txt', 'w', encoding='utf-8') as archivo_salida:
97     archivo_salida.write(f"Número de registros leídos: {NUM_REGISTROS}\n")
98     archivo_salida.write("Resultados:\n")
99     for clave, valor in estadisticas.items():
100         archivo_salida.write(f"{clave}: {valor}\n")
101     archivo_salida.write(f"Tiempo transcurrido: {tiempo_transcurrido:.4f} segundos\n")
102
```

Below the editor, the `TERMINAL` panel shows the execution of the script for two test cases, TC1 and TC2. The output for TC1 is:

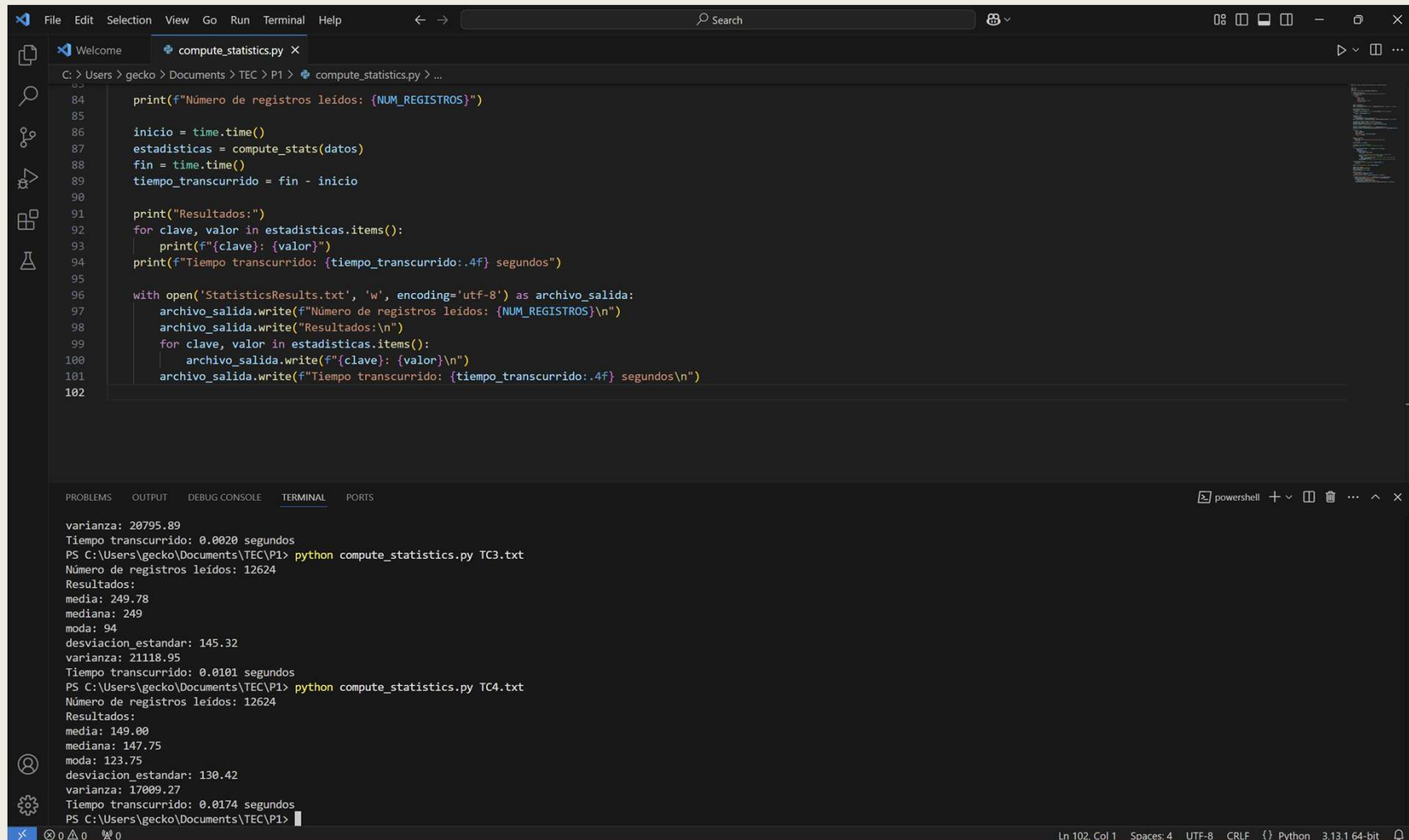
```
PS C:\Users\gecko\Documents\TEC\P1> python compute_statistics.py TC1.txt
Número de registros leídos: 400
Resultados:
media: 242.32
mediana: 239.5
moda: 393
desviacion_estandar: 145.44
varianza: 21152.80
Tiempo transcurrido: 0.0005 segundos
```

The output for TC2 is:

```
PS C:\Users\gecko\Documents\TEC\P1> python compute_statistics.py TC2.txt
Número de registros leídos: 1977
Resultados:
media: 250.78
mediana: 247
moda: 230
desviacion_estandar: 144.21
varianza: 20795.89
Tiempo transcurrido: 0.0020 segundos
```

The status bar at the bottom indicates the current position is `Ln 102, Col 1`, with `Spaces: 4`, `UTF-8` encoding, `CRLF` line endings, and the interpreter is `Python 3.13.1 64-bit`.

## Problema 1 corregido con Pylint: Ejecución ejercicio TC3 y TC4



The image shows a Visual Studio Code editor window with a Python file named `compute_statistics.py` open. The file path is `C:\Users\gecko\Documents\TEC\P1> compute_statistics.py > ...`. The script contains the following code:

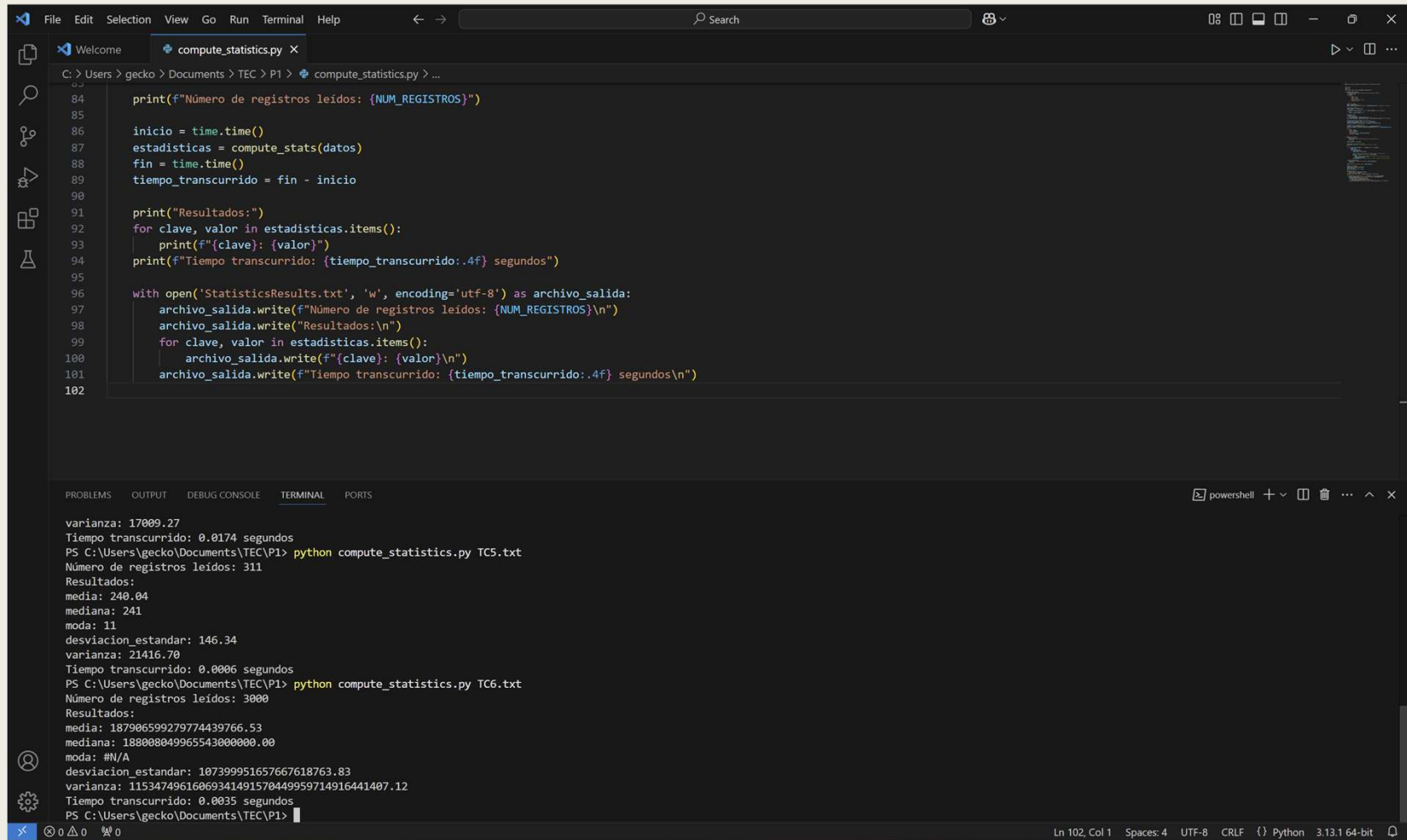
```
84 print(f"Número de registros leídos: {NUM_REGISTROS}")
85
86 inicio = time.time()
87 estadisticas = compute_stats(datos)
88 fin = time.time()
89 tiempo_transcurrido = fin - inicio
90
91 print("Resultados:")
92 for clave, valor in estadisticas.items():
93     print(f"{clave}: {valor}")
94 print(f"Tiempo transcurrido: {tiempo_transcurrido:.4f} segundos")
95
96 with open('StatisticsResults.txt', 'w', encoding='utf-8') as archivo_salida:
97     archivo_salida.write(f"Número de registros leídos: {NUM_REGISTROS}\n")
98     archivo_salida.write("Resultados:\n")
99     for clave, valor in estadisticas.items():
100         archivo_salida.write(f"{clave}: {valor}\n")
101     archivo_salida.write(f"Tiempo transcurrido: {tiempo_transcurrido:.4f} segundos\n")
102
```

The bottom panel shows the terminal output for two test cases:

