

# Instituto Tecnológico y de Estudios Superiores de Monterrey



## Pruebas de Software y Aseguramiento de la calidad 4.2 Ejercicio de programación 1

### Presentado por:

Maritza Liceth Guerrero Millan A01795976

### Profesores:

Dr. Gerardo Padilla Zárate

Mtra. María Mylen Treviño E.

01 de Febrero del 2025

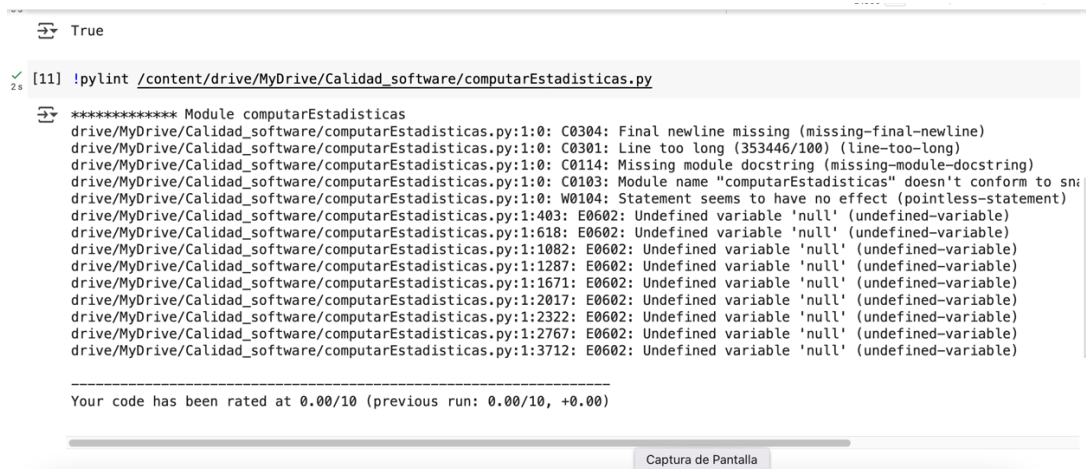
Las posibles validaciones

There are 5 kind of message types :

- \* (C) convention, for programming standard violation
- \* (R) refactor, for bad code smell
- \* (W) warning, for python specific problems
- \* (E) error, for probable bugs in the code
- \* (F) fatal, if an error occurred which prevented pylint from doing

## Programa 1

### 1) Validación inicial

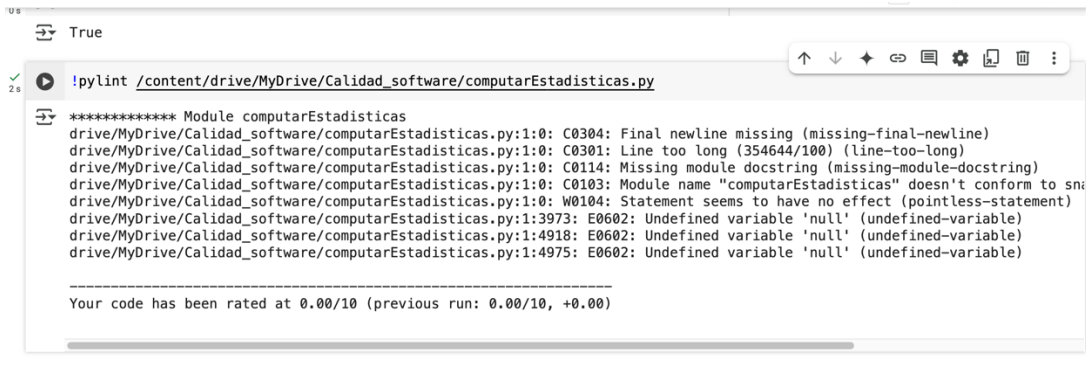


```
[11] !pylint /content/drive/MyDrive/Calidad_software/computarEstadisticas.py

***** Module computarEstadisticas
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: C0304: Final newline missing (missing-final-newline)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: C0301: Line too long (353446/100) (line-too-long)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: C0114: Missing module docstring (missing-module-docstring)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: C0103: Module name "computarEstadisticas" doesn't conform to snake case naming style
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: W0104: Statement seems to have no effect (pointless-statement)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:403: E0602: Undefined variable 'null' (undefined-variable)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:618: E0602: Undefined variable 'null' (undefined-variable)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:1082: E0602: Undefined variable 'null' (undefined-variable)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:1287: E0602: Undefined variable 'null' (undefined-variable)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:1671: E0602: Undefined variable 'null' (undefined-variable)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:2017: E0602: Undefined variable 'null' (undefined-variable)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:2322: E0602: Undefined variable 'null' (undefined-variable)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:2767: E0602: Undefined variable 'null' (undefined-variable)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:3712: E0602: Undefined variable 'null' (undefined-variable)

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

Se ajusta la primera



