



# Tecnológico de Monterrey

Maestría en Inteligencia Artificial Aplicada - Programa de Posgrados en Línea

## **Actividad 4.2 Ejercicios de programación 1**

A01795690. Michelle Alejandro Durán Sánchez

Escuela de Ingeniería y Ciencias, Tecnológico de Monterrey

TC4017.10: Pruebas de software y aseguramiento de la calidad

Dr. Gerardo Padilla Zárate

Mtra. María Mylen Treviño E.

Ciudad de México, 02 de febrero de 2025

## Ejercicios de programación

### Ejercicio 1: Compute Statistics

*Requerimiento:*

**Requisito 1.** El programa deberá ser invocado desde una línea de comandos. El programa recibirá un archivo como parámetro. El archivo contendrá una lista de elementos (presumiblemente números).

**Requisito 2.** El programa deberá calcular todas las estadísticas descriptivas a partir de un archivo que contenga números. Los resultados deberán ser impresos en la pantalla y en un archivo llamado StatisticsResults.txt. Todos los cálculos deberán realizarse utilizando los algoritmos básicos, no funciones ni bibliotecas. Las estadísticas descriptivas son el conteo, la media, mediana, moda, desviación estándar y varianza muestral.

**Requisito 3.** El programa deberá incluir un mecanismo para manejar datos inválidos en el archivo. Los errores deben mostrarse en la consola y la ejecución deberá continuar.

**Requisito 4.** El nombre del programa será computeStatistics.py.

**Requisito 5.** El formato mínimo para invocar el programa será el siguiente:

- python computeStatistics.py fileWithData.txt

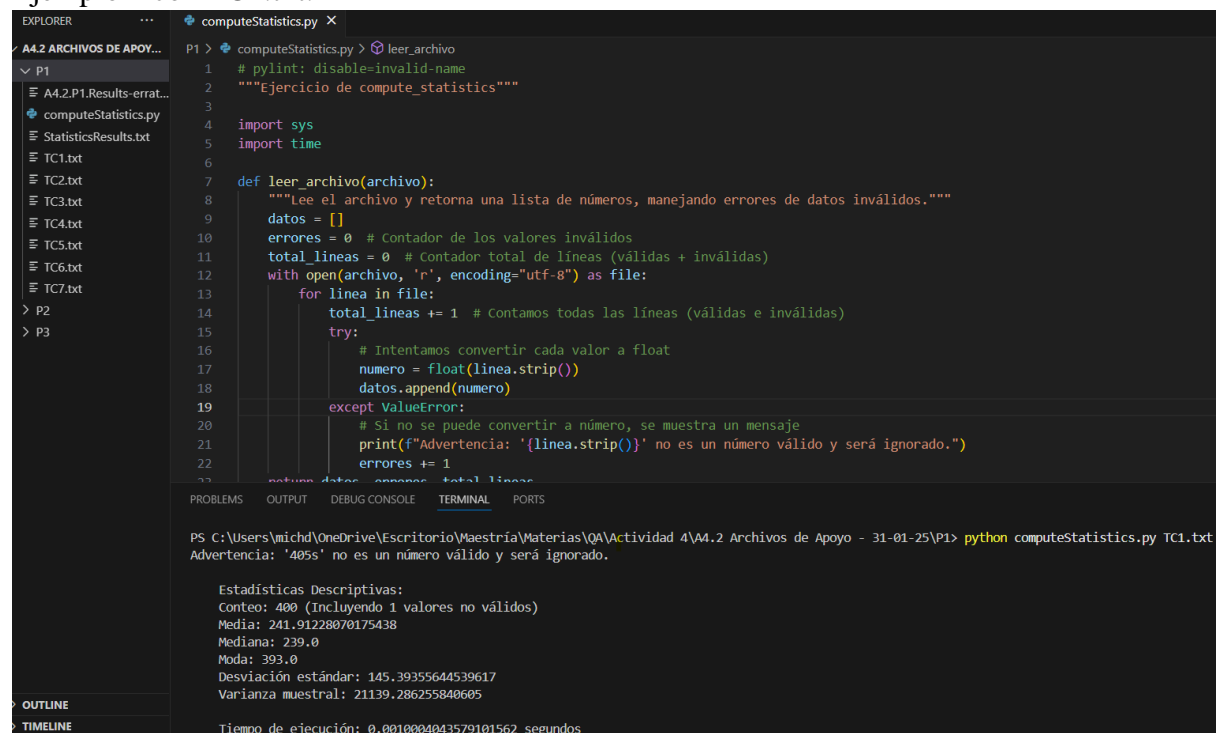
**Requisito 6.** El programa deberá gestionar archivos que contengan desde cientos hasta miles de elementos.

**Requisito 7.** El programa deberá incluir, al final de la ejecución, el tiempo transcurrido para la ejecución y el cálculo de los datos. Este número deberá ser incluido en el archivo de resultados y en la pantalla.