```
varianza: 20795.89
Tiempo transcurrido: 0.0020 segundos
PS C:\Users\gecko\Documents\TEC\P1> python compute_statistics.py TC3.txt
Número de registros leídos: 12624
Resultados:
media: 249.78
mediana: 249
moda: 94
desviacion_estandar: 145.32
varianza: 21118.95
Tiempo transcurrido: 0.0101 segundos
PS C:\Users\gecko\Documents\TEC\P1> python compute_statistics.py TC4.txt
Número de registros leídos: 12624
Resultados:
media: 149.00
mediana: 147.75
moda: 123.75
desviacion_estandar: 130.42
varianza: 17009.27
Tiempo transcurrido: 0.0174 segundos
PS C:\Users\gecko\Documents\TEC\P1>
```

The status bar at the bottom indicates the file is at line 102, column 1, using UTF-8 encoding, CRLF line endings, and is a Python 3.13.1 64-bit file.

## Problema 1 corregido con Pylint: Ejecución ejercicio TC5 y TC6



The image shows a Visual Studio Code editor window with a Python file named `compute_statistics.py` open. The file is located at `C:\Users\gecko\Documents\TEC\P1\compute_statistics.py`. The code in the file is as follows:

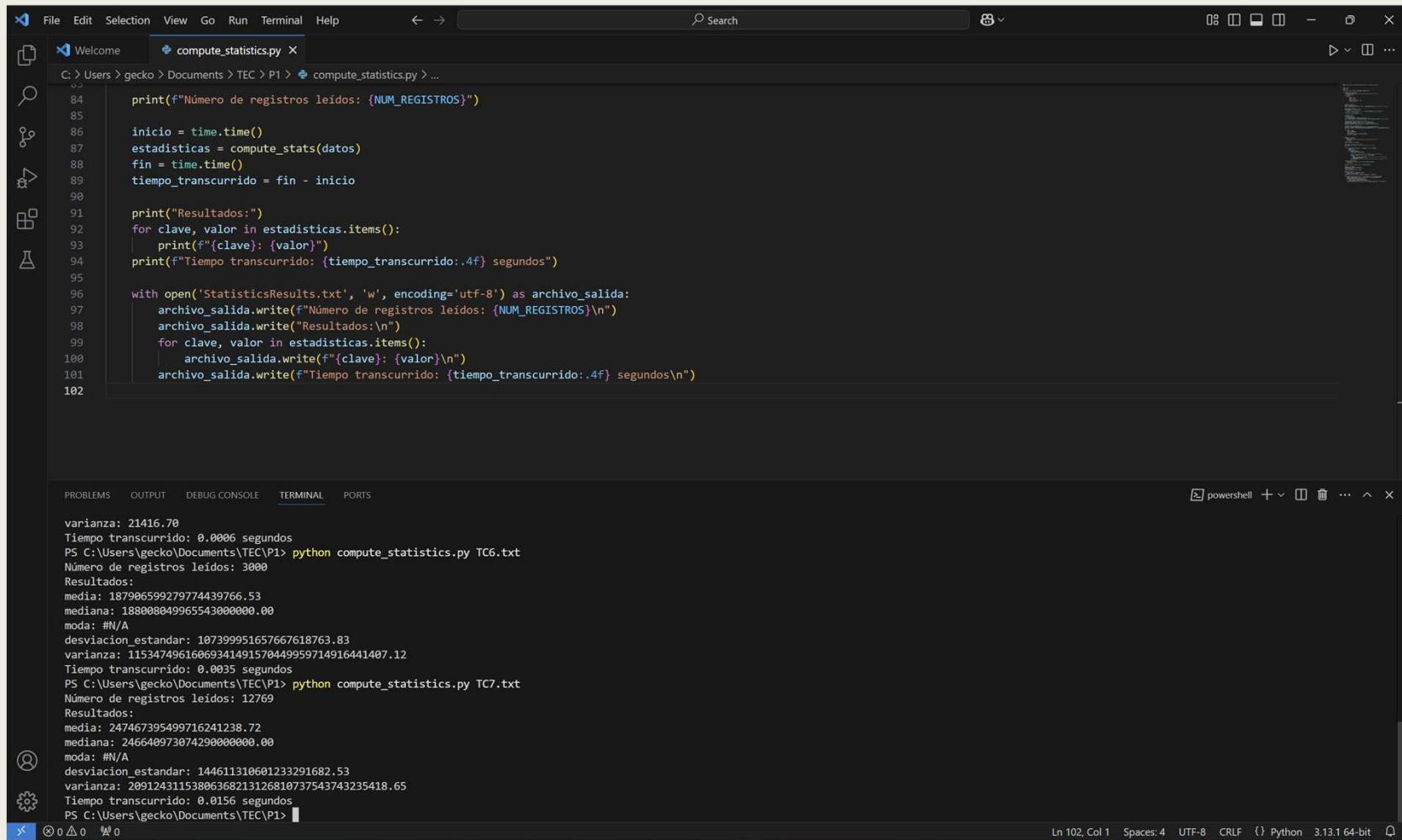
```
84 print(f"Número de registros leídos: {NUM_REGISTROS}")
85
86 inicio = time.time()
87 estadisticas = compute_stats(datos)
88 fin = time.time()
89 tiempo_transcurrido = fin - inicio
90
91 print("Resultados:")
92 for clave, valor in estadisticas.items():
93     print(f"{clave}: {valor}")
94 print(f"Tiempo transcurrido: {tiempo_transcurrido:.4f} segundos")
95
96 with open('StatisticsResults.txt', 'w', encoding='utf-8') as archivo_salida:
97     archivo_salida.write(f"Número de registros leídos: {NUM_REGISTROS}\n")
98     archivo_salida.write("Resultados:\n")
99     for clave, valor in estadisticas.items():
100         archivo_salida.write(f"{clave}: {valor}\n")
101     archivo_salida.write(f"Tiempo transcurrido: {tiempo_transcurrido:.4f} segundos\n")
102
```

Below the editor, the `TERMINAL` tab is active, showing the output of two test cases executed in a PowerShell terminal:

```
varianza: 17009.27
Tiempo transcurrido: 0.0174 segundos
PS C:\Users\gecko\Documents\TEC\P1> python compute_statistics.py TC5.txt
Número de registros leídos: 311
Resultados:
media: 240.04
mediana: 241
moda: 11
desviacion_estandar: 146.34
varianza: 21416.70
Tiempo transcurrido: 0.0006 segundos
PS C:\Users\gecko\Documents\TEC\P1> python compute_statistics.py TC6.txt
Número de registros leídos: 3000
Resultados:
media: 187906599279774439766.53
mediana: 188008049965543000000.00
moda: #N/A
desviacion_estandar: 107399951657667618763.83
varianza: 11534749616069341491570449959714916441407.12
Tiempo transcurrido: 0.0035 segundos
PS C:\Users\gecko\Documents\TEC\P1>
```

The status bar at the bottom indicates the current line is 102, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, Python 3.13.1 64-bit.

## Problema 1 corregido con Pylint: Ejecución ejercicio TC7



The image shows a Visual Studio Code editor window with a Python file named `compute_statistics.py` open. The file is located at `C:\Users\gecko\Documents\TEC\P1\compute_statistics.py`. The code in the file is as follows:

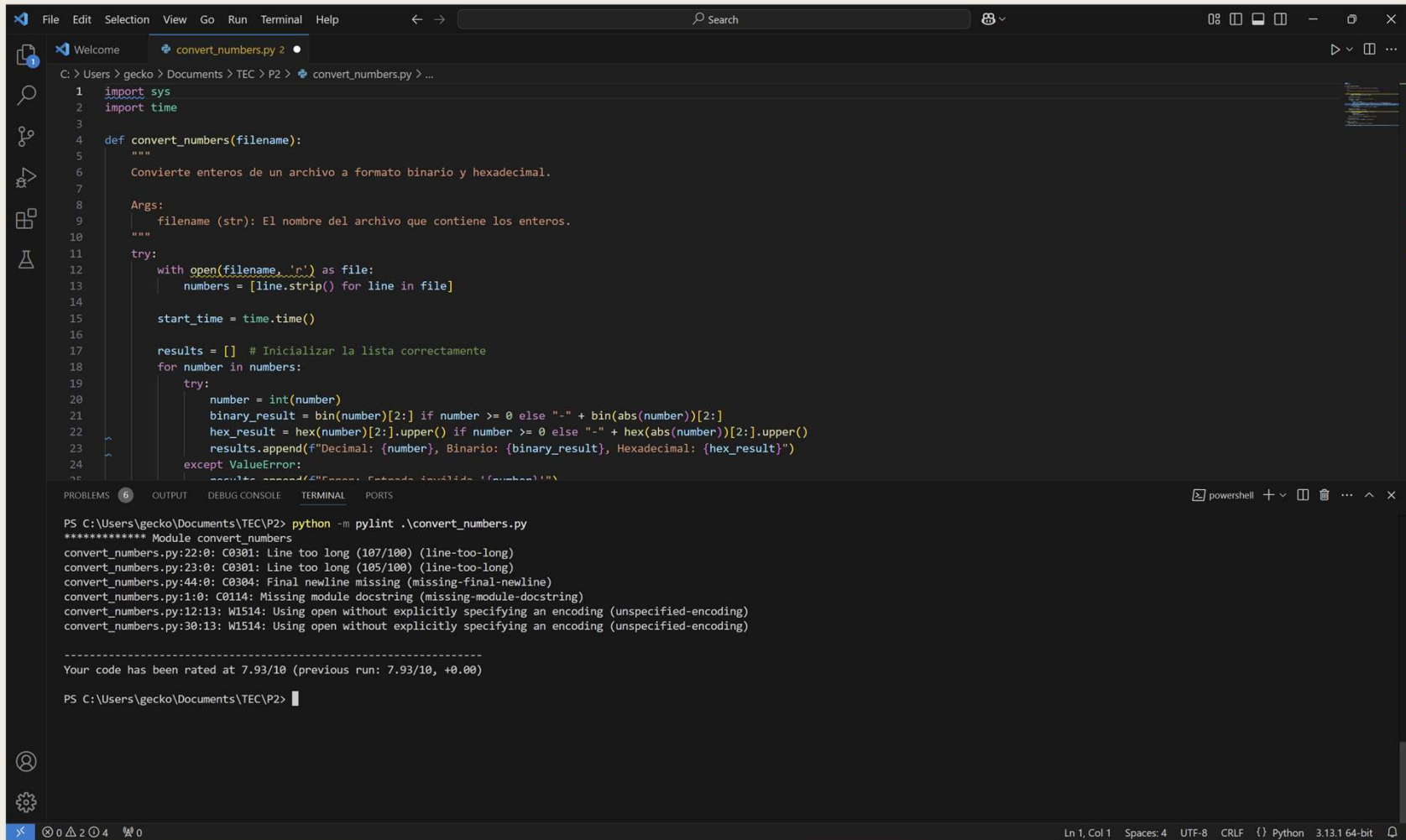
```
84 print(f"Número de registros leídos: {NUM_REGISTROS}")
85
86 inicio = time.time()
87 estadisticas = compute_stats(datos)
88 fin = time.time()
89 tiempo_transcurrido = fin - inicio
90
91 print("Resultados:")
92 for clave, valor in estadisticas.items():
93     print(f"{clave}: {valor}")
94 print(f"Tiempo transcurrido: {tiempo_transcurrido:.4f} segundos")
95
96 with open('StatisticsResults.txt', 'w', encoding='utf-8') as archivo_salida:
97     archivo_salida.write(f"Número de registros leídos: {NUM_REGISTROS}\n")
98     archivo_salida.write("Resultados:\n")
99     for clave, valor in estadisticas.items():
100         archivo_salida.write(f"{clave}: {valor}\n")
101     archivo_salida.write(f"Tiempo transcurrido: {tiempo_transcurrido:.4f} segundos\n")
102
```

Below the editor, the TERMINAL panel shows the output of running the script twice. The first run is for `TC6.txt` and the second for `TC7.txt`.

```
varianza: 21416.70
Tiempo transcurrido: 0.0006 segundos
PS C:\Users\gecko\Documents\TEC\P1> python compute_statistics.py TC6.txt
Número de registros leídos: 3000
Resultados:
media: 187906599279774439766.53
mediana: 188008049965543000000.00
moda: #N/A
desviacion_estandar: 107399951657667618763.83
varianza: 11534749616069341491570449959714916441407.12
Tiempo transcurrido: 0.0035 segundos
PS C:\Users\gecko\Documents\TEC\P1> python compute_statistics.py TC7.txt
Número de registros leídos: 12769
Resultados:
media: 247467395499716241238.72
mediana: 246640973874290000000.00
moda: #N/A
desviacion_estandar: 144611310601233291682.53
varianza: 20912431153806368213126810737543743235418.65
Tiempo transcurrido: 0.0156 segundos
PS C:\Users\gecko\Documents\TEC\P1>
```

The status bar at the bottom indicates the current position is Line 102, Column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, Python 3.13.1 64-bit.

## Evidencia del ejercicio 2 “Converter” al usar Pylint y antes de corregir



The screenshot shows a Visual Studio Code editor window with a Python file named `convert_numbers.py` open. The file is located at `C:\Users\gecko\Documents\TEC\P2\convert_numbers.py`. The code defines a function `convert_numbers(filename)` that reads a file, converts its contents to binary and hexadecimal, and appends the results to a list. The code includes comments in Spanish and uses `try-except` blocks to handle errors.

