



**Tecnológico
de Monterrey**

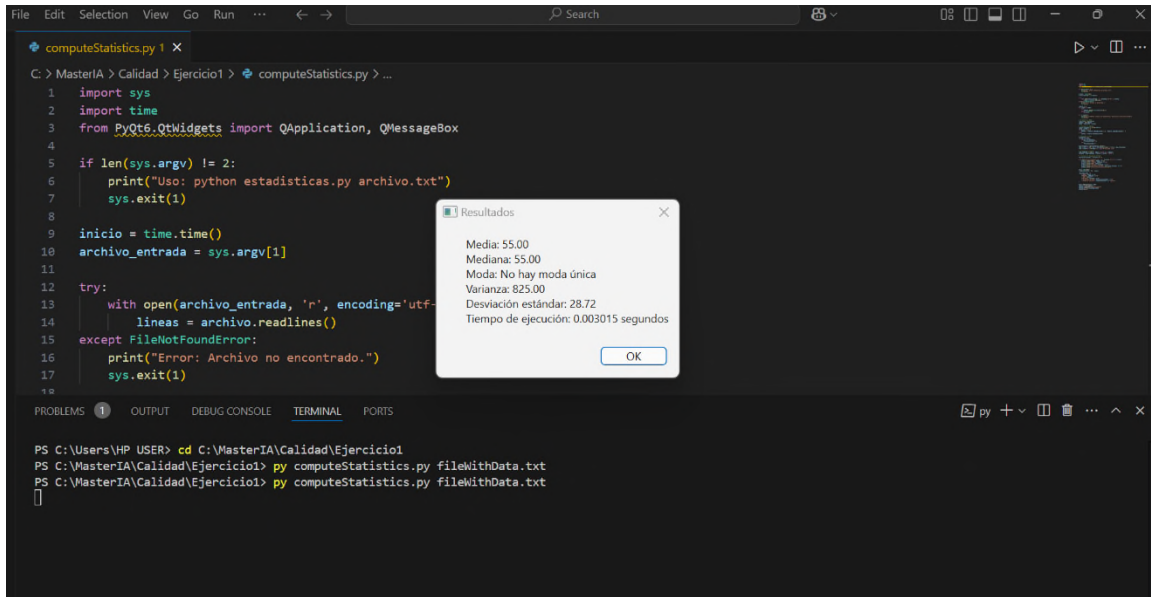
4.2 Ejercicio de programación 1

A01794822 - Juan Gallardo Ambriz

Pruebas de software y aseguramiento de la calidad

2 de febrero del 2025

1. Evidencia: computeStatistics.py



The screenshot shows a code editor with the following Python script:

