


Calidad en el Código Fuente y Estándares de Codificación	 Tecnológico de Monterrey
Gerardo Solis Hernandez 952702	Monterrey N.L. a 30 Enero 2026

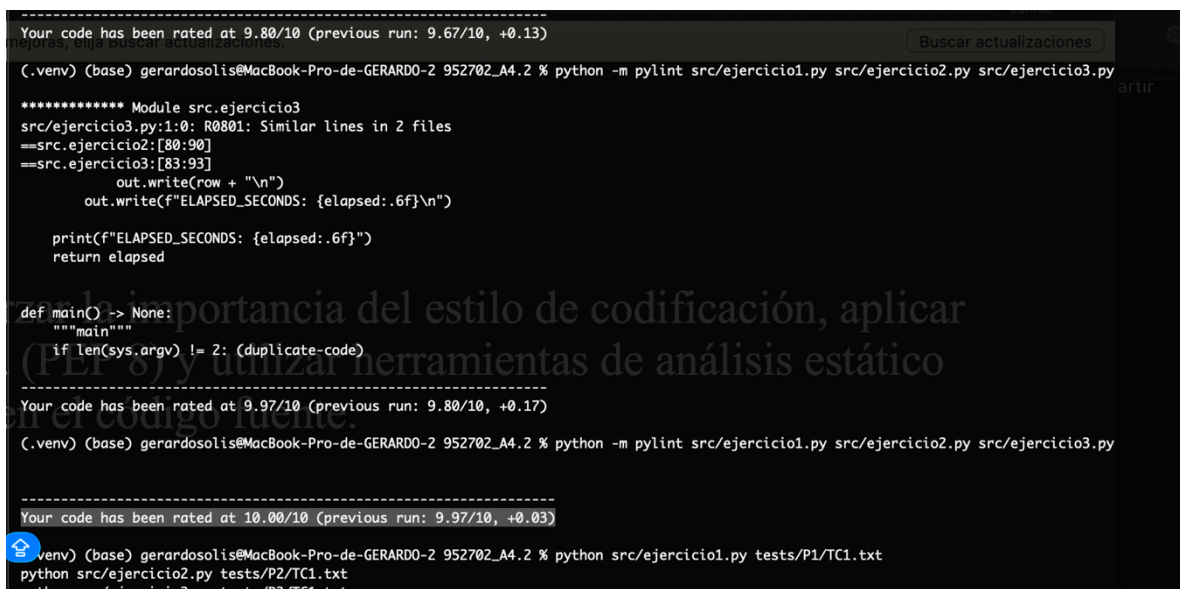
Repositorio: 952702_A4.2

Lenguaje: Python

Estándar aplicado: PEP 8

1. Objetivo de la actividad

El objetivo de esta actividad fue reforzar la importancia del estilo de codificación, aplicar un estándar reconocido en la industria (PEP 8) y utilizar herramientas de análisis estático para identificar y corregir problemas en el código fuente.



```

Your code has been rated at 9.80/10 (previous run: 9.67/10, +0.13)

(.venv) (base) gerardosolis@MacBook-Pro-de-GERARDO-2 952702_A4.2 % python -m pylint src/ejercicio1.py src/ejercicio2.py src/ejercicio3.py

***** Module src.ejercicio3
src/ejercicio3.py:1:0: R0801: Similar lines in 2 files
==src.ejercicio2:[80:90]
==src.ejercicio3:[83:93]
    out.write(row + "\n")
    out.write(f"ELAPSED_SECONDS: {elapsed:.6f}\n")

    print(f"ELAPSED_SECONDS: {elapsed:.6f}")
    return elapsed

def main() -> None:
    """main"""
    if len(sys.argv) != 2: (duplicate-code)

Your code has been rated at 9.97/10 (previous run: 9.80/10, +0.17)

(.venv) (base) gerardosolis@MacBook-Pro-de-GERARDO-2 952702_A4.2 % python -m pylint src/ejercicio1.py src/ejercicio2.py src/ejercicio3.py

Your code has been rated at 10.00/10 (previous run: 9.97/10, +0.03)

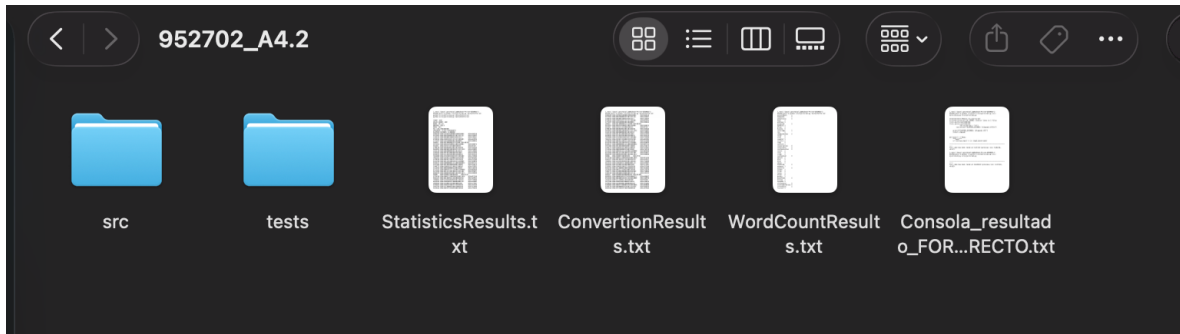
(.venv) (base) gerardosolis@MacBook-Pro-de-GERARDO-2 952702_A4.2 % python src/ejercicio1.py tests/P1/TC1.txt
python src/ejercicio2.py tests/P2/TC1.txt
python src/ejercicio3.py tests/P3/TC1.txt