```
1 import sys
2 import time
3
4 def convert_numbers(filename):
5     """
6     Convierte enteros de un archivo a formato binario y hexadecimal.
7
8     Args:
9         filename (str): El nombre del archivo que contiene los enteros.
10    """
11    try:
12        with open(filename, 'r') as file:
13            numbers = [line.strip() for line in file]
14
15        start_time = time.time()
16
17        results = [] # Inicializar la lista correctamente
18        for number in numbers:
19            try:
20                number = int(number)
21                binary_result = bin(number)[2:] if number >= 0 else "-" + bin(abs(number))[2:]
22                hex_result = hex(number)[2:].upper() if number >= 0 else "-" + hex(abs(number))[2:].upper()
23                results.append(f"Decimal: {number}, Binario: {binary_result}, Hexadecimal: {hex_result}")
24            except ValueError:
25                results.append(f"Error: Entero inválido '{number}'")
```

The bottom panel of the editor shows the output of the command `python -m pylint .\convert_numbers.py`. The output lists several Pylint warnings and errors, including line length issues, missing docstrings, and missing final newlines.

```
PS C:\Users\gecko\Documents\TEC\P2> python -m pylint .\convert_numbers.py
***** Module convert_numbers
convert_numbers.py:22:0: C0301: Line too long (107/100) (line-too-long)
convert_numbers.py:23:0: C0301: Line too long (105/100) (line-too-long)
convert_numbers.py:44:0: C0304: Final newline missing (missing-final-newline)
convert_numbers.py:1:0: C0114: Missing module docstring (missing-module-docstring)
convert_numbers.py:12:13: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
convert_numbers.py:30:13: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)

Your code has been rated at 7.93/10 (previous run: 7.93/10, +0.00)

PS C:\Users\gecko\Documents\TEC\P2>
```

The status bar at the bottom indicates the current file is at line 1, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and is a Python 3.13.1 64-bit file.



## Comprobación del ejercicio 2 “Converter” al 100% con Pylint

The image shows a Windows IDE with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The top toolbar has icons for search, save, and other standard IDE functions. The main editor window displays a Python script named `convert_numbers.py`. The script defines a function `convert_numbers(filename)` that takes a filename as input and returns a list of results. The results are formatted as a dictionary with keys 'Decimal', 'Binario', and 'Hexadecimal'. The script also includes error handling for `ValueError` and `FileNotFoundError`. The script is executed from the command line using `python -m pylint .\convert_numbers.py`. The output of the script is displayed in the terminal window at the bottom. The terminal shows the command prompt, the command to run the script, and the output of the script, which includes the results of the conversion for the input file `convert_numbers.py`.

```
8 def convert_numbers(filename):
26     hex_result = (
27         hex(num)[2:].upper() if num >= 0 else "-" + hex(abs(num))[2:].upper()
28     )
29     results.append(
30         f"Decimal: {num}, Binario: {binary_result}, Hexadecimal: {hex_result}"
31     )
32 except ValueError:
33     results.append(f"Error: Entrada inválida '{number}'")
34
35 end_time = time.time()
36 elapsed_time = end_time - start_time
37
38 with open('ConversionResults.txt', 'w', encoding='utf-8') as output_file:
39     for result in results:
40         print(result)
41         output_file.write(result + '\n')
42
43 print(f"Tiempo transcurrido: {elapsed_time:.4f} segundos")
44
45 except FileNotFoundError:
46     print(f"Error: Archivo '{filename}' no encontrado.")
47
48 if __name__ == "__main__":
49     if len(sys.argv) != 2:
50         print("Uso: python convert_numbers.py <filename>")
51     else:
52         convert_numbers(sys.argv[1])
53
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gecko\Documents\TEC\P2> python -m pylint .\convert\_numbers.py

\*\*\*\*\* Module convert\_numbers

convert\_numbers.py:27:0: C0301: Line too long (102/100) (line-too-long)

convert\_numbers.py:40:0: C0303: Trailing whitespace (trailing-whitespace)

-----

Your code has been rated at 9.31/10 (previous run: 7.93/10, +1.38)

PS C:\Users\gecko\Documents\TEC\P2> python -m pylint .\convert\_numbers.py

-----

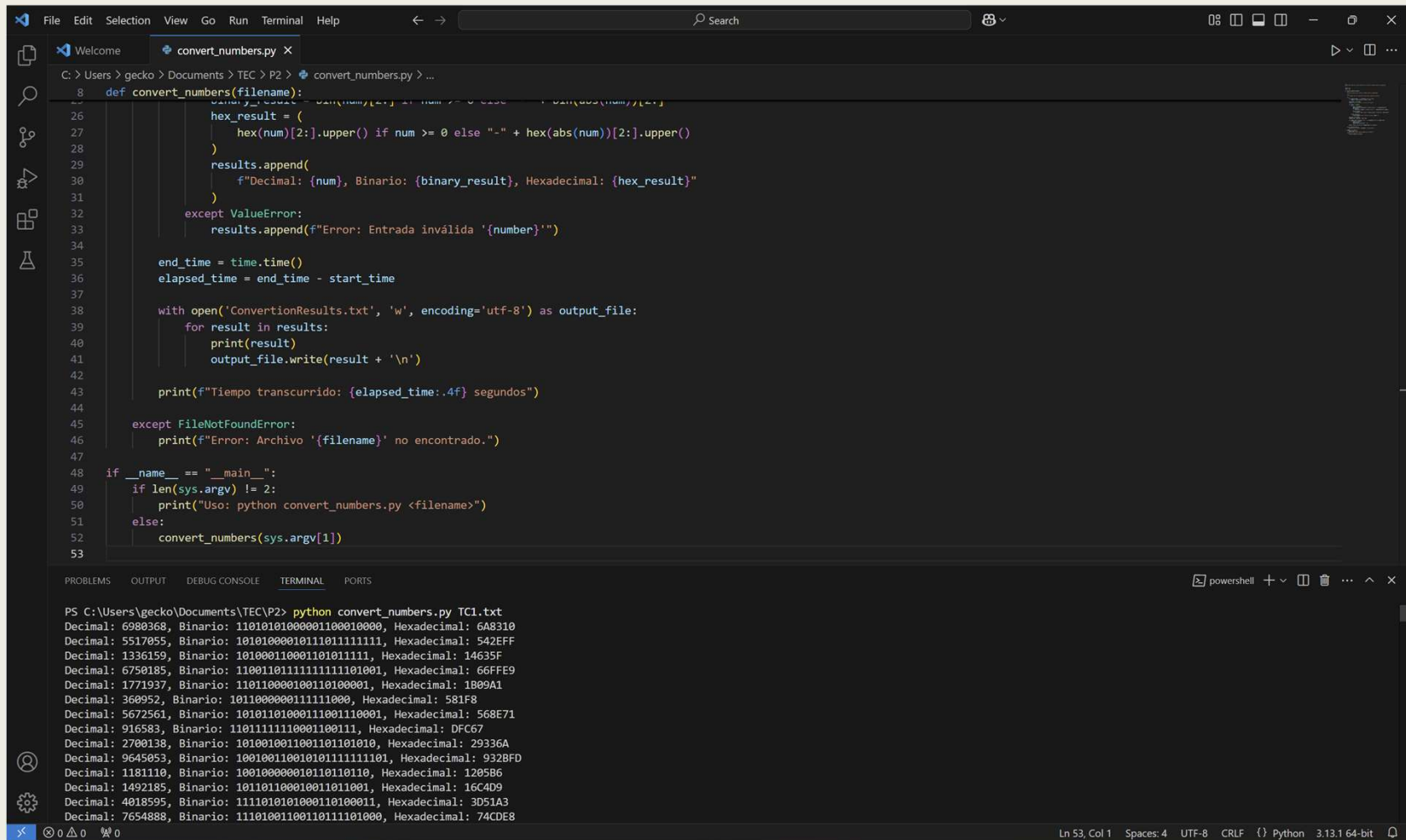
Your code has been rated at 10.00/10 (previous run: 9.31/10, +0.69)

PS C:\Users\gecko\Documents\TEC\P2>

Ln 53, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.13.1 64-bit



## Problema 2 corregido con Pylint: Ejecución ejercicio TC1



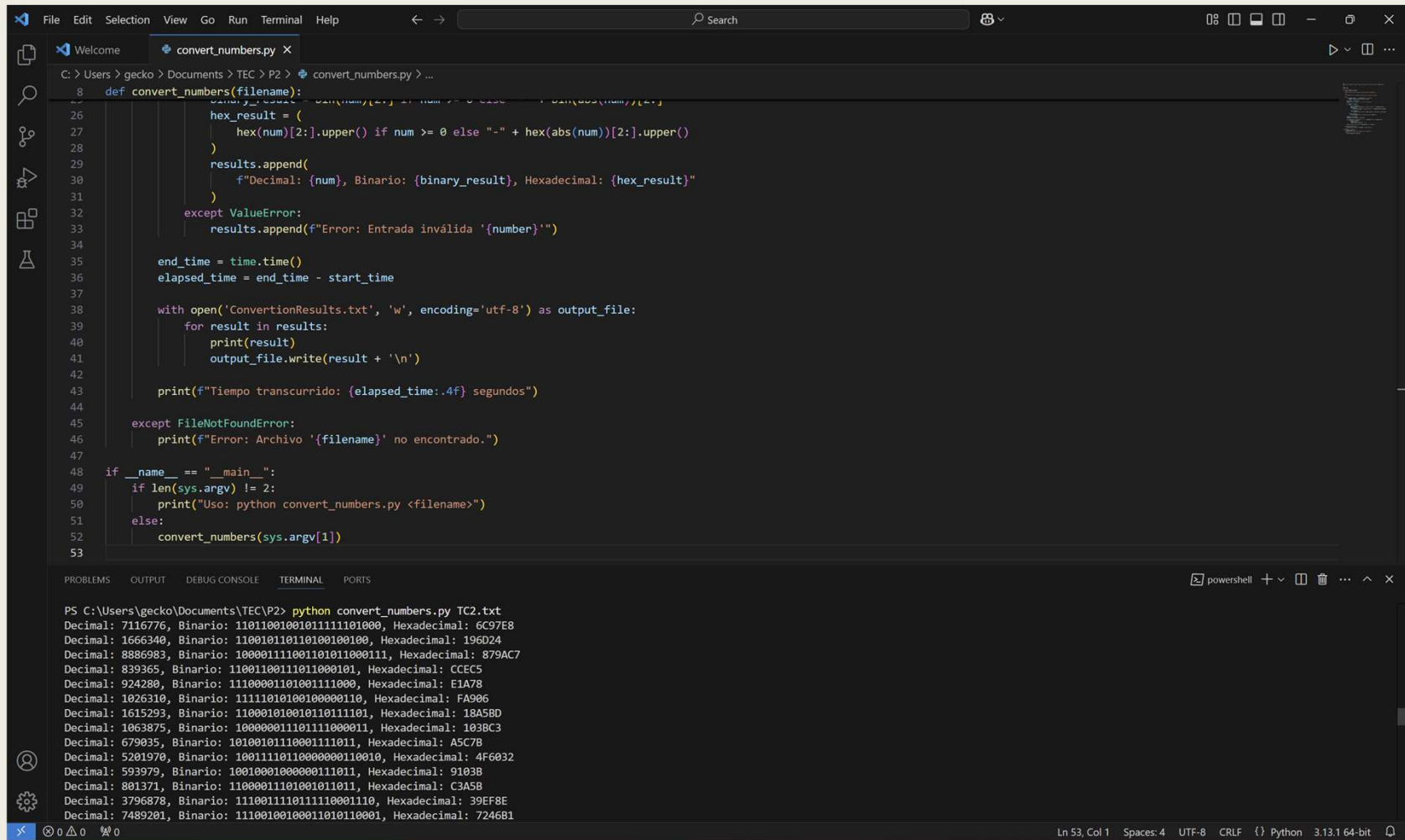
The image shows a Visual Studio Code editor window with a Python script named `convert_numbers.py` open. The script is designed to convert decimal numbers to binary and hexadecimal. It includes a function `convert_numbers(filename)` that takes a filename as input, processes a list of numbers, and writes the results to a file. The script also includes a main block that checks for command-line arguments and calls the `convert_numbers` function.

```
8 def convert_numbers(filename):
26     hex_result = (
27         hex(num)[2:].upper() if num >= 0 else "-" + hex(abs(num))[2:].upper()
28     )
29     results.append(
30         f"Decimal: {num}, Binario: {binary_result}, Hexadecimal: {hex_result}"
31     )
32 except ValueError:
33     results.append(f"Error: Entrada inválida '{number}'")
34
35 end_time = time.time()
36 elapsed_time = end_time - start_time
37
38 with open('ConversionResults.txt', 'w', encoding='utf-8') as output_file:
39     for result in results:
40         print(result)
41         output_file.write(result + '\n')
42
43 print(f"Tiempo transcurrido: {elapsed_time:.4f} segundos")
44
45 except FileNotFoundError:
46     print(f"Error: Archivo '{filename}' no encontrado.")
47
48 if __name__ == "__main__":
49     if len(sys.argv) != 2:
50         print("Uso: python convert_numbers.py <filename>")
51     else:
52         convert_numbers(sys.argv[1])
53
```