**Requisito 8.** Cumplir con PEP8.

*Resultados:*

Ejemplo 1 con TC1.txt:



```
EXPLORER
...
A4.2 ARCHIVOS DE APOYO...
  P1
    A4.2.P1.Results-errat...
    computeStatistics.py
    StatisticsResults.txt
    TC1.txt
    TC2.txt
    TC3.txt
    TC4.txt
    TC5.txt
    TC6.txt
    TC7.txt
  P2
  P3

computeStatistics.py
1  # pylint: disable=invalid-name
2  """Ejercicio de compute_statistics"""
3
4  import sys
5  import time
6
7  def leer_archivo(archivo):
8      """Lee el archivo y retorna una lista de números, manejando errores de datos inválidos."""
9      datos = []
10     errores = 0 # Contador de los valores inválidos
11     total_lineas = 0 # Contador total de líneas (válidas + inválidas)
12     with open(archivo, 'r', encoding="utf-8") as file:
13         for linea in file:
14             total_lineas += 1 # Contamos todas las líneas (válidas e inválidas)
15             try:
16                 # Intentamos convertir cada valor a float
17                 numero = float(linea.strip())
18                 datos.append(numero)
19             except ValueError:
20                 # Si no se puede convertir a número, se muestra un mensaje
21                 print(f"Advertencia: '{linea.strip()}' no es un número válido y será ignorado.")
22                 errores += 1
23     return datos, errores, total_lineas

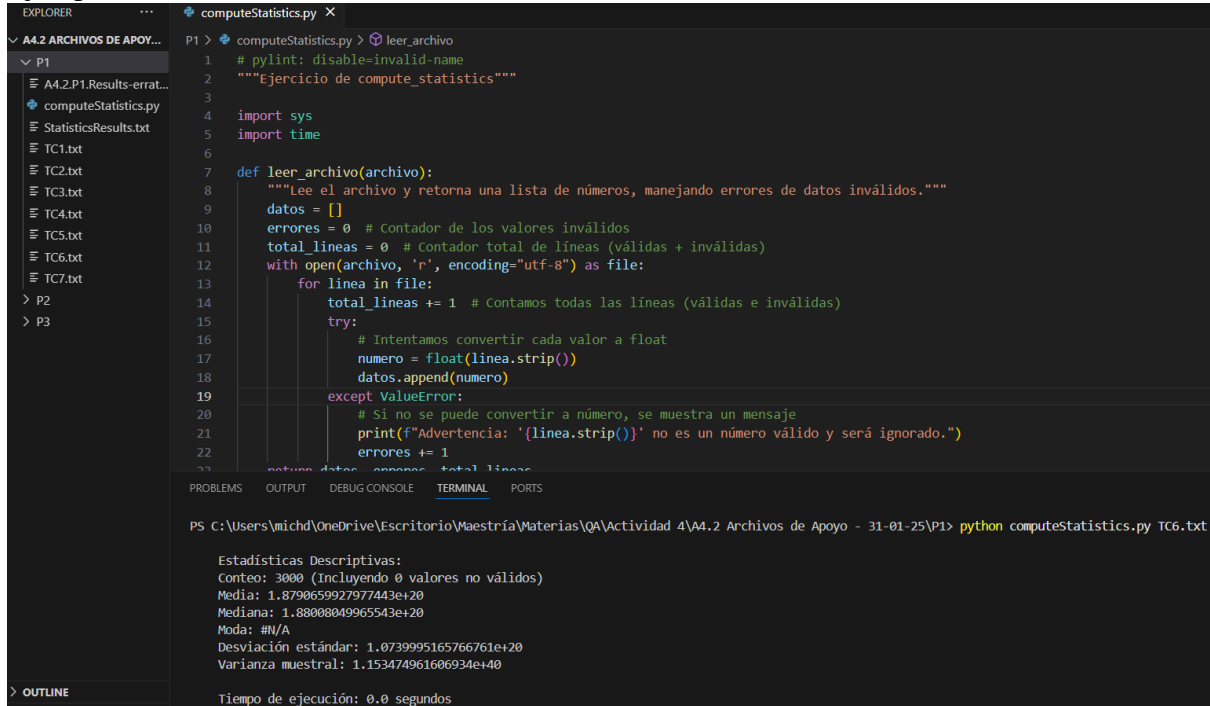
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\michd\OneDrive\Escritorio\Maestria\Materias\QA\Actividad 4\A4.2 Archivos de Apoyo - 31-01-25\P1> python computeStatistics.py TC1.txt
Advertencia: '405s' no es un número válido y será ignorado.

Estadísticas Descriptivas:
Conteo: 400 (Incluyendo 1 valores no válidos)
Media: 241.91228070175438
Mediana: 239.0
Moda: 393.0
Desviación estándar: 145.39355644539617
Varianza muestral: 21139.286255840605

OUTLINE
TIMELINE
Tiempo de ejecución: 0.0010004043579101562 segundos
```