```
True

!pylint /content/drive/MyDrive/Calidad_software/computarEstadisticas.py

***** Module computarEstadisticas
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: C0304: Final newline missing (missing-final-newline)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: C0301: Line too long (354644/100) (line-too-long)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: C0114: Missing module docstring (missing-module-docstring)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: C0103: Module name "computarEstadisticas" doesn't conform to snake_case (invalid-name)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: W0104: Statement seems to have no effect (pointless-statement)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:3973: E0602: Undefined variable 'null' (undefined-variable)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:4918: E0602: Undefined variable 'null' (undefined-variable)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:4975: E0602: Undefined variable 'null' (undefined-variable)

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

E0602: Undefined variable 'null' (undefined-variable)

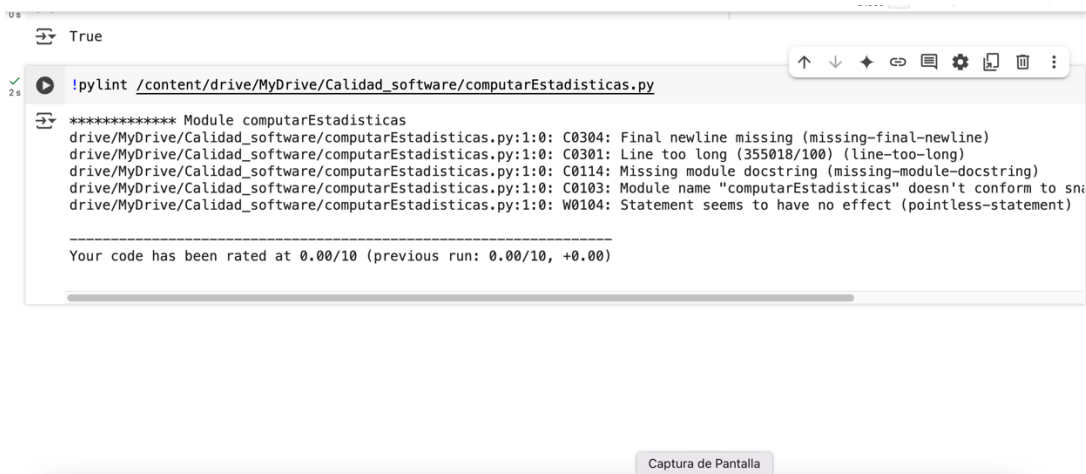
Cambiando la sentencia así:

Estaba antes:

```
print(f'Error: '{linea.strip()}' en {archivo} no es un número válido y será ignorado.")
```