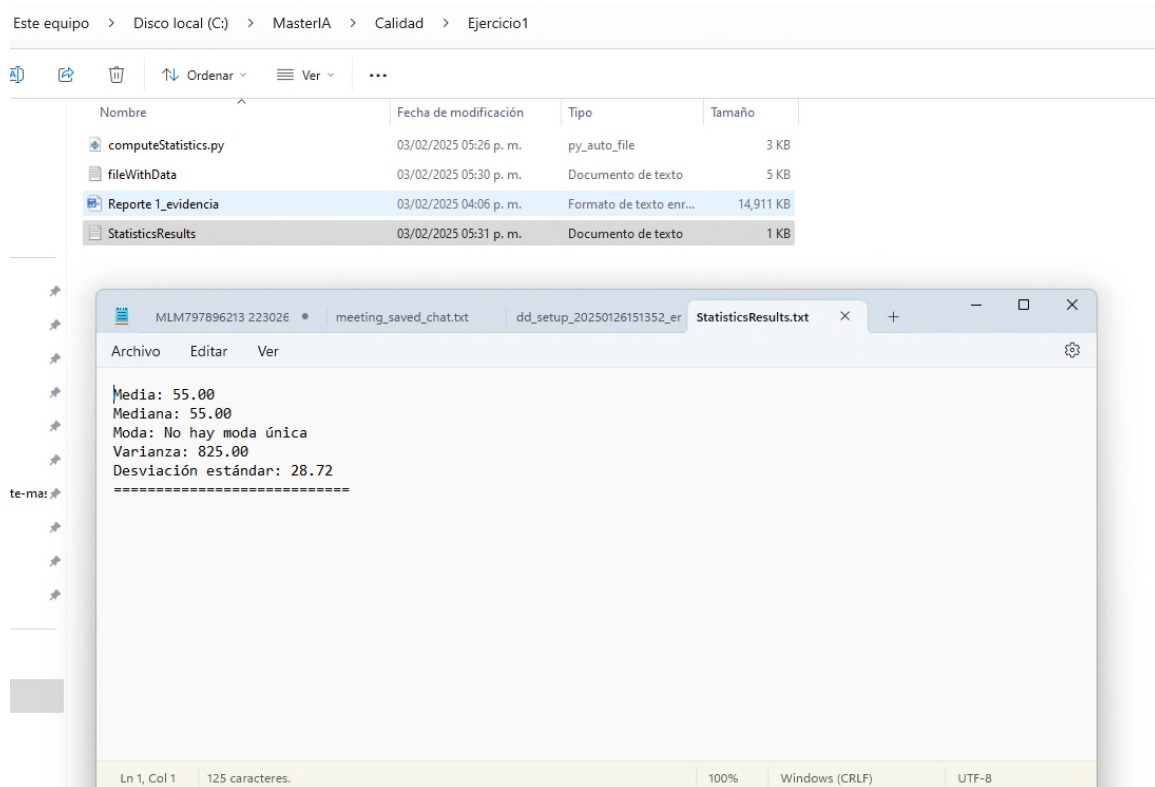
```
1 import sys
2 import time
3 from PyQt6.QtWidgets import QApplication, QMessageBox
4
5 if len(sys.argv) != 2:
6     print("Uso: python estadísticas.py archivo.txt")
7     sys.exit(1)
8
9 inicio = time.time()
10 archivo_entrada = sys.argv[1]
11
12 try:
13     with open(archivo_entrada, 'r', encoding='utf-8') as f:
14         lineas = f.readlines()
15 except FileNotFoundError:
16     print("Error: Archivo no encontrado.")
17     sys.exit(1)
```

A dialog box titled "Resultados" displays the output of the script:

```
Media: 55.00
Mediana: 55.00
Moda: No hay moda única
Varianza: 825.00
Desviación estándar: 28.72
Tiempo de ejecución: 0.003015 segundos
```

The terminal at the bottom shows the command execution:

```
PS C:\Users\HP USER> cd C:\MasterIA\Calidad\Ejercicio1
PS C:\MasterIA\Calidad\Ejercicio1> py computeStatistics.py fileWithData.txt
PS C:\MasterIA\Calidad\Ejercicio1> py computeStatistics.py fileWithData.txt
```



2. Evidencia: convertNumbers.py

The screenshot shows a code editor with the following Python script:

```
def process_file(file_path):
    elapsed_time = end_time - start_time
    results.append(f"Tiempo de ejecución: {elapsed_time}")

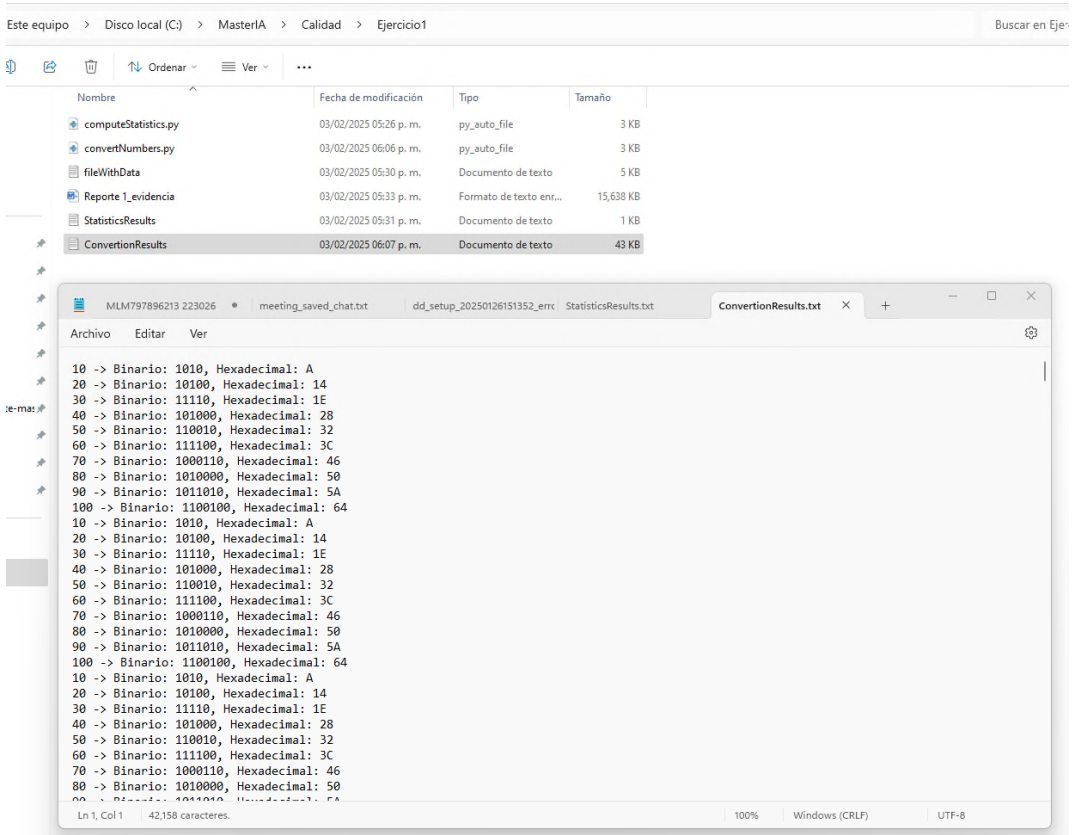
    # Guardar resultados en un archivo
    with open(output_file, "w", encoding="utf-8"):
        for res in results:
            file.write(res + "\n")

    for res in results:
        print(res)

# Mostrar resultados en un cuadro de diálogo
app = QApplication(sys.argv)
msg = QMessageBox()
msg.setWindowTitle("Resultados de Conversión")
msg.setText("\n".join(results))
msg.setIcon(QMessageBox.Icon.Information)
```

The terminal output shows the results of the script execution:

```
30 -> Binario: 11110, Hexadecimal: 1E
40 -> Binario: 101000, Hexadecimal: 28
50 -> Binario: 110010, Hexadecimal: 32
60 -> Binario: 111100, Hexadecimal: 3C
70 -> Binario: 1000110, Hexadecimal: 46
80 -> Binario: 1010000, Hexadecimal: 50
90 -> Binario: 1011010, Hexadecimal: 5A
100 -> Binario: 1100100, Hexadecimal: 64
Tiempo de ejecución: 0.00190 segundos
```



3. Evidencia: wordCount.py

The screenshot shows a Python IDE with a dark theme. The main editor displays the `wordCount.py` script. The script imports `sys`, `time`, and `re`, and uses `PyQt6.QtWidgets` for GUI elements. It defines two functions: `read_file(filename)` which reads a file and returns its lines, and `process_text(lines)` which processes the lines and counts the words. The script is executed, and the output is shown in the terminal and a separate window titled "Resultados del Conteo de Palabras".

```
1 import sys
2 import time
3 import re
4 from PyQt6.QtWidgets import QApplication, QMessageBox
5
6 def read_file(filename):
7     """Lee el contenido de un archivo y devuelve su contenido en una lista de líneas.
8     """
9     try:
10         with open(filename, 'r', encoding='utf-8') as file:
11             return file.readlines()
12     except Exception as e:
13         print(f"Error al leer el archivo {filename}: {e}")
14         return []
15
16 def process_text(lines):
17     """Procesa las líneas de texto y cuenta la ocurrencia de cada palabra.
18     """
19     word_count = {}
20     for line in lines:
21         words = re.findall(r'\w+', line.lower())
22         for word in words:
23             word_count[word] = word_count.get(word, 0) + 1
24
25     return word_count
```

Terminal output:

```
puerta: 3
reloj: 3
rio: 4
silla: 3
sol: 3
telefono: 3
ventana: 3
zapato: 3
Tiempo de ejecución: 0.0005 segundos
```

Resultados del Conteo de Palabras:

```
agua: 3
arena: 4
bicicleta: 4
camino: 3
carro: 2
casa: 3
cielo: 3
computadora: 3
cuaderno: 2
dinero: 3
espejo: 3
estrella: 3
fuego: 3
gato: 3
lápiz: 2
libro: 2
llave: 2
luna: 3
mar: 3
mesa: 3
mochila: 5
montaña: 4
moto: 2
nube: 3
papel: 6
perro: 3
playa: 4
puerta: 3
reloj: 3
rio: 4
silla: 3
sol: 3
telefono: 3
ventana: 3
zapato: 3
Tiempo de ejecución: 0.0005 segundos
```