## Ejemplo 2 con TC6.txt:



```
EXPLORER
...
A4.2 ARCHIVOS DE APOY...
  P1
    A4.2.P1.Results-errat...
    computeStatistics.py
    StatisticsResults.txt
    TC1.txt
    TC2.txt
    TC3.txt
    TC4.txt
    TC5.txt
    TC6.txt
    TC7.txt
  P2
  P3

P1 > computeStatistics.py > leer_archivo
1 # pylint: disable=invalid-name
2 """Ejercicio de compute_statistics"""
3
4 import sys
5 import time
6
7 def leer_archivo(archivo):
8     """Lee el archivo y retorna una lista de números, manejando errores de datos inválidos."""
9     datos = []
10    errores = 0 # Contador de los valores inválidos
11    total_lineas = 0 # Contador total de líneas (válidas + inválidas)
12    with open(archivo, 'r', encoding="utf-8") as file:
13        for linea in file:
14            total_lineas += 1 # Contamos todas las líneas (válidas e inválidas)
15            try:
16                # Intentamos convertir cada valor a float
17                numero = float(linea.strip())
18                datos.append(numero)
19            except ValueError:
20                # Si no se puede convertir a número, se muestra un mensaje
21                print(f"Advertencia: '{linea.strip()}' no es un número válido y será ignorado.")
22                errores += 1
23    return datos, errores, total_lineas

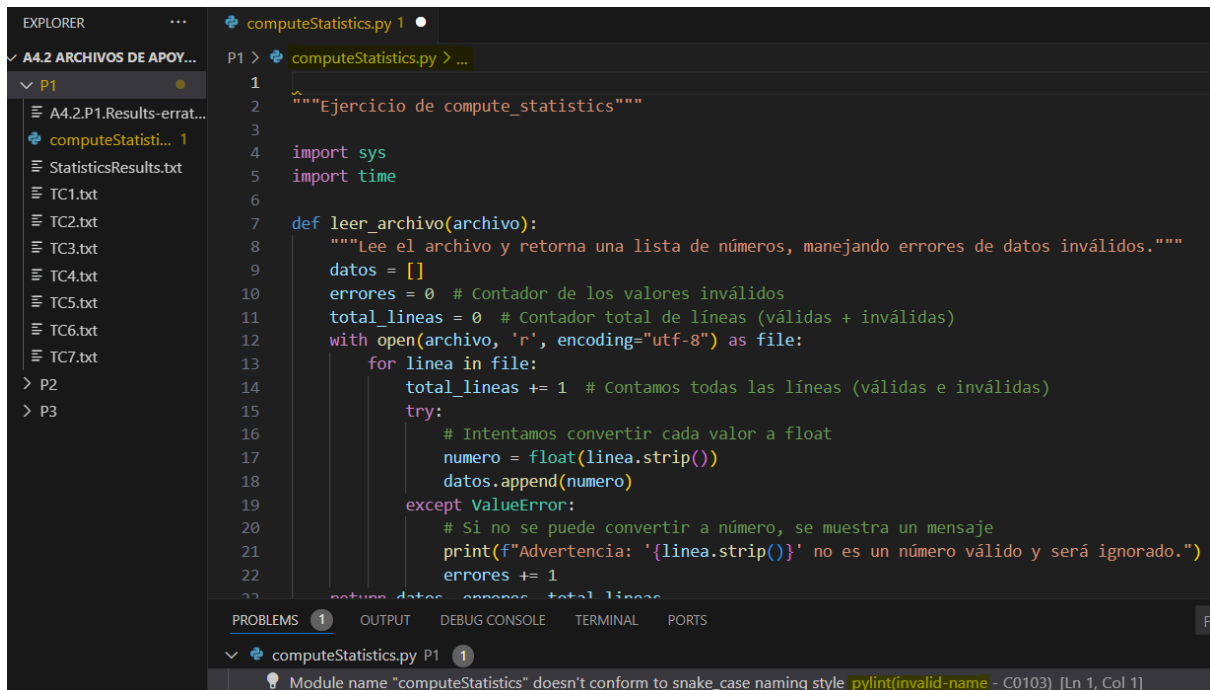
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\michd\OneDrive\Escritorio\Maestría\Materias\QA\Actividad 4\A4.2 Archivos de Apoyo - 31-01-25\P1> python computeStatistics.py TC6.txt

Estadísticas Descriptivas:
Conteo: 3000 (Incluyendo 0 valores no válidos)
Media: 1.8790659927977443e+20
Mediana: 1.88008049965543e+20
Moda: #N/A
Desviación estándar: 1.0739995165766761e+20
Varianza muestral: 1.153474961606934e+40

Tiempo de ejecución: 0.0 segundos
```

Ejemplo 1 de validación con Pylint. Se tuvo que deshabilitar el nombrado porque nos pedía renombrar el archivo como: “Compute\_Statistics.py” pero el requerimiento solicitaba guardarlo como “computeStatistics.py”:

- Sin deshabilitar el nombrado:

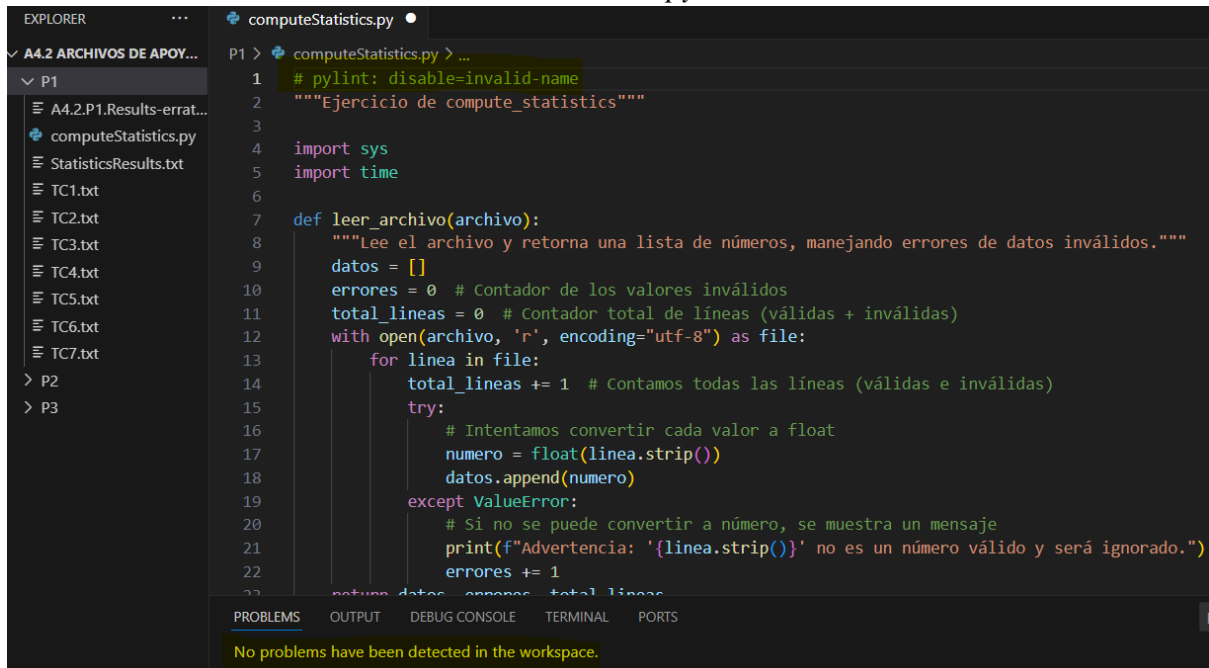


```
EXPLORER
...
A4.2 ARCHIVOS DE APOY...
  P1
    A4.2.P1.Results-errat...
    computeStatistics.py 1
    StatisticsResults.txt
    TC1.txt
    TC2.txt
    TC3.txt
    TC4.txt
    TC5.txt
    TC6.txt
    TC7.txt
  P2
  P3

P1 > computeStatistics.py > ...
1
2 """Ejercicio de compute_statistics"""
3
4 import sys
5 import time
6
7 def leer_archivo(archivo):
8     """Lee el archivo y retorna una lista de números, manejando errores de datos inválidos."""
9     datos = []
10    errores = 0 # Contador de los valores inválidos
11    total_lineas = 0 # Contador total de líneas (válidas + inválidas)
12    with open(archivo, 'r', encoding="utf-8") as file:
13        for linea in file:
14            total_lineas += 1 # Contamos todas las líneas (válidas e inválidas)
15            try:
16                # Intentamos convertir cada valor a float
17                numero = float(linea.strip())
18                datos.append(numero)
19            except ValueError:
20                # Si no se puede convertir a número, se muestra un mensaje
21                print(f"Advertencia: '{linea.strip()}' no es un número válido y será ignorado.")
22                errores += 1
23    return datos, errores, total_lineas

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Module name "computeStatistics" doesn't conform to snake_case naming style pylint(invalid-name - C0103) [Ln 1, Col 1]
```