Después de ajustada:

```
print(f'Error: '{linea.strip()}' en {archivo} no es un número válido y "
      "será ignorado.")
```



```
True

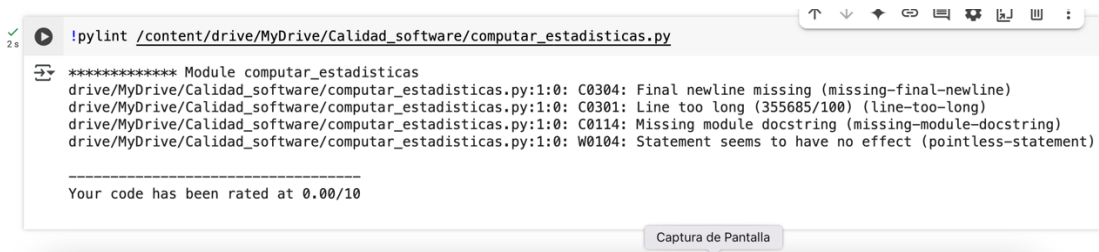
!pylint /content/drive/MyDrive/Calidad_software/computarEstadisticas.py

***** Module computarEstadisticas
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: C0304: Final newline missing (missing-final-newline)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: C0301: Line too long (355018/100) (line-too-long)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: C0114: Missing module docstring (missing-module-docstring)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: C0103: Module name "computarEstadisticas" doesn't conform to snake_case (invalid-name)
drive/MyDrive/Calidad_software/computarEstadisticas.py:1:0: W0104: Statement seems to have no effect (pointless-statement)

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

**C0103: Module name "computarEstadisticas" doesn't conform to snake\_case naming style (invalid-name)**

Renombre el archivo para que siga el estilo snake\_case. Lo cambie computarEstadisticas.py a computar\_estadisticas.py.



```
!pylint /content/drive/MyDrive/Calidad_software/computar_estadisticas.py
***** Module computar_estadisticas
drive/MyDrive/Calidad_software/computar_estadisticas.py:1:0: C0304: Final newline missing (missing-final-newline)
drive/MyDrive/Calidad_software/computar_estadisticas.py:1:0: C0301: Line too long (355685/100) (line-too-long)
drive/MyDrive/Calidad_software/computar_estadisticas.py:1:0: C0114: Missing module docstring (missing-module-docstring)
drive/MyDrive/Calidad_software/computar_estadisticas.py:1:0: W0104: Statement seems to have no effect (pointless-statement)

-----
Your code has been rated at 0.00/10
```

### **C0304: Final newline missing (missing-final-newline)**

Este aviso indica que falta una línea en blanco al final del archivo.

### **C0301: Line too long (354986/100) (line-too-long)**

La longitud de una línea es demasiado larga (357,655 caracteres).

### **C0114: Missing module docstring (missing-module-docstring)**

El archivo no tiene una docstring (comentario de descripción del módulo) al principio.

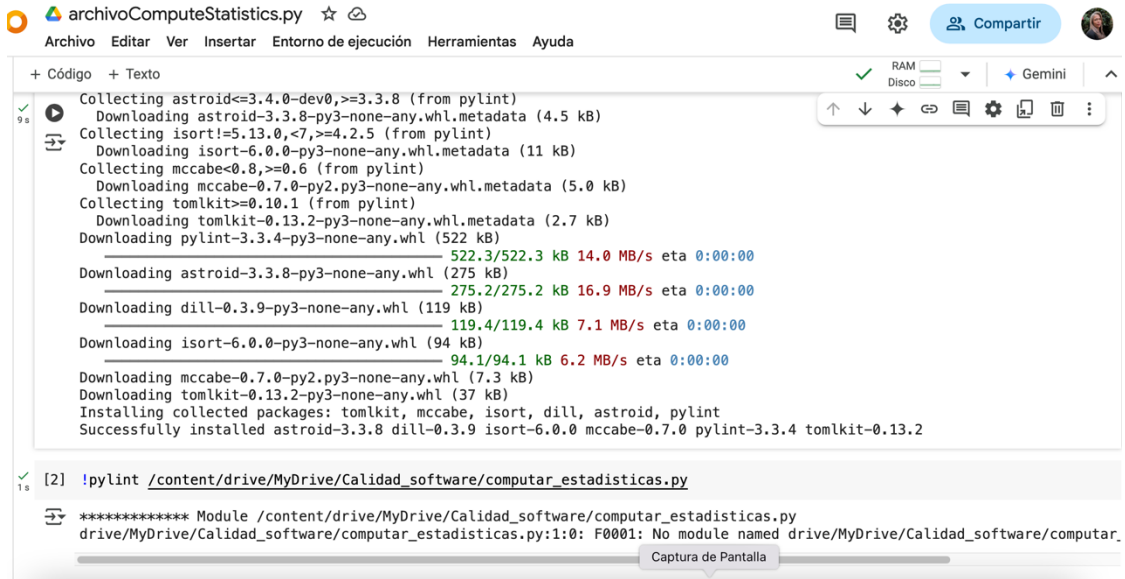
### **W0104: Statement seems to have no effect (pointless-statement)**

Este aviso indica que hay una instrucción que no tiene efecto.

Se ajustaron:

- 1) Se documentó con un texto cada función
- 2) Se ajustaron nombre de variables, por ejemplo tenía la variable `f` y la cambie por `file`, así mismo ocurrió con la variable `n` y la cambie por `numero`
- 3) Manejo de errores
- 4) Nombre de variables constantes, estaban en minúscula y las deje en mayúscula, por ejemplo, `carpeta_drive` y la deje como `CARPETA_DRIVE`, al igual paso con la variable `carpeta_salida` y la deje como `CARPETA_SALIDA`
- 5) Elimine librerías que las incluí al inicio y que luego no use
- 6) Mensajes en pantalla, minimice los textos, ya que la sentencia quedaba muy larga
- 7) El nombre de los archivos tuve que cambiarlos porque me indicó que los colocara muy diciente

Después de esto , al ejecutar el comando `pylint`, la salida ya salió limpia