```

2. Estructura del proyecto

Se creó un repositorio en GitHub con la siguiente estructura:

- src/: Contiene los programas ejercicio1.py, ejercicio2.py y ejercicio3.py
- tests/: Contiene las carpetas P1, P2 y P3 con los casos de prueba (TC*.txt)



3. Implementación de los programas

Los tres ejercicios fueron implementados en Python siguiendo el estándar PEP 8. Se cuidó el uso correcto de nombres de variables, funciones, sangrías, comentarios y docstrings para mejorar la legibilidad y mantenibilidad del código.

4. Pruebas de ejecución

Cada programa fue probado utilizando los archivos de entrada proporcionados en la carpeta tests.

La ejecución se realizó desde la terminal pasando el archivo de prueba como argumento al programa.

```
(base) gerardosolis@MacBook-Pro-de-GERARDO-2 952702_A4.2 % python src/ejercicio1.py tests/P1/TC1.txt
python src/ejercicio2.py tests/P2/TC1.txt
python src/ejercicio3.py tests/P3/TC1.txt

COUNT: 400
VALID_COUNT: 400
MEAN: 242.32
MEDIAN: 239.5
MODE: 393
SD: 145.2581068306
VARIANCE: 21152.7995989975
ELAPSED_SECONDS: 0.000830

5980368 BIN=11010101000001100010000    HEX=6A8310
5517055 BIN=10101000010111011111111    HEX=542EFF
1336159 BIN=101000110001101011111      HEX=14635F
6750185 BIN=1100110111111111101001     HEX=66FFE9
1771937 BIN=110110000100110100001      HEX=1B09A1
360952  BIN=1011000000111111000        HEX=581F8
5672561 BIN=10101101000111001110001    HEX=568E71
916583  BIN=11011111110001100111       HEX=DFC67
2700138 BIN=1010010011001101101010     HEX=29336A
9645053 BIN=100100110010101111111101   HEX=932BFD
1181110 BIN=100100000010110110110      HEX=1205B6
1492185 BIN=101101100010011011001      HEX=16C4D9
4018595 BIN=1111010101000110100011     HEX=3D51A3
7654888 BIN=11101001100110111101000    HEX=74CDE8
```

5. Uso de PyLint

Se instaló y ejecutó la herramienta PyLint para realizar análisis estático del código.

Inicialmente se detectaron advertencias relacionadas con docstrings, código duplicado y convenciones de estilo.

Todas las advertencias fueron corregidas agregando docstrings, refactorizando código duplicado y ajustando el estilo. Finalmente, el proyecto obtuvo una calificación de 10.00/10 en PyLint.

6. Control de versiones y entrega

El proyecto fue versionado utilizando Git y publicado en un repositorio de GitHub.

Para la autenticación se configuró el acceso mediante llaves SSH, permitiendo realizar el push del código de forma segura.

Con esto se cumple completamente la actividad 4.2, demostrando el uso de estándares decodificación, análisis estático y buenas prácticas de desarrollo de software.