- Deshabilitando el nombrado con la línea “`# pylint: disable=invalid-name`”



```

1  # pylint: disable=invalid-name
2  """Ejercicio de compute_statistics"""
3
4  import sys
5  import time
6
7  def leer_archivo(archivo):
8      """Lee el archivo y retorna una lista de números, manejando errores de datos inválidos."""
9      datos = []
10     errores = 0 # Contador de los valores inválidos
11     total_lineas = 0 # Contador total de líneas (válidas + inválidas)
12     with open(archivo, 'r', encoding="utf-8") as file:
13         for linea in file:
14             total_lineas += 1 # Contamos todas las líneas (válidas e inválidas)
15             try:
16                 # Intentamos convertir cada valor a float
17                 numero = float(linea.strip())
18                 datos.append(numero)
19             except ValueError:
20                 # Si no se puede convertir a número, se muestra un mensaje
21                 print(f"Advertencia: '{linea.strip()}' no es un número válido y será ignorado.")
22                 errores += 1
23     return datos, errores, total_lineas

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

No problems have been detected in the workspace.

## Ejercicio 2: Converter

*Requerimiento:*

**Requisito 1.** El programa debe ser invocado desde una línea de comandos. El programa debe recibir un archivo como parámetro. El archivo contendrá una lista de elementos (presumiblemente números).

**Requisito 2.** El programa debe convertir los números a bases binaria y hexadecimal. Los resultados deben ser impresos en la pantalla y en un archivo llamado ConversionResults.txt. Todos los cálculos deben ser realizados utilizando algoritmos básicos, no funciones ni bibliotecas.

**Requisito 3.** El programa debe incluir un mecanismo para manejar datos inválidos en el archivo. Los errores deben ser mostrados en la consola y la ejecución debe continuar.

**Requisito 4.** El nombre del programa debe ser convertNumbers.py.

**Requisito 5.** El formato mínimo para invocar el programa debe ser el siguiente:

- python convertNumbers.py fileWithData.txt

**Requisito 6.** El programa debe gestionar archivos que contengan desde cientos hasta miles de elementos.

**Requisito 7.** El programa debe incluir al final de la ejecución el tiempo transcurrido para la ejecución y el cálculo de los datos. Este número debe ser incluido en el archivo de resultados y en la pantalla.

**Requisito 8.** Cumplir con PEP8.

*Resultados:*

Ejemplo 1 con TC1.txt:

**Nota.** Para este primer documento de texto, modifiqué el TC1.txt de acuerdo a lo que se esperaba porque tenía 1 columna de más, por lo que sólo dejé la columna que inicia con “82” y la columna que inicia con “6980368” la puse en un TC5.txt.

The screenshot shows a VS Code interface with the following components:

- EXPLORER:** A4.2 ARCHIVOS DE APOYO...
  - > P1
  - ✓ P2
    - A4.2.P2.Results.txt
    - ConversionResults.txt
    - convertNumbers.py
    - TC1.txt
    - TC2.txt
    - TC3.txt
    - TC4.txt
    - TC5.txt
  - > P3
- FILE LIST:**
  - 1 ITEM TC1 BIN HEX
  - 2 1 6980368 82 1010010 52
  - 3 2 5517055 100 1100100 64
  - 4 3 1336159 93 1011101 50
  - 5 4 6750185 65 1000001 41
  - 6 5 1771937 58 111010 3A
  - 7 6 360952 51 110011 33
  - 8 7 5672561 67 1000011 43
  - 9 8 916583 75 1001011 4B
  - 10 9 2700138 10 1010 A
  - 11 10 9645053 93 1011101 50
  - 12 11 1181110 29 11101 10
  - 13 12 1492185 86 1010110 56
  - 14 13 4018595 100 1100100 64
  - 15 14 7654888 21 10101 15
  - 16 15 7062453 40 101000 28
  - 17 16 2478010 90 1011010 5A
  - 18 17 6134768 1 1 1
  - 19 18 8420417 100 1100100 64
  - 20 19 2017489 0 0 0
- TERMINAL:**

```
PS C:\Users\michd\OneDrive\Escritorio\Maestría\Materias\QA\Actividad 4\A4.2 Archivos de Apoyo - 31-01-25\P2> python convertNumbers.py TC1.txt
82 en binario: 0b1010010 en hexadecimal: 0x52
100 en binario: 0b1100100 en hexadecimal: 0x64
93 en binario: 0b1011101 en hexadecimal: 0x5D
65 en binario: 0b1000001 en hexadecimal: 0x41
58 en binario: 0b111010 en hexadecimal: 0x3A
51 en binario: 0b110011 en hexadecimal: 0x33
67 en binario: 0b1000011 en hexadecimal: 0x43
75 en binario: 0b1001011 en hexadecimal: 0x4B
10 en binario: 0b1010 en hexadecimal: 0xA
93 en binario: 0b1011101 en hexadecimal: 0x5D
29 en binario: 0b11101 en hexadecimal: 0x1D
86 en binario: 0b1010110 en hexadecimal: 0x56
100 en binario: 0b1100100 en hexadecimal: 0x64
21 en binario: 0b10101 en hexadecimal: 0x15
```

Ejemplo 2 con TC3.txt:

Estuvo interesante el reto con números negativos. La solución la obtuve de un blog que coloqué en las referencias bibliográficas.

The screenshot shows a VS Code interface with the following components:

- EXPLORER:** A4.2 ARCHIVOS DE APOYO...
  - > P1
  - ✓ P2
    - A4.2.P2.Results.txt
    - ConversionResults.txt
    - convertNumbers.py
    - TC1.txt
    - TC2.txt
    - TC3.txt
    - TC4.txt
    - TC5.txt
  - > P3
- FILE LIST:**
  - 1 ITEM TC1 BIN HEX
  - 2 1 6980368 82 1010010 52
  - 3 2 5517055 100 1100100 64
  - 4 3 1336159 93 1011101 50
  - 5 4 6750185 65 1000001 41
  - 6 5 1771937 58 111010 3A
  - 7 6 360952 51 110011 33
  - 8 7 5672561 67 1000011 43
  - 9 8 916583 75 1001011 4B
  - 10 9 2700138 10 1010 A
  - 11 10 9645053 93 1011101 50
  - 12 11 1181110 29 11101 10
  - 13 12 1492185 86 1010110 56
  - 14 13 4018595 100 1100100 64
  - 15 14 7654888 21 10101 15
  - 16 15 7062453 40 101000 28
  - 17 16 2478010 90 1011010 5A
  - 18 17 6134768 1 1 1
  - 19 18 8420417 100 1100100 64
  - 20 19 2017489 0 0 0
- TERMINAL:**