The terminal output shows the execution of the script with the command `python convert_numbers.py TC1.txt`. The output displays a list of decimal numbers, their corresponding binary representations, and their hexadecimal representations, all formatted as specified in the script.

```
PS C:\Users\gecko\Documents\TEC\P2> python convert_numbers.py TC1.txt
Decimal: 6980368, Binario: 1101010100001100010000, Hexadecimal: 6A8310
Decimal: 5517055, Binario: 1010100001011101111111, Hexadecimal: 542EFF
Decimal: 1336159, Binario: 10100011000110101111, Hexadecimal: 14635F
Decimal: 6750185, Binario: 11001101111111111101001, Hexadecimal: 66FFE9
Decimal: 1771937, Binario: 110110000100110100001, Hexadecimal: 1809A1
Decimal: 360952, Binario: 1011000000111111000, Hexadecimal: 581F8
Decimal: 5672561, Binario: 10101101000111001110001, Hexadecimal: 568E71
Decimal: 916583, Binario: 11011111110001100111, Hexadecimal: DFC67
Decimal: 2700138, Binario: 101001001100110101010, Hexadecimal: 29336A
Decimal: 9645053, Binario: 100100110010101111111101, Hexadecimal: 932BFD
Decimal: 1181110, Binario: 1001000000101101110, Hexadecimal: 120586
Decimal: 1492185, Binario: 101101100010011011001, Hexadecimal: 16C4D9
Decimal: 4018595, Binario: 1111010101000110100011, Hexadecimal: 3D51A3
Decimal: 7654888, Binario: 11101001100110111101000, Hexadecimal: 74CDE8
```

## Problema 2 corregido con Pylint: Ejecución ejercicio TC2



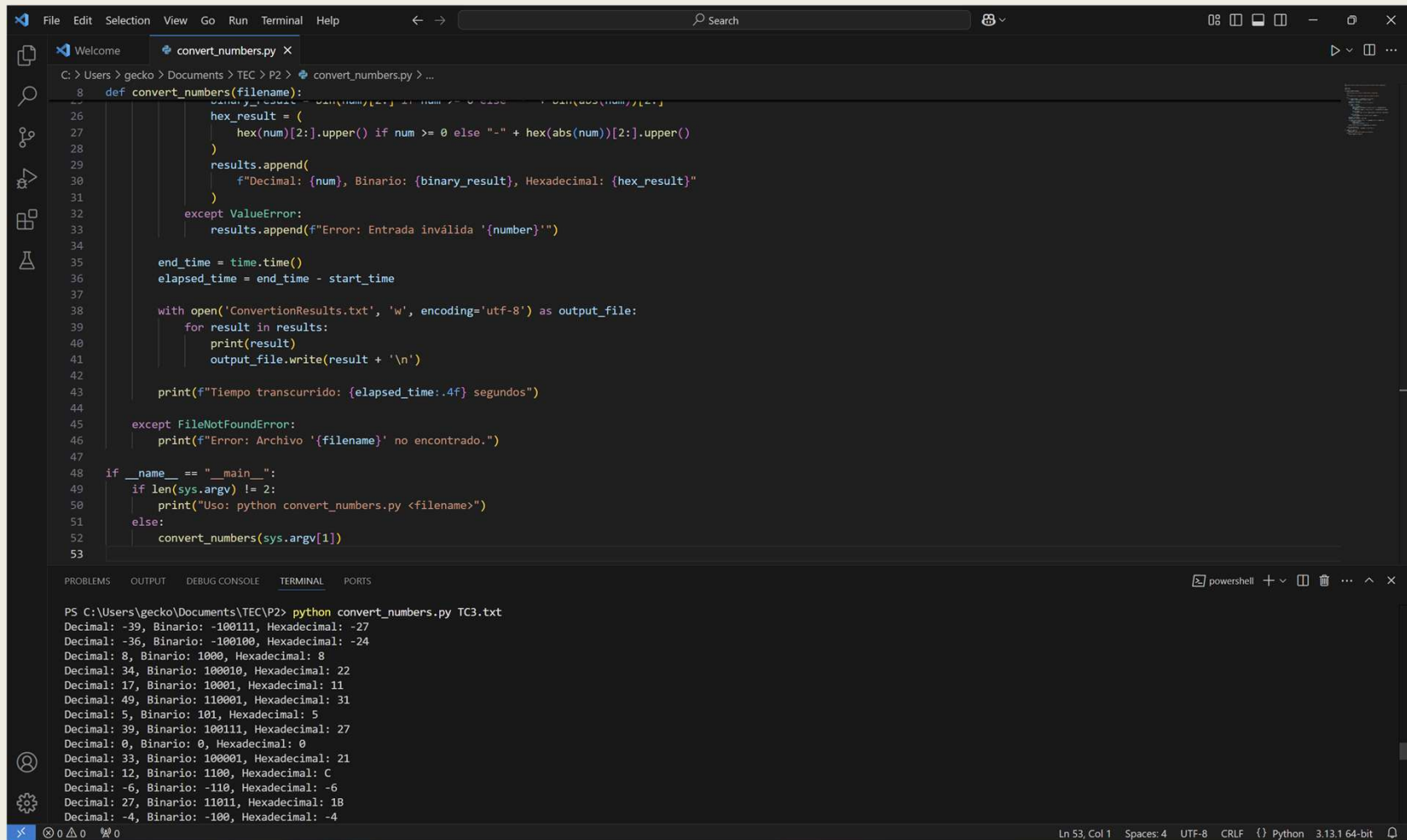
The image shows a Visual Studio Code editor window with a Python script named `convert_numbers.py` open. The script is designed to convert decimal numbers to binary and hexadecimal. It includes error handling for invalid inputs and file not found errors. The script is executed in a terminal window, and the output shows a list of decimal numbers and their corresponding binary and hexadecimal representations.

```
8 def convert_numbers(filename):
26     hex_result = (
27         hex(num)[2:].upper() if num >= 0 else "-" + hex(abs(num))[2:].upper()
28     )
29     results.append(
30         f"Decimal: {num}, Binario: {binary_result}, Hexadecimal: {hex_result}"
31     )
32 except ValueError:
33     results.append(f"Error: Entrada inválida '{number}'")
34
35 end_time = time.time()
36 elapsed_time = end_time - start_time
37
38 with open('ConversionResults.txt', 'w', encoding='utf-8') as output_file:
39     for result in results:
40         print(result)
41         output_file.write(result + '\n')
42
43 print(f"Tiempo transcurrido: {elapsed_time:.4f} segundos")
44
45 except FileNotFoundError:
46     print(f"Error: Archivo '{filename}' no encontrado.")
47
48 if __name__ == "__main__":
49     if len(sys.argv) != 2:
50         print("Uso: python convert_numbers.py <filename>")
51     else:
52         convert_numbers(sys.argv[1])
53
```

The terminal output shows the execution of the script with the following results:

```
PS C:\Users\gecko\Documents\TEC\P2> python convert_numbers.py TC2.txt
Decimal: 7116776, Binario: 1101100100101111101000, Hexadecimal: 6C97E8
Decimal: 1666340, Binario: 110010110110100100100, Hexadecimal: 196D24
Decimal: 8886983, Binario: 100001111001101011000111, Hexadecimal: 879AC7
Decimal: 839365, Binario: 11001100111011000101, Hexadecimal: CCEC5
Decimal: 924280, Binario: 11100001101001111000, Hexadecimal: E1A78
Decimal: 1026310, Binario: 11111010100100000110, Hexadecimal: FA906
Decimal: 1615293, Binario: 110001010010110111101, Hexadecimal: 18A5B0
Decimal: 1063875, Binario: 100000011101111000011, Hexadecimal: 103BC3
Decimal: 679035, Binario: 10100101110001111011, Hexadecimal: A5C7B
Decimal: 5201970, Binario: 10011110110000000110010, Hexadecimal: 4F6032
Decimal: 592979, Binario: 1001000100000011011, Hexadecimal: 9103B
Decimal: 801371, Binario: 11000011101001011011, Hexadecimal: C3A5B
Decimal: 3796878, Binario: 11100111011110001110, Hexadecimal: 39EF8E
Decimal: 7489201, Binario: 11100100100011010110001, Hexadecimal: 7246B1
```

## Problema 2 corregido con Pylint: Ejecución ejercicio TC3



The image shows a Visual Studio Code editor window with a Python file named `convert_numbers.py` open. The script is designed to convert decimal numbers to binary and hexadecimal. It includes error handling for invalid input and file not found. The terminal at the bottom shows the execution of the script with the command `python convert_numbers.py TC3.txt`, resulting in a list of conversions for various decimal values.