```
archivoComputeStatistics.py ☆
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

+ Código + Texto

Collecting astroid<=3.4.0-dev0,>=3.3.8 (from pylint)
  Downloading astroid-3.3.8-py3-none-any.whl.metadata (4.5 kB)
Collecting isort!=5.13.0,<7,>=4.2.5 (from pylint)
  Downloading isort-6.0.0-py3-none-any.whl.metadata (11 kB)
Collecting mccabe<0.8,>=0.6 (from pylint)
  Downloading mccabe-0.7.0-py2.py3-none-any.whl.metadata (5.0 kB)
Collecting tomlkit>=0.10.1 (from pylint)
  Downloading tomlkit-0.13.2-py3-none-any.whl.metadata (2.7 kB)
Downloading pylint-3.3.4-py3-none-any.whl (522 kB)
522.3/522.3 kB 14.0 MB/s eta 0:00:00
Downloading astroid-3.3.8-py3-none-any.whl (275 kB)
275.2/275.2 kB 16.9 MB/s eta 0:00:00
Downloading dill-0.3.9-py3-none-any.whl (119 kB)
119.4/119.4 kB 7.1 MB/s eta 0:00:00
Downloading isort-6.0.0-py3-none-any.whl (94 kB)
94.1/94.1 kB 6.2 MB/s eta 0:00:00
Downloading mccabe-0.7.0-py2.py3-none-any.whl (7.3 kB)
Downloading tomlkit-0.13.2-py3-none-any.whl (37 kB)
Installing collected packages: tomlkit, mccabe, isort, dill, astroid, pylint
Successfully installed astroid-3.3.8 dill-0.3.9 isort-6.0.0 mccabe-0.7.0 pylint-3.3.4 tomlkit-0.13.2

[2] !pylint /content/drive/MyDrive/Calidad_software/computar_estadisticas.py

***** Module /content/drive/MyDrive/Calidad_software/computar_estadisticas.py
drive/MyDrive/Calidad_software/computar_estadisticas.py:1:0: F0001: No module named drive/MyDrive/Calidad_software/computar_

Captura de Pantalla
```

## Código del programa1

```
# -*- coding: utf-8 -*-
```

```
"""computar_estadisticas.py
```

Automatically generated by Colab.

Original file is located at

[https://colab.research.google.com/drive/15pVury8Xk1wZ5L\\_-jjwtPwos7NG2Y1Nx](https://colab.research.google.com/drive/15pVury8Xk1wZ5L_-jjwtPwos7NG2Y1Nx)

```
"""
```

```
from google.colab import drive
```

```
drive.mount('/content/drive', force_remount=True)
```

```
"""
```

Módulo para procesar archivos de texto y calcular estadísticas como:

- Media
- Mediana
- Moda
- Varianza

- Desviación estándar

Los resultados se almacenan en un archivo de salida y se muestran en pantalla.