```
PS C:\Users\michd\OneDrive\Escritorio\Maestría\Materias\QA\Actividad 4\A4.2 Archivos de Apoyo - 31-01-25\P2> python convertNumbers.py TC3.txt
-10 en binario: 0b1111111111111111111111111111110110 en hexadecimal: 0xFFFFFFF6
47 en binario: 0b101111 en hexadecimal: 0x2F
47 en binario: 0b101111 en hexadecimal: 0x2F
-13 en binario: 0b1111111111111111111111111111110011 en hexadecimal: 0xFFFFFFF3
-32 en binario: 0b111111111111111111111111111111100000 en hexadecimal: 0xFFFFFFF0
1 en binario: 0b1 en hexadecimal: 0x1
1 en binario: 0b1 en hexadecimal: 0x1
-25 en binario: 0b111111111111111111111111111111100111 en hexadecimal: 0xFFFFFE7
-33 en binario: 0b111111111111111111111111111111011111 in hexadecimal: 0xFFFFFDF
16 en binario: 0b10000 en hexadecimal: 0x10
17 en binario: 0b10001 en hexadecimal: 0x11
4 en binario: 0b100 en hexadecimal: 0x4
```

Ejemplo 3 con TC4.txt:

Aquí el reto fue el despliegue de valores no válidos.

[illegible]

Muestra de validación con Pylint sin errores en la imagen 1. Si los hubiera se vería como en la imagen 2.

Imagen 1. Sin errores

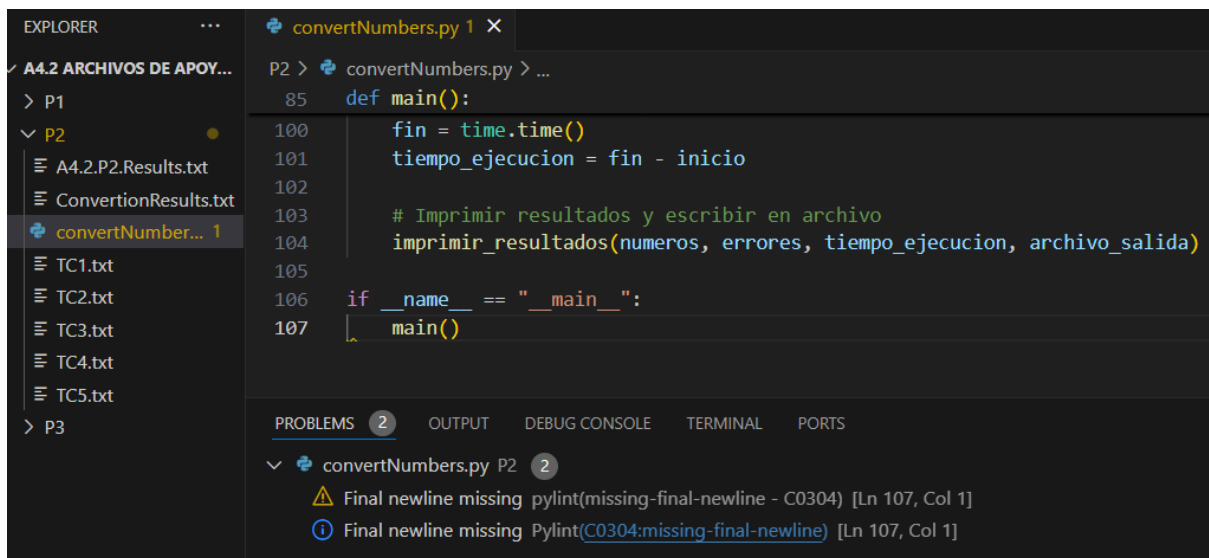


Imagen 2. Con error de que falta la última línea

### Ejercicio 3: Count words

*Requerimiento:*

**Requisito 1:** El programa debe ser invocado desde una línea de comandos. El programa debe recibir un archivo como parámetro. El archivo contendrá palabras (presumiblemente entre espacios).

**Requisito 2:** El programa debe identificar todas las palabras distintas y su frecuencia (cuántas veces aparece la palabra "X" en el archivo). Los resultados deben ser impresos en la pantalla y en un archivo llamado WordCountResults.txt.

Toda la computación debe ser realizada utilizando los algoritmos básicos, no funciones ni bibliotecas.

**Requisito 3:** El programa debe incluir un mecanismo para manejar datos inválidos en el archivo. Los errores deben mostrarse en la consola y la ejecución debe continuar.

**Requisito 4:** El nombre del programa debe ser wordCount.py.

**Requisito 5:** El formato mínimo para invocar el programa debe ser el siguiente:

- python wordCount.py fileWithData.txt