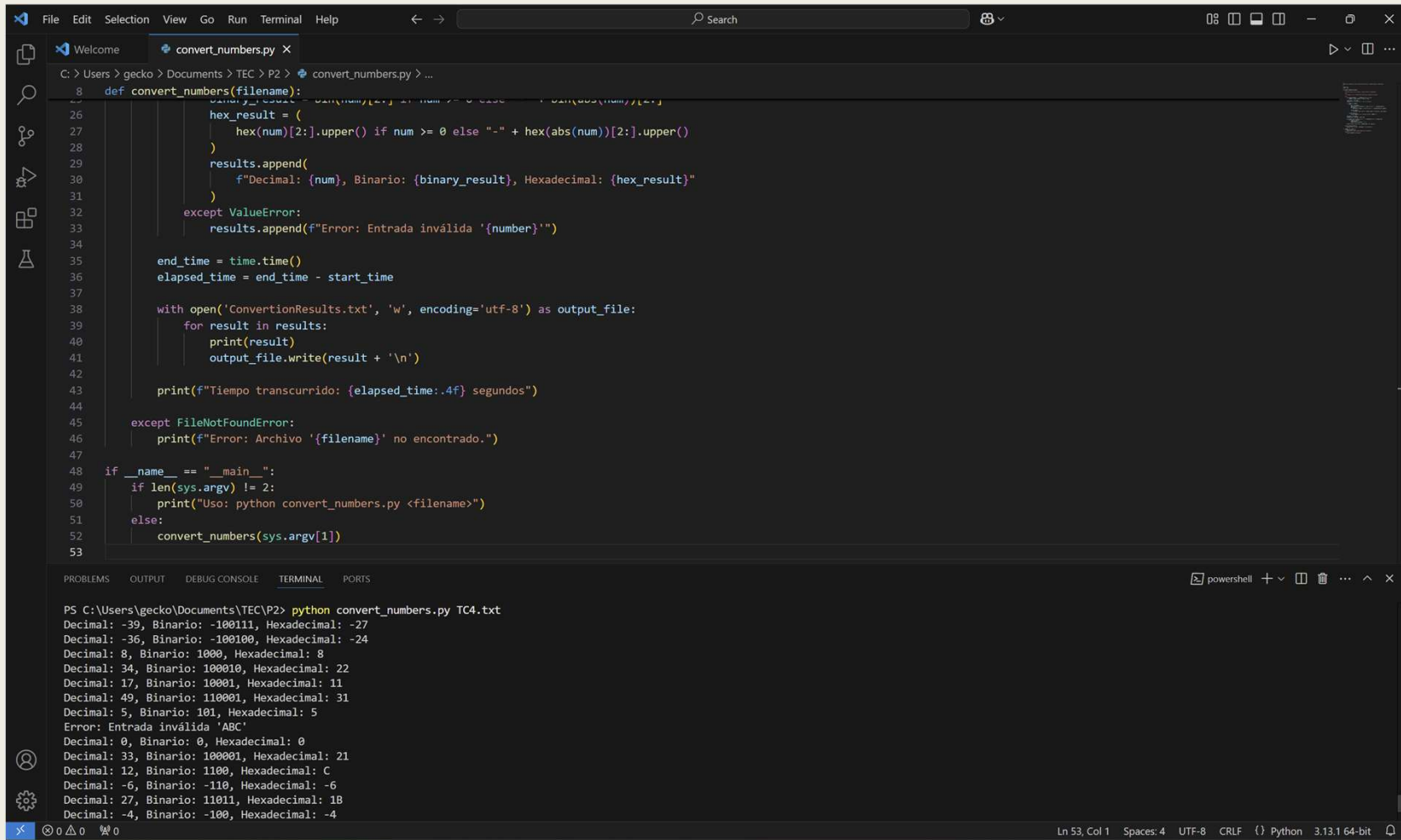
```
8 def convert_numbers(filename):
26     results = []
27     for num in open(filename, 'r').read().split():
28         hex_result = hex(num)[2:].upper() if num >= 0 else "-" + hex(abs(num))[2:].upper()
29         results.append(f"Decimal: {num}, Binario: {bin(num)[2:].upper()}, Hexadecimal: {hex_result}")
30     except ValueError:
31         results.append(f"Error: Entrada inválida '{number}'")
32
33     end_time = time.time()
34     elapsed_time = end_time - start_time
35
36     with open('ConversionResults.txt', 'w', encoding='utf-8') as output_file:
37         for result in results:
38             print(result)
39             output_file.write(result + '\n')
40
41     print(f"Tiempo transcurrido: {elapsed_time:.4f} segundos")
42
43 except FileNotFoundError:
44     print(f"Error: Archivo '{filename}' no encontrado.")
45
46 if __name__ == "__main__":
47     if len(sys.argv) != 2:
48         print("Uso: python convert_numbers.py <filename>")
49     else:
50         convert_numbers(sys.argv[1])
51
52
53
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\gecko\Documents\TEC\P2> python convert_numbers.py TC3.txt
Decimal: -39, Binario: -100111, Hexadecimal: -27
Decimal: -36, Binario: -100100, Hexadecimal: -24
Decimal: 8, Binario: 1000, Hexadecimal: 8
Decimal: 34, Binario: 100010, Hexadecimal: 22
Decimal: 17, Binario: 10001, Hexadecimal: 11
Decimal: 49, Binario: 110001, Hexadecimal: 31
Decimal: 5, Binario: 101, Hexadecimal: 5
Decimal: 39, Binario: 100111, Hexadecimal: 27
Decimal: 0, Binario: 0, Hexadecimal: 0
Decimal: 33, Binario: 100001, Hexadecimal: 21
Decimal: 12, Binario: 1100, Hexadecimal: C
Decimal: -6, Binario: -110, Hexadecimal: -6
Decimal: 27, Binario: 11011, Hexadecimal: 1B
Decimal: -4, Binario: -100, Hexadecimal: -4
```

Ln 53, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.13.1 64-bit

## Problema 2 corregido con Pylint: Ejecución ejercicio TC4



The image shows a Visual Studio Code editor window with a Python script named `convert_numbers.py` and its execution output in the terminal.

**Python Script (`convert_numbers.py`):**

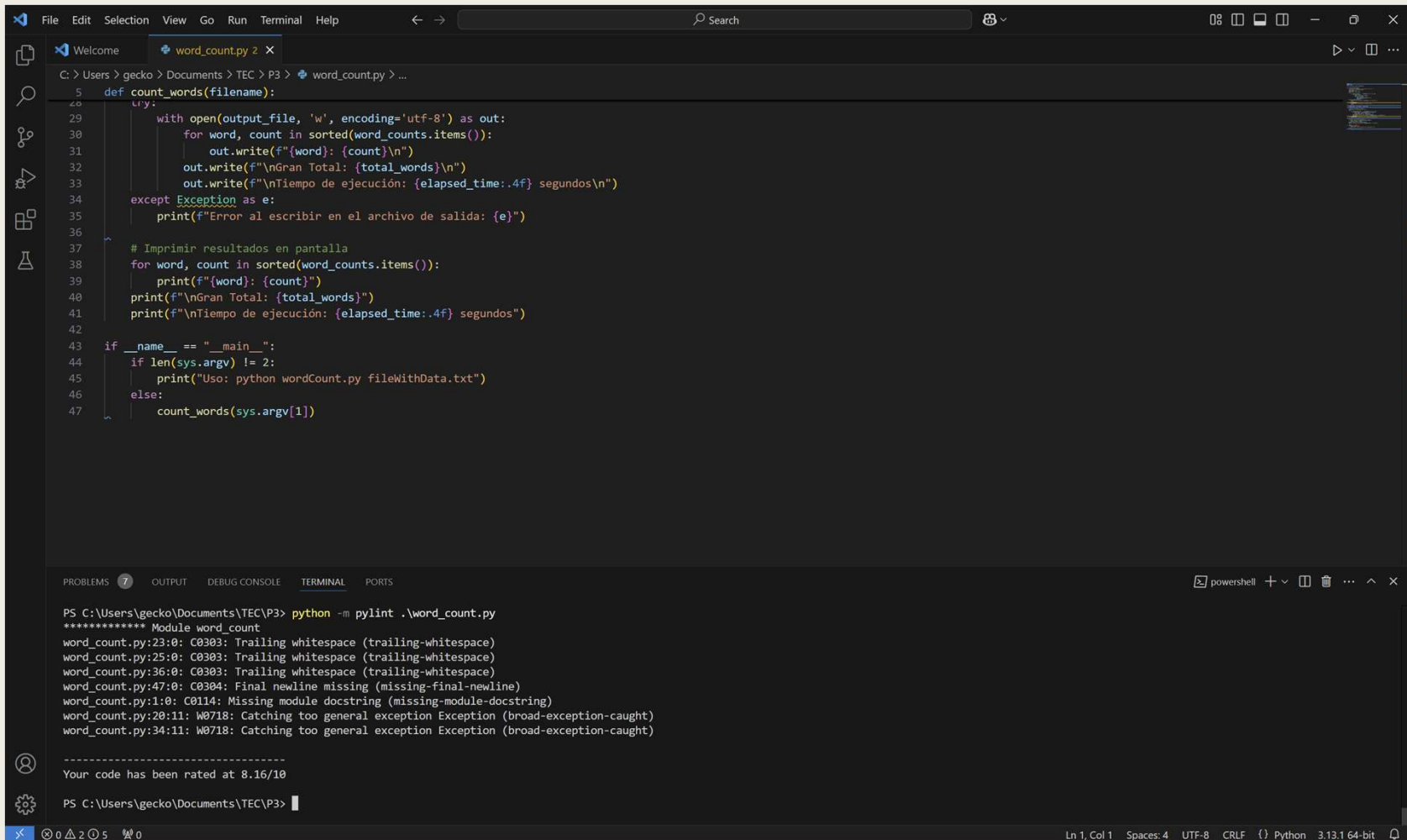
```
8 def convert_numbers(filename):
26     binary_result = ""
27     hex_result = ""
28     hex(num)[2:].upper() if num >= 0 else "-" + hex(abs(num))[2:].upper()
29     results.append(
30         f"Decimal: {num}, Binario: {binary_result}, Hexadecimal: {hex_result}"
31     )
32 except ValueError:
33     results.append(f"Error: Entrada inválida '{number}'")
34
35 end_time = time.time()
36 elapsed_time = end_time - start_time
37
38 with open('ConversionResults.txt', 'w', encoding='utf-8') as output_file:
39     for result in results:
40         print(result)
41         output_file.write(result + '\n')
42
43 print(f"Tiempo transcurrido: {elapsed_time:.4f} segundos")
44
45 except FileNotFoundError:
46     print(f"Error: Archivo '{filename}' no encontrado.")
47
48 if __name__ == "__main__":
49     if len(sys.argv) != 2:
50         print("Uso: python convert_numbers.py <filename>")
51     else:
52         convert_numbers(sys.argv[1])
53
```

**Terminal Output:**

```
PS C:\Users\gecko\Documents\TEC\P2> python convert_numbers.py TC4.txt
Decimal: -39, Binario: -100111, Hexadecimal: -27
Decimal: -36, Binario: -100100, Hexadecimal: -24
Decimal: 8, Binario: 1000, Hexadecimal: 8
Decimal: 34, Binario: 100010, Hexadecimal: 22
Decimal: 17, Binario: 10001, Hexadecimal: 11
Decimal: 49, Binario: 110001, Hexadecimal: 31
Decimal: 5, Binario: 101, Hexadecimal: 5
Error: Entrada inválida 'ABC'
Decimal: 0, Binario: 0, Hexadecimal: 0
Decimal: 33, Binario: 100001, Hexadecimal: 21
Decimal: 12, Binario: 1100, Hexadecimal: C
Decimal: -6, Binario: -110, Hexadecimal: -6
Decimal: 27, Binario: 11011, Hexadecimal: 1B
Decimal: -4, Binario: -100, Hexadecimal: -4
```

The terminal output shows the script successfully converting various decimal numbers to binary and hexadecimal. It also handles an invalid input 'ABC' with an error message. The script is executed from the command prompt using the command `python convert_numbers.py TC4.txt`.

## Evidencia del ejercicio 3 “Count Words” al usar Pylint y antes de corregir



The screenshot shows a Visual Studio Code editor window with a Python file named `word_count.py` open. The file contains a function `count_words` that takes a filename as input, reads the file, counts the words, and writes the results to an output file. The code is as follows:

```
5 def count_words(filename):
6     try:
7         with open(output_file, 'w', encoding='utf-8') as out:
8             for word, count in sorted(word_counts.items()):
9                 out.write(f"{word}: {count}\n")
10            out.write(f"\nGran Total: {total_words}\n")
11            out.write(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos\n")
12        except Exception as e:
13            print(f"Error al escribir en el archivo de salida: {e}")
14
15        # Imprimir resultados en pantalla
16        for word, count in sorted(word_counts.items()):
17            print(f"{word}: {count}")
18        print(f"\nGran Total: {total_words}")
19        print(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos")
20
21    if __name__ == "__main__":
22        if len(sys.argv) != 2:
23            print("Uso: python wordCount.py fileWithData.txt")
24        else:
25            count_words(sys.argv[1])
```

The bottom panel of the editor shows the Pylint output in the terminal. The output indicates that the code has been rated at 8.16/10 and lists several warnings:

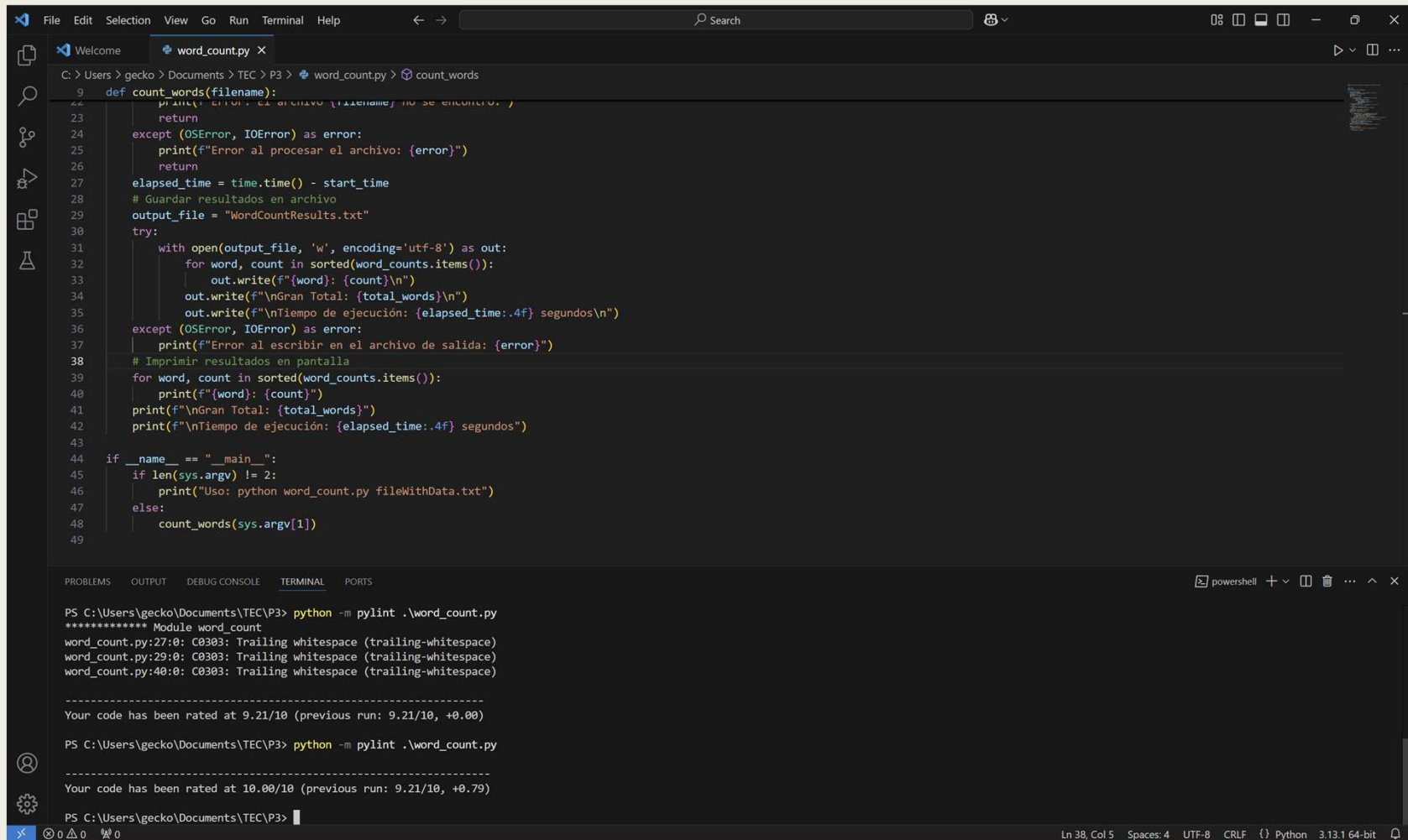
```
PS C:\Users\gecko\Documents\TEC\P3> python -m pylint .\word_count.py
***** Module word_count
word_count.py:23:0: C0303: Trailing whitespace (trailing-whitespace)
word_count.py:25:0: C0303: Trailing whitespace (trailing-whitespace)
word_count.py:36:0: C0303: Trailing whitespace (trailing-whitespace)
word_count.py:47:0: C0304: Final newline missing (missing-final-newline)
word_count.py:1:0: C0114: Missing module docstring (missing-module-docstring)
word_count.py:20:11: W0718: Catching too general exception Exception (broad-exception-caught)
word_count.py:34:11: W0718: Catching too general exception Exception (broad-exception-caught)

-----
Your code has been rated at 8.16/10

PS C:\Users\gecko\Documents\TEC\P3>
```

The status bar at the bottom of the editor shows the following information: Ln 1, Col 1, Spaces: 4, UTF-8, CRLF, Python, 3.13.1 64-bit.

## Comprobación del ejercicio 3 “Count Words” al 100% con Pylint



```
File Edit Selection View Go Run Terminal Help
word_count.py X
C:\Users\gecko> Documents\TEC\P3> word_count.py > count_words
9 def count_words(filename):
23     return
24 except (OSError, IOError) as error:
25     print(f"Error al procesar el archivo: {error}")
26     return
27 elapsed_time = time.time() - start_time
28 # Guardar resultados en archivo
29 output_file = "WordCountResults.txt"
30 try:
31     with open(output_file, 'w', encoding='utf-8') as out:
32         for word, count in sorted(word_counts.items()):
33             out.write(f"{word}: {count}\n")
34             out.write(f"\nGran Total: {total_words}\n")
35             out.write(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos\n")
36 except (OSError, IOError) as error:
37     print(f"Error al escribir en el archivo de salida: {error}")
38 # Imprimir resultados en pantalla
39 for word, count in sorted(word_counts.items()):
40     print(f"{word}: {count}")
41 print(f"\nGran Total: {total_words}")
42 print(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos")
43
44 if __name__ == "__main__":
45     if len(sys.argv) != 2:
46         print("Uso: python word_count.py fileWithData.txt")
47     else:
48         count_words(sys.argv[1])
49
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\gecko\Documents\TEC\P3> python -m pylint .\word_count.py
***** Module word_count
word_count.py:27:0: C0303: Trailing whitespace (trailing-whitespace)
word_count.py:29:0: C0303: Trailing whitespace (trailing-whitespace)
word_count.py:40:0: C0303: Trailing whitespace (trailing-whitespace)

-----
Your code has been rated at 9.21/10 (previous run: 9.21/10, +0.00)

PS C:\Users\gecko\Documents\TEC\P3> python -m pylint .\word_count.py

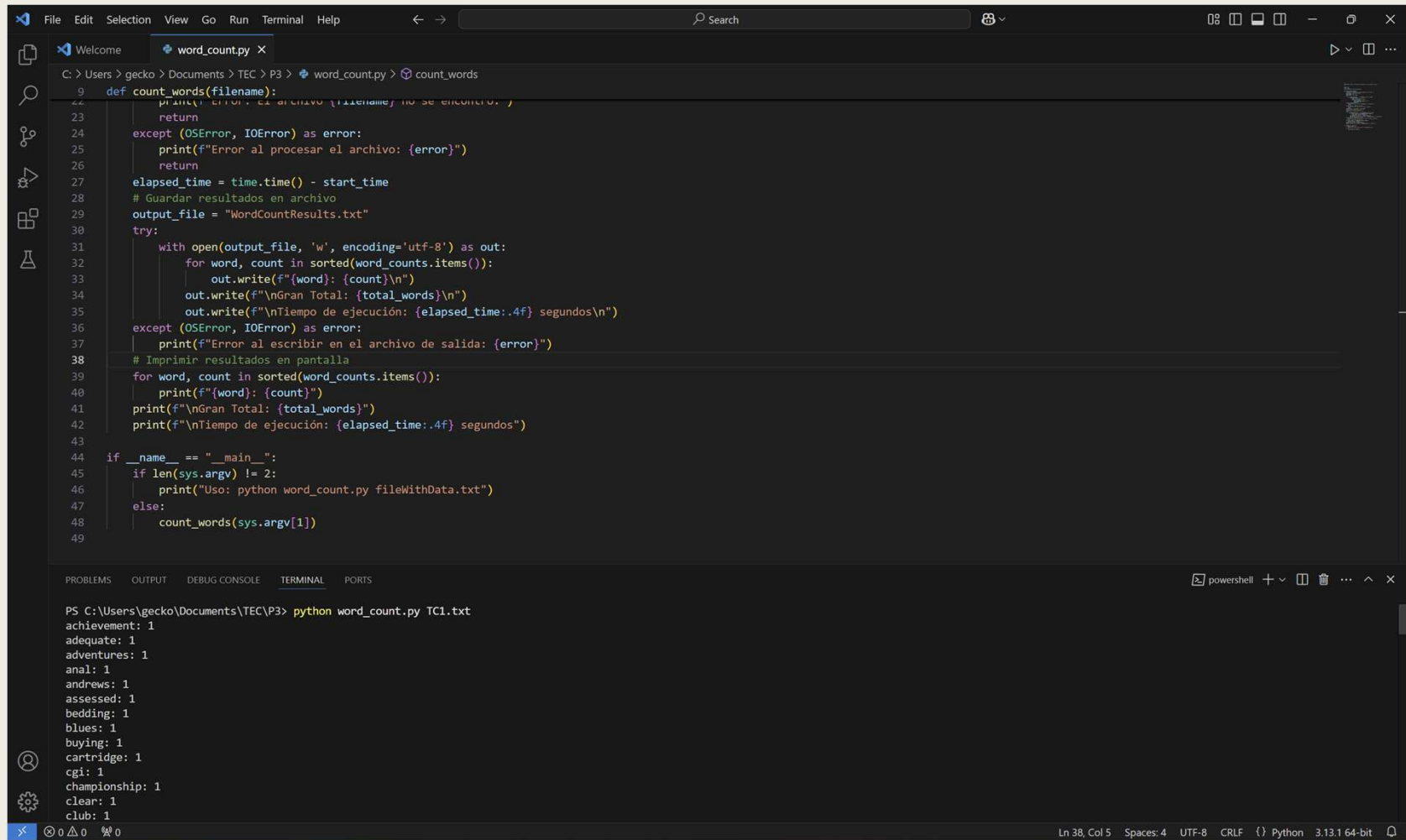
-----
Your code has been rated at 10.00/10 (previous run: 9.21/10, +0.79)

PS C:\Users\gecko\Documents\TEC\P3>
```

Ln 38, Col 5 Spaces: 4 UTF-8 CRLF {} Python 3.13.1 64-bit



## Problema 3 corregido con Pylint: Ejecución ejercicio TC1



The image shows a Visual Studio Code editor window with a Python file named `word_count.py` open. The file is located at `C:\Users\gecko\Documents\TEC\P3\word_count.py`. The code defines a function `count_words(filename)` that reads a file, counts the words, and writes the results to `WordCountResults.txt`. It also prints the results to the console. The script is executed from the terminal using the command `python word_count.py TC1.txt`. The output shows a list of words and their counts, such as `achievement: 1`, `adequate: 1`, etc.