```
"""

import time
import os
from collections import Counter

def leer_datos(archivo):
    """
    Lee los datos de un archivo y devuelve una lista de números.
    Ignora valores no numéricos y muestra un mensaje de error.
    """
    numeros = []
    with open(archivo, 'r', encoding='utf-8') as file:
        for linea in file:
            try:
                numeros.append(float(linea.strip()))
            except ValueError:
                print(
                    f'Error: '{linea.strip()}' en {archivo} no es un número válido "
                    "y será ignorado."
                )
    return numeros

def calcular_media(datos):
    """Calcula y devuelve la media de una lista de números."""
    return sum(datos) / len(datos) if datos else 0

def calcular_mediana(datos):
    """Calcula y devuelve la mediana de una lista de números."""
    datos_ordenados = sorted(datos)
    numeros = len(datos_ordenados)
```

```

mitad = numeros // 2
if numeros % 2 == 0:
    return (datos_ordenados[mitad - 1] + datos_ordenados[mitad]) / 2
return datos_ordenados[mitad]

def calcular_moda(datos):
    """
    Calcula y devuelve la moda de una lista de números.
    Si hay múltiples modas, devuelve una lista de modas.
    """
    contador = Counter(datos)
    max_frecuencia = max(contador.values())
    modas = [num for num, freq in contador.items() if freq == max_frecuencia]
    return modas if len(modas) > 1 else modas[0]

def calcular_varianza(datos, media):
    """Calcula y devuelve la varianza de una lista de números."""
    return sum((x - media) ** 2 for x in datos) / len(datos) if datos else 0

def calcular_desviacion_estandar(varianza):
    """Calcula y devuelve la desviación estándar a partir de la varianza."""

def escribir_resultados(archivo_salida, resultados_totales):
    """Escribe los resultados de cada archivo en un archivo de salida."""
    with open(archivo_salida, 'w', encoding='utf-8') as file:
        for resultados in resultados_totales:
            for linea in resultados:
                file.write(linea + '\n')
            file.write("\n") # Añadir una línea en blanco
            file.write("\n") # Línea en blanco final

def procesar_archivo(archivo_entrada):

```

```

"""Procesa un archivo, calcula estadísticas y devuelve los resultados."""
datos = leer_datos(archivo_entrada)
if not datos:
    print(f"No se encontraron datos válidos en {archivo_entrada}.")
    return None

total_registros = len(datos)
media = calcular_media(datos)
mediana = calcular_mediana(datos)
moda = calcular_moda(datos)
varianza = calcular_varianza(datos, media)
desviacion = calcular_desviacion_estandar(varianza)

resultados = [
    f"Archivo: {archivo_entrada}",
    f"Número de registros: {total_registros}",
    f"Media: {media}",
    f"Mediana: {mediana}",
    f"Moda: {moda}",
    f"Varianza: {varianza}",
    f"Desviación estándar: {desviacion}"
]

return resultados

def main():
    """
    Define las rutas de los archivos, procesa los datos, guarda los resultados
    y muestra en pantalla el tiempo de ejecución.
    """
    CARPETA_DRIVE = "/content/drive/MyDrive/Calidad_software/Datos1/"
    CARPETA_SALIDA = "/content/drive/MyDrive/Calidad_software/"
    os.makedirs(CARPETA_SALIDA, exist_ok=True)

```