**Requisito 6:** El programa debe gestionar archivos que contengan desde cientos de elementos hasta miles de elementos.

**Requisito 7:** El programa debe incluir al final de la ejecución el tiempo transcurrido durante la ejecución y el cálculo de los datos. Este número debe ser incluido tanto en el archivo de resultados como en la pantalla.

**Requisito 8:** Debe cumplir con las directrices de estilo PEP8.

*Resultados:*

Ejemplo 1 con TC1.txt:

No hubo tanto problema con el contador de palabras. Me parece que hizo falta colocar algunos espacios entre las palabras, así como intercalar minúsculas con mayúsculas e insertar números entre las palabras. Sin embargo, el problema resultó bien y colocó un contador al final de la cantidad de palabras leídas y cantidad de palabras diferentes.

EXPLORER

...

A4.2 ARCHIVOS DE APOY...

> P1

> P2

▼ P3

TC1.Results.txt

TC1.txt

TC2.Results.txt

TC2.txt

TC3.Results.txt

TC3.txt

TC4.Results.txt

TC4.txt

TC5.Results.txt

TC5.txt

wordCount.py

WordCountResults.txt

wordCount.py

TC1.Results.txt

P3 > TC1.Results.txt > data

1 Row Labels Count of TC1

2 conservative 2

3 oakland 1

4 mon 1

5 achievement 1

6 adequate 1

7 male 1

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

PS C:\Users\michd\OneDrive\Escritorio\Maestria\Materias\QA\Actividad 4\A4.2 Archivos de Apoyo - 31-01-25\P3> python wordCount.py TC1.txt

Palabra - Frecuencia

conservative - 2

mother - 1

tions - 1

coastal - 1

Total de palabras leídas: 100

Total de palabras diferentes: 99

Tiempo de ejecución: 0.0 segundos

PS C:\Users\michd\OneDrive\Escritorio\Maestria\Materias\QA\Actividad 4\A4.2 Archivos de Apoyo - 31-01-25\P3>

## Ejemplo 2 con TC2.txt:

EXPLORER

...

A4.2 ARCHIVOS DE APOY...

> P1

> P2

▼ P3

TC1.Results.txt

TC1.txt

TC2.Results.txt

TC2.txt