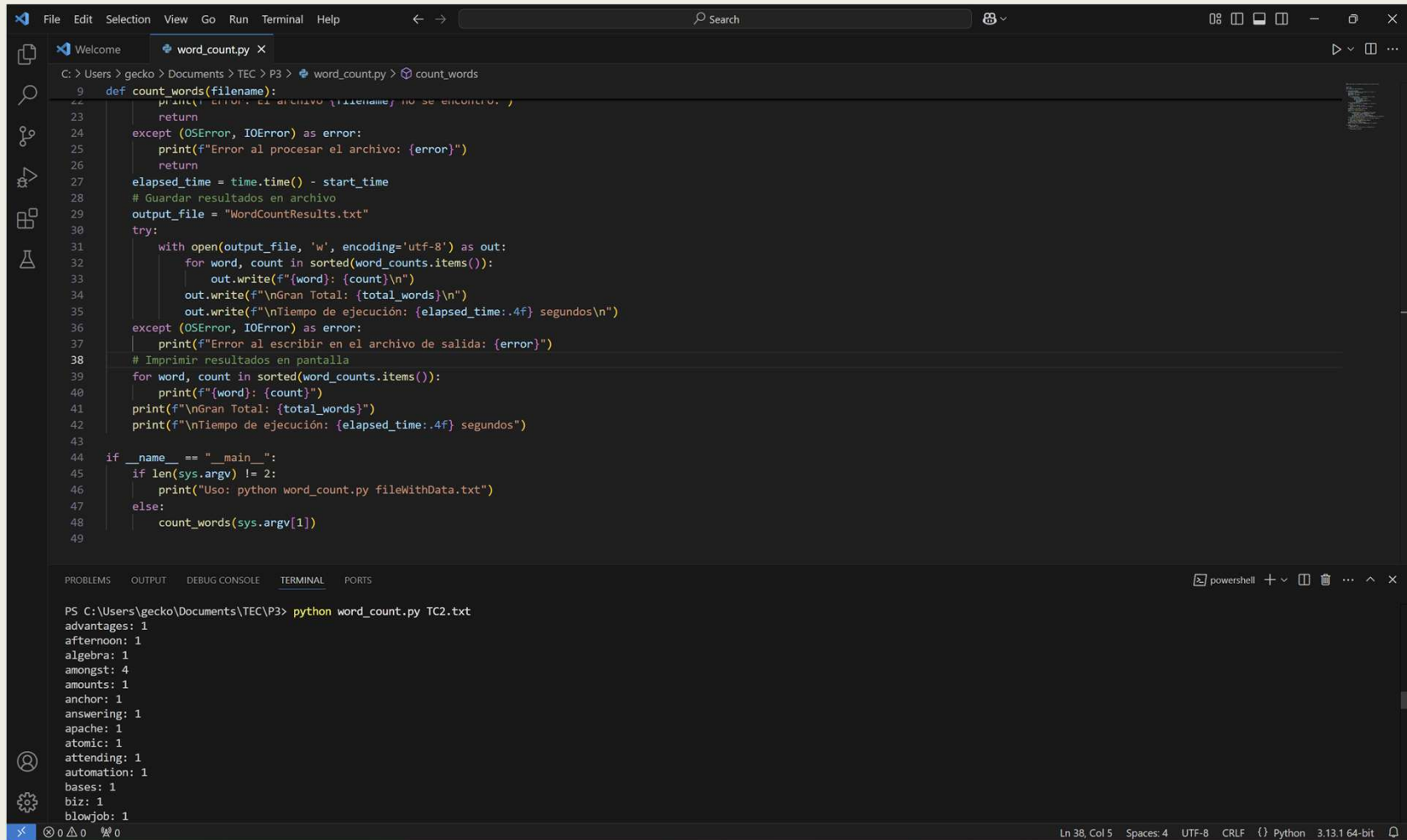
```
9 def count_words(filename):
10     try:
11         with open(filename, 'r', encoding='utf-8') as f:
12             word_counts = {}
13             for line in f:
14                 words = line.split()
15                 for word in words:
16                     word_counts[word] = word_counts.get(word, 0) + 1
17             return word_counts
18     except (OSError, IOError) as error:
19         print(f"Error al procesar el archivo: {error}")
20         return {}
21
22     elapsed_time = time.time() - start_time
23     # Guardar resultados en archivo
24     output_file = "WordCountResults.txt"
25     try:
26         with open(output_file, 'w', encoding='utf-8') as out:
27             for word, count in sorted(word_counts.items()):
28                 out.write(f"{word}: {count}\n")
29             out.write(f"\nGran Total: {total_words}\n")
30             out.write(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos\n")
31     except (OSError, IOError) as error:
32         print(f"Error al escribir en el archivo de salida: {error}")
33
34     # Imprimir resultados en pantalla
35     for word, count in sorted(word_counts.items()):
36         print(f"{word}: {count}")
37     print(f"\nGran Total: {total_words}")
38     print(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos")
39
40 if __name__ == "__main__":
41     if len(sys.argv) != 2:
42         print("Uso: python word_count.py fileWithData.txt")
43     else:
44         count_words(sys.argv[1])
45
46
```

Terminal Output:

```
PS C:\Users\gecko\Documents\TEC\P3> python word_count.py TC1.txt
achievement: 1
adequate: 1
adventures: 1
anal: 1
andrews: 1
assessed: 1
bedding: 1
blues: 1
buying: 1
cartridge: 1
cgi: 1
championship: 1
clear: 1
club: 1
```



## Problema 3 corregido con Pylint: Ejecución ejercicio TC2



The image shows a Visual Studio Code editor window with a Python file named `word_count.py` open. The file contains a function `count_words` that processes a text file and prints the results. The script is executed in a terminal window, showing the output for the file `TC2.txt`.

```
def count_words(filename):
    try:
        with open(filename, 'r', encoding='utf-8') as f:
            word_counts = {}
            for line in f:
                words = line.split()
                for word in words:
                    word_counts[word] = word_counts.get(word, 0) + 1
            return word_counts
    except (OSError, IOError) as error:
        print(f"Error al procesar el archivo: {error}")
        return {}

    elapsed_time = time.time() - start_time
    # Guardar resultados en archivo
    output_file = "WordCountResults.txt"
    try:
        with open(output_file, 'w', encoding='utf-8') as out:
            for word, count in sorted(word_counts.items()):
                out.write(f"{word}: {count}\n")
            out.write(f"\nGran Total: {total_words}\n")
            out.write(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos\n")
    except (OSError, IOError) as error:
        print(f"Error al escribir en el archivo de salida: {error}")

    # Imprimir resultados en pantalla
    for word, count in sorted(word_counts.items()):
        print(f"{word}: {count}")
    print(f"\nGran Total: {total_words}")
    print(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos")

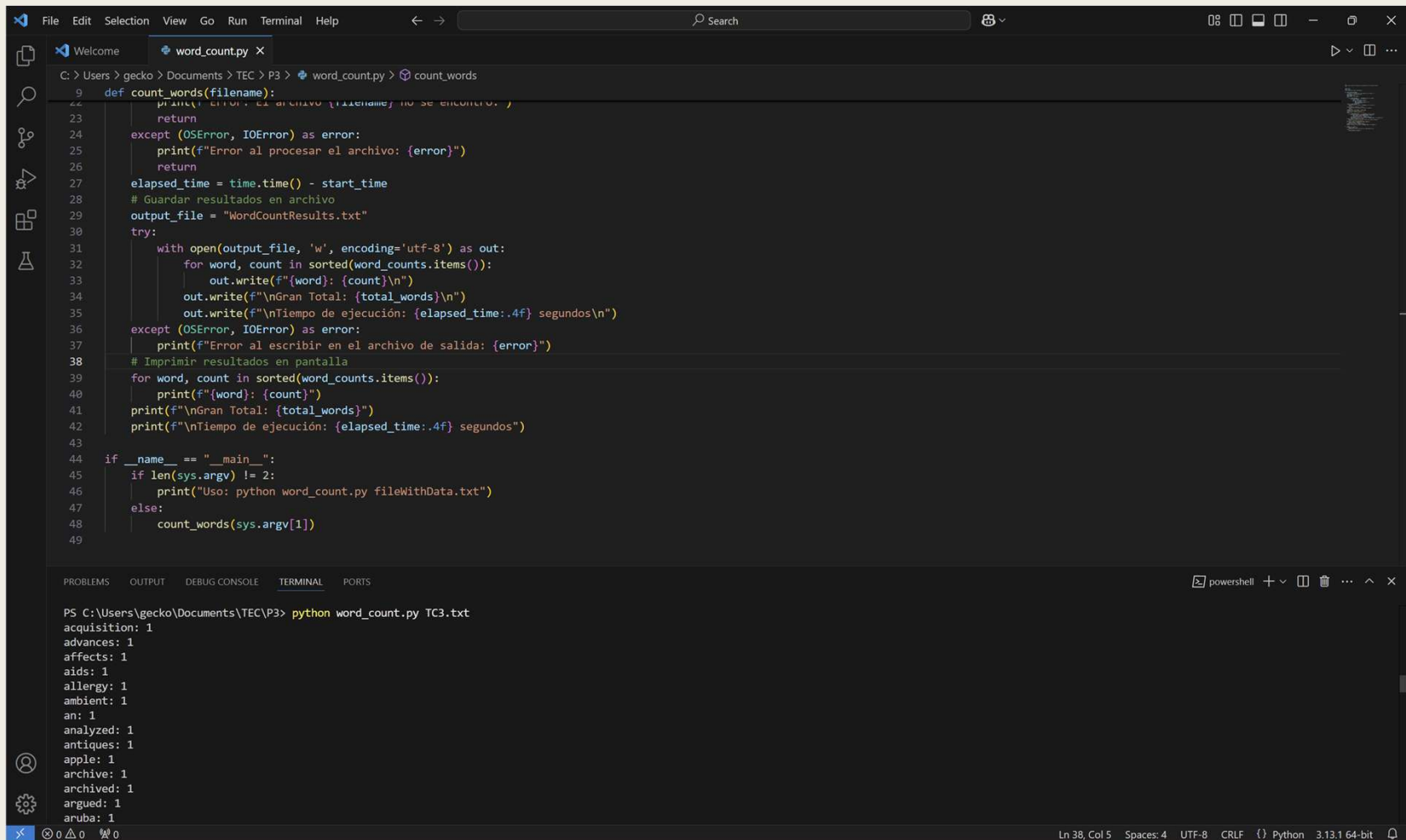
if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("Uso: python word_count.py fileWithData.txt")
    else:
        count_words(sys.argv[1])
```

Terminal Output:

```
PS C:\Users\gecko\Documents\TEC\P3> python word_count.py TC2.txt
advantages: 1
afternoon: 1
algebra: 1
amongst: 4
amounts: 1
anchor: 1
answering: 1
apache: 1
atomic: 1
attending: 1
automation: 1
bases: 1
biz: 1
blowjob: 1
```

Ln 38, Col 5 Spaces: 4 UTF-8 CRLF {} Python 3.13.1 64-bit

## Problema 3 corregido con Pylint: Ejecución ejercicio TC3



The image shows a Visual Studio Code editor window with a Python file named `word_count.py` open. The file is located at `C:\Users\gecko\Documents\TEC\P3\word_count.py`. The script defines a `count_words` function that takes a filename as input, counts the words in the file, and writes the results to `WordCountResults.txt`. It also prints the results to the console. The script includes error handling for `OSError` and `IOError`.

```
def count_words(filename):
    """Función que cuenta las palabras de un archivo y las guarda en un archivo de texto.
    """
    try:
        with open(filename, 'r', encoding='utf-8') as f:
            word_counts = {}
            for line in f:
                words = line.split()
                for word in words:
                    word_counts[word] = word_counts.get(word, 0) + 1
    except (OSError, IOError) as error:
        print(f"Error al procesar el archivo: {error}")
        return

    elapsed_time = time.time() - start_time
    # Guardar resultados en archivo
    output_file = "WordCountResults.txt"
    try:
        with open(output_file, 'w', encoding='utf-8') as out:
            for word, count in sorted(word_counts.items()):
                out.write(f"{word}: {count}\n")
            out.write(f"\nGran Total: {total_words}\n")
            out.write(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos\n")
    except (OSError, IOError) as error:
        print(f"Error al escribir en el archivo de salida: {error}")

    # Imprimir resultados en pantalla
    for word, count in sorted(word_counts.items()):
        print(f"{word}: {count}")
    print(f"\nGran Total: {total_words}")
    print(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos")

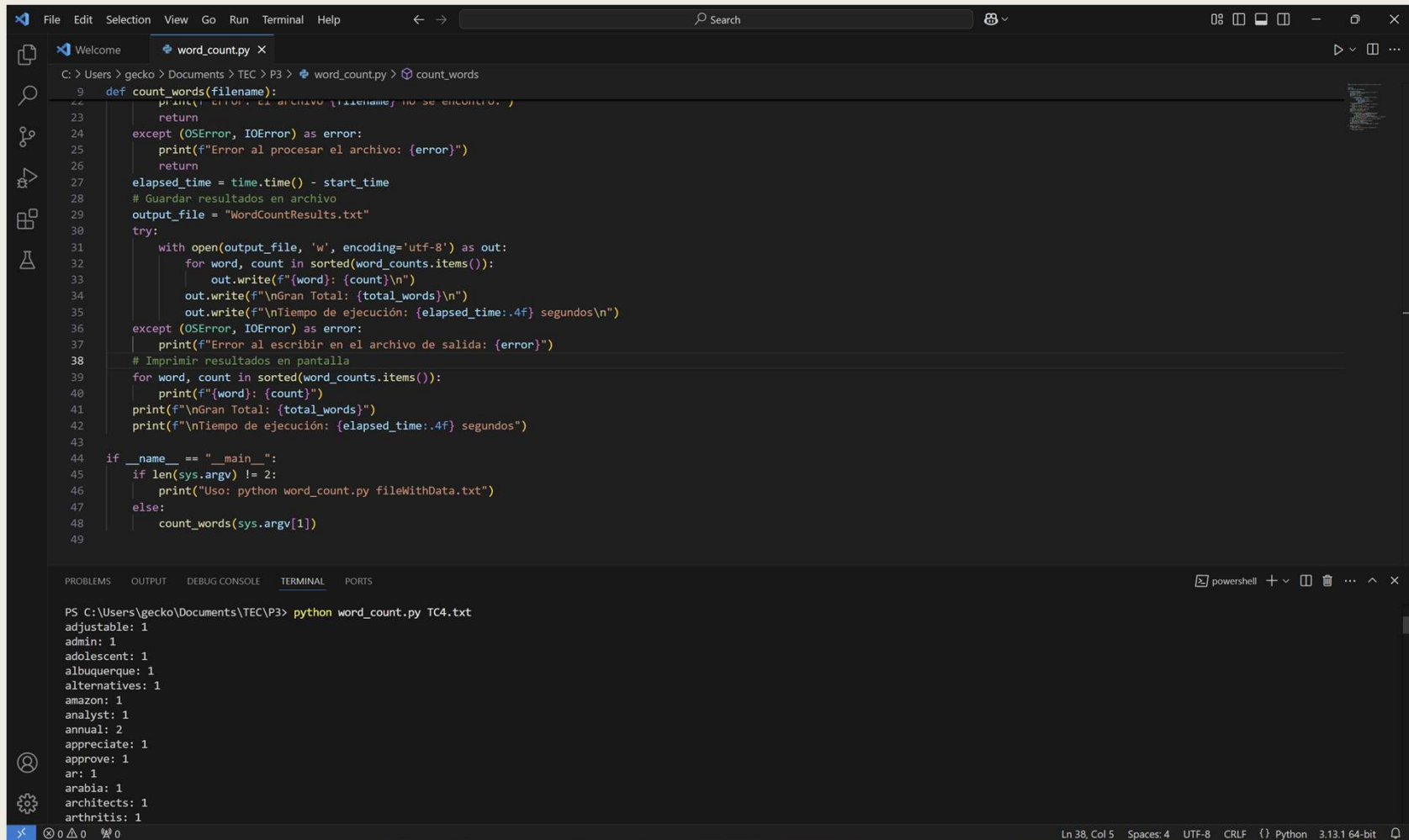
if __name__ == "__main__":
    if len(sys.argv) != 2:
        print("Uso: python word_count.py fileWithData.txt")
    else:
        count_words(sys.argv[1])
```

The terminal output shows the execution of the script with the file `TC3.txt`. The output lists the word counts for each word in the file:

```
PS C:\Users\gecko\Documents\TEC\P3> python word_count.py TC3.txt
acquisition: 1
advances: 1
affects: 1
aids: 1
allergy: 1
ambient: 1
an: 1
analyzed: 1
antiques: 1
apple: 1
archive: 1
archived: 1
argued: 1
aruba: 1
```

The status bar at the bottom indicates the current line and column (Ln 38, Col 5), the number of spaces (4), the encoding (UTF-8), the line ending (CRLF), the interpreter (Python), and the bitness (3.13.1 64-bit).

## Problema 3 corregido con Pylint: Ejecución ejercicio TC4



The image shows a Visual Studio Code editor window with a Python file named `word_count.py` open. The file is located at `C:\Users\gecko\Documents\TEC\P3\word_count.py`. The code defines a function `count_words(filename)` that reads a file, counts the words, and writes the results to `WordCountResults.txt`. It also prints the results to the console. The script is executed from the terminal using the command `python word_count.py TC4.txt`. The output shows the word counts for various words in the file `TC4.txt`.

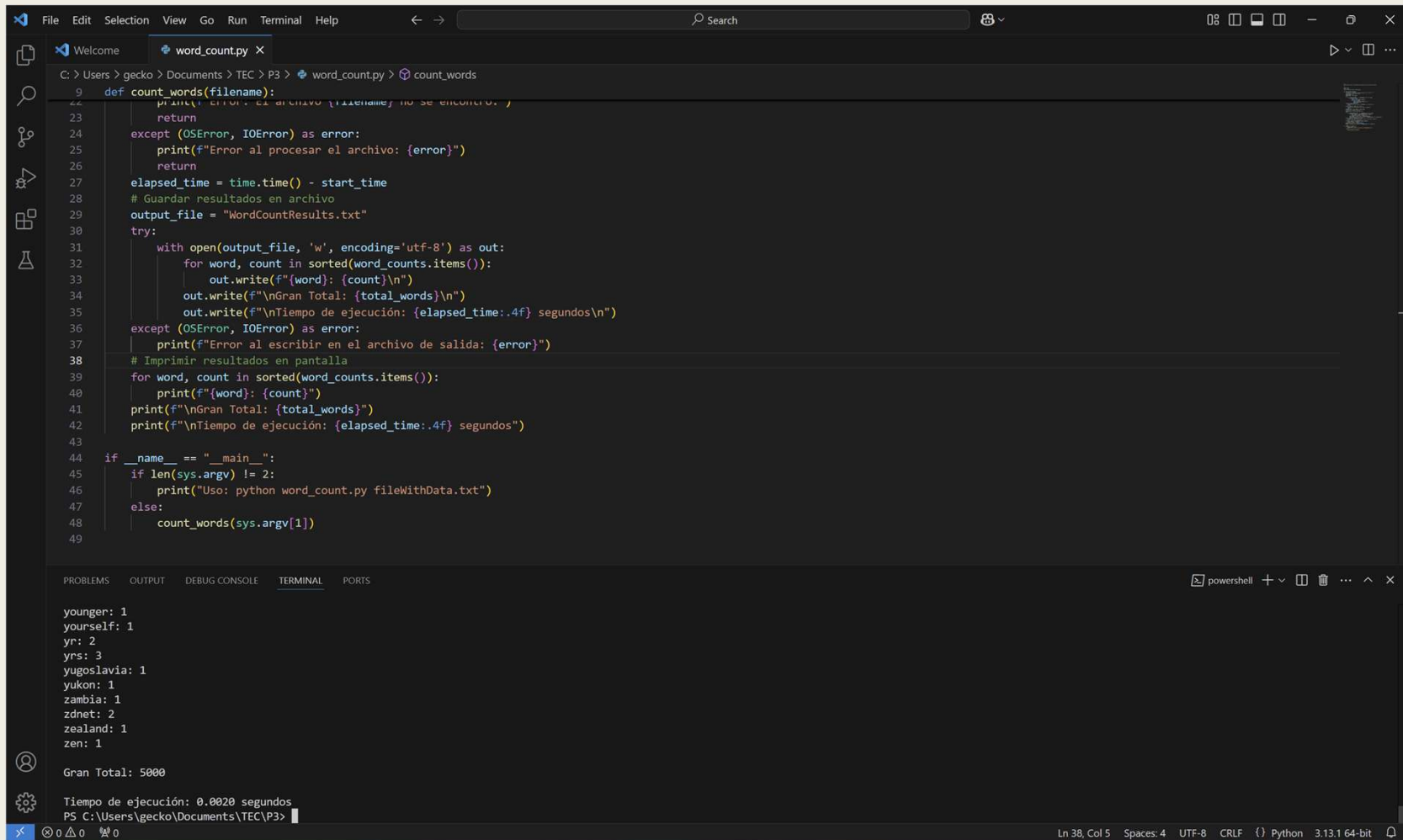
```
9 def count_words(filename):
10     try:
11         with open(filename, 'r', encoding='utf-8') as f:
12             word_counts = {}
13             for line in f:
14                 words = line.split()
15                 for word in words:
16                     word_counts[word] = word_counts.get(word, 0) + 1
17             return word_counts
18     except (OSError, IOError) as error:
19         print(f"Error al procesar el archivo: {error}")
20         return {}
21
22     elapsed_time = time.time() - start_time
23     # Guardar resultados en archivo
24     output_file = "WordCountResults.txt"
25     try:
26         with open(output_file, 'w', encoding='utf-8') as out:
27             for word, count in sorted(word_counts.items()):
28                 out.write(f"{word}: {count}\n")
29             out.write(f"\nGran Total: {total_words}\n")
30             out.write(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos\n")
31     except (OSError, IOError) as error:
32         print(f"Error al escribir en el archivo de salida: {error}")
33
34     # Imprimir resultados en pantalla
35     for word, count in sorted(word_counts.items()):
36         print(f"{word}: {count}")
37     print(f"\nGran Total: {total_words}")
38     print(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos")
39
40 if __name__ == "__main__":
41     if len(sys.argv) != 2:
42         print("Uso: python word_count.py fileWithData.txt")
43     else:
44         count_words(sys.argv[1])
45
46 47
48 49
```

Terminal Output:

```
PS C:\Users\gecko\Documents\TEC\P3> python word_count.py TC4.txt
adjustable: 1
admin: 1
adolescent: 1
albuquerque: 1
alternatives: 1
amazon: 1
analyst: 1
annual: 2
appreciate: 1
approve: 1
ar: 1
arabia: 1
architects: 1
arthritis: 1
```

Ln 38, Col 5 Spaces: 4 UTF-8 CRLF {} Python 3.13.1 64-bit

## Problema 3 corregido con Pylint: Ejecución ejercicio TC5



The image shows a Visual Studio Code editor window with a Python file named `word_count.py` open. The file is located at `C:\Users\gecko\Documents\TEC\P3\word_count.py`. The script defines a function `count_words(filename)` that processes a text file, counts the words, and writes the results to `WordCountResults.txt`. It also prints the total word count and execution time to the console. The script is executed from the command line, and the output is shown in the terminal window at the bottom.

```
9 def count_words(filename):
22     print(f"Error: El archivo {filename} no se encontró.")
23     return
24 except (OSError, IOError) as error:
25     print(f"Error al procesar el archivo: {error}")
26     return
27 elapsed_time = time.time() - start_time
28 # Guardar resultados en archivo
29 output_file = "WordCountResults.txt"
30 try:
31     with open(output_file, 'w', encoding='utf-8') as out:
32         for word, count in sorted(word_counts.items()):
33             out.write(f"{word}: {count}\n")
34         out.write(f"\nGran Total: {total_words}\n")
35         out.write(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos\n")
36 except (OSError, IOError) as error:
37     print(f"Error al escribir en el archivo de salida: {error}")
38 # Imprimir resultados en pantalla
39 for word, count in sorted(word_counts.items()):
40     print(f"{word}: {count}")
41 print(f"\nGran Total: {total_words}")
42 print(f"\nTiempo de ejecución: {elapsed_time:.4f} segundos")
43
44 if __name__ == "__main__":
45     if len(sys.argv) != 2:
46         print("Uso: python word_count.py fileWithData.txt")
47     else:
48         count_words(sys.argv[1])
49
```

The terminal output shows the execution results:

```
younger: 1
yourself: 1
yr: 2
yrs: 3
yugoslavia: 1
yukon: 1
zambia: 1
zdnet: 2
zealand: 1
zen: 1

Gran Total: 5000

Tiempo de ejecución: 0.0020 segundos
PS C:\Users\gecko\Documents\TEC\P3>
```

The status bar at the bottom indicates the current line is 38, column 5, with 4 spaces, UTF-8 encoding, CRLF line endings, Python 3.13.1 64-bit.