```

archivos = [
    "TC1.txt", "TC2.txt", "TC3.txt",
    "TC4.txt", "TC5.txt", "TC6.txt",
    "TC7.txt"
]

rutas_archivos = [
    os.path.join(CARPETA_DRIVE, archivo)
    for archivo in archivos
]

resultados_totales = [] # Para almacenar los resultados
inicio = time.time() # Inicio del tiempo de ejecución

for archivo in rutas_archivos:
    if os.path.exists(archivo):
        resultados = procesar_archivo(archivo)
        if resultados:
            resultados_totales.append(resultados)
    else:
        print(f"Archivo no encontrado: {archivo}")

# Escribir todos los resultados en un solo archivo consolidado
archivo_salida = os.path.join(CARPETA_SALIDA, "StatisticsResults.txt")
escribir_resultados(archivo_salida, resultados_totales)

# Imprimir resultados en la pantalla
for resultados in resultados_totales:
    for linea in resultados:
        print(linea)

# Calcular y mostrar el tiempo de ejecución solo en la pantalla

```

```
fin = time.time()

print(f"Tiempo de ejecución: {fin - inicio:.5f} segundos")
```

```
if __name__ == "__main__":
    main()
```

## Programa2

### Primera ejecución comando pylint

```
!pylint /content/drive/MyDrive/Calidad_software/convertir_Numeros.py

***** Module convertir_Numeros
drive/MyDrive/Calidad_software/convertir_Numeros.py:1:0: C0304: Final newline missing (missing-final-newline)
drive/MyDrive/Calidad_software/convertir_Numeros.py:1:0: C0301: Line too long (37936/100) (line-too-long)
drive/MyDrive/Calidad_software/convertir_Numeros.py:1:0: C0114: Missing module docstring (missing-module-docstring)
drive/MyDrive/Calidad_software/convertir_Numeros.py:1:0: C0103: Module name "convertir_Numeros" doesn't conform to snake_case
drive/MyDrive/Calidad_software/convertir_Numeros.py:1:0: W0104: Statement seems to have no effect (pointless-statement)
drive/MyDrive/Calidad_software/convertir_Numeros.py:1:1718: E0602: Undefined variable 'null' (undefined-variable)

Your code has been rated at 0.00/10
```

Después de realizar los mismos cambios del programa 1, al ejecutar el comando pylint, la salida ya salió limpia

```
archivo convertNumbers.py
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda

+ Código + Texto

Downloading astroid-3.3.8-py3-none-any.whl.metadata (4.5 kB)
Collecting isort!=5.13.0,<7,>=4.2.5 (from pylint)
Downloading isort-6.0.0-py3-none-any.whl.metadata (11 kB)
Collecting mccabe<0.8,>=0.6 (from pylint)
Downloading mccabe-0.7.0-py2.py3-none-any.whl.metadata (5.0 kB)
Collecting tomlkit>=0.10.1 (from pylint)
Downloading tomlkit-0.13.2-py3-none-any.whl.metadata (2.7 kB)
Downloading pylint-3.3.4-py3-none-any.whl (522 kB)
522.3/522.3 kB 9.4 MB/s eta 0:00:00
Downloading astroid-3.3.8-py3-none-any.whl (275 kB)
275.2/275.2 kB 13.4 MB/s eta 0:00:00
Downloading dill-0.3.9-py3-none-any.whl (119 kB)
119.4/119.4 kB 6.2 MB/s eta 0:00:00
Downloading isort-6.0.0-py3-none-any.whl (94 kB)
94.1/94.1 kB 5.0 MB/s eta 0:00:00
Downloading mccabe-0.7.0-py2.py3-none-any.whl (7.3 kB)
Downloading tomlkit-0.13.2-py3-none-any.whl (37 kB)
Installing collected packages: tomlkit, mccabe, isort, dill, astroid, pylint
Successfully installed astroid-3.3.8 dill-0.3.9 isort-6.0.0 mccabe-0.7.0 pylint-3.3.4 tomlkit-0.13.2

[ ] !pylint /content/drive/MyDrive/Calidad_software/convertir_Numeros.py

***** Module /content/drive/MyDrive/Calidad_software/convertir_Numeros.py
drive/MyDrive/