TC3.Results.txt

TC3.txt

TC4.Results.txt

TC4.txt

TC5.Results.txt

TC5.txt

wordCount.py

WordCountResults.txt

wordCount.py

TC2.Results.txt

P3 > TC2.Results.txt > data

1 Row Labels Count of TC2

8 lease 4

9 chain 4

10 revenues 4

11 doc 4

12 filme 4

13 inflation 4

14 wood 3

15 pre 3

16 literacy 1

17 returned 1

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

PS C:\Users\michd\OneDrive\Escritorio\Maestria\Materias\QA\Actividad 4\A4.2 Archivos de Apoyo - 31-01-25\P3> python wordCount.py TC2.txt

Palabra - Frecuencia

lease - 4

brass - 4

revenues - 4

targeted - 4

inflation - 4

chain - 4

holders - 4

amongst - 4

monaco - 4

filme - 4

doc - 4

kingston - 4

wood - 3

pre - 3

conduct - 1

kuwait - 1

EXPLORER

...

A4.2 ARCHIVOS DE APOY...

> P1

> P2

▼ P3

TC1.Results.txt

TC1.txt

TC2.Results.txt

TC2.txt

TC3.Results.txt

TC3.txt

TC4.Results.txt

TC4.txt

TC5.Results.txt

TC5.txt

wordCount.py

WordCountResults.txt

wordCount.py

TC2.Results.txt

P3 > TC2.Results.txt > data

1 Row Labels Count of TC2

141 italy 1

142 win 1

143 j 1

144 jeff 1

145 jon 1

146 Grand Total 184

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

ion - 1

interface - 1

remained - 1

icons - 1

excessive - 1

jon - 1

Total de palabras leídas: 184

Total de palabras diferentes: 144

Tiempo de ejecución: 0.0 segundos



Muestra de validación con Pylint sin errores en la imagen 1. Si los hubiera se vería como en la imagen 2.

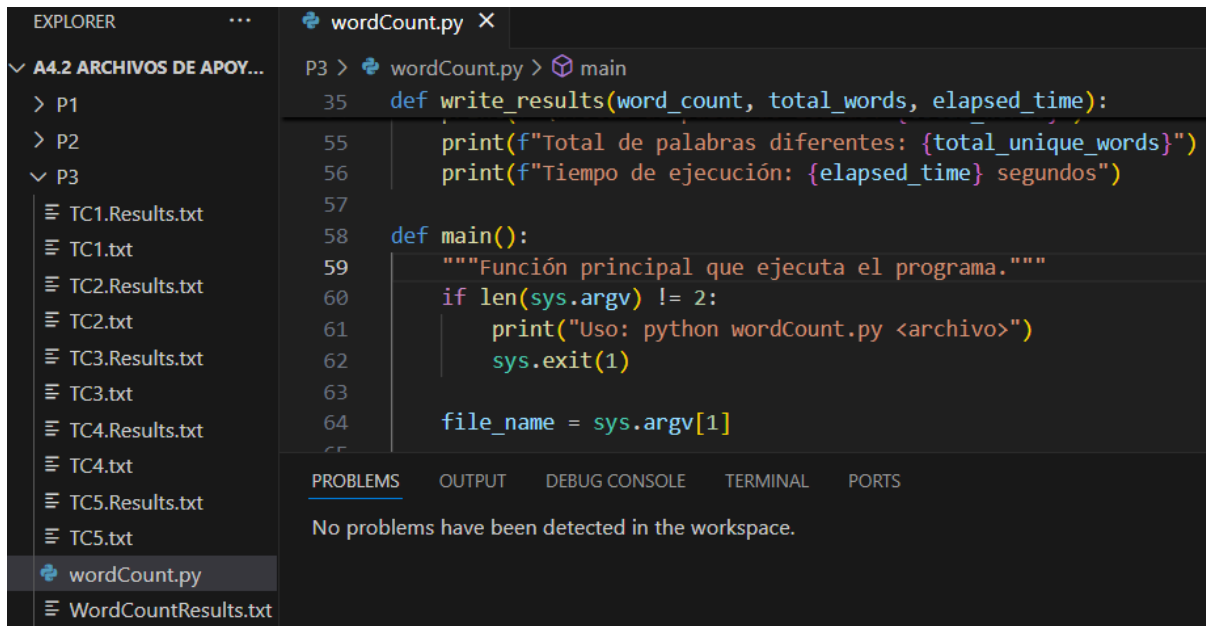


Imagen 1. Sin errores

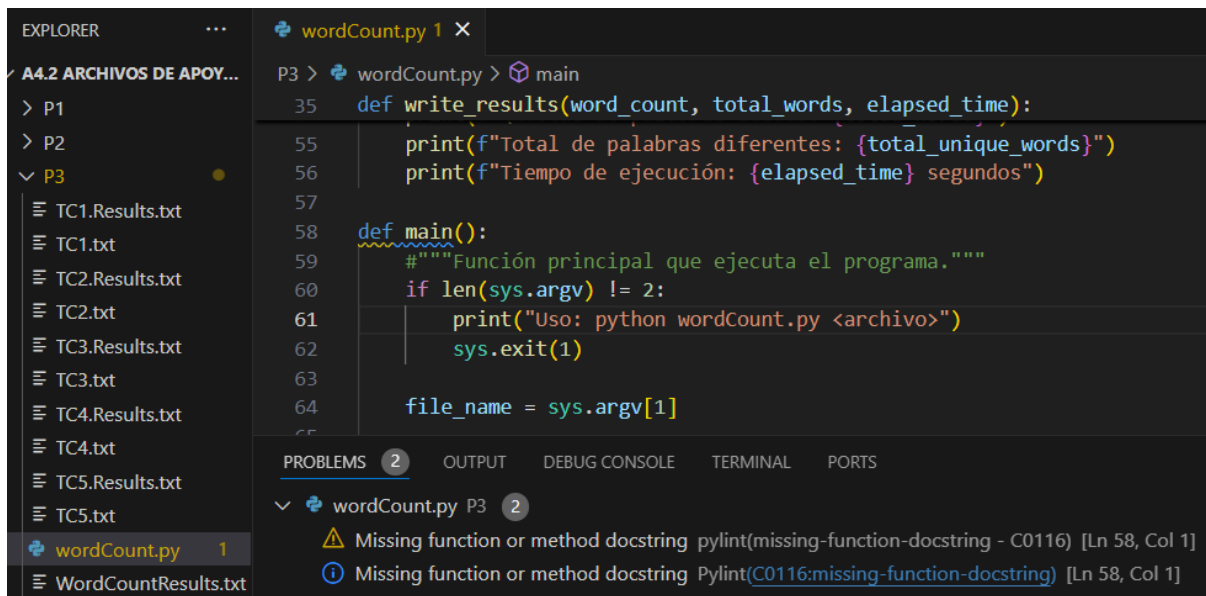


Imagen 2. Con error de que falta la documentación del método main()

**Liga de GitHub para ver ejercicios:**

- [https://github.com/MichelleDuranTEC/A01795690\\_A4.2.git](https://github.com/MichelleDuranTEC/A01795690_A4.2.git)

## Referencias Bibliográficas

HVSW. (2018, 6 de noviembre). *Números negativos*. Disponible en: <https://programacion-retro-c64.blog/2018/11/06/numeros-negativos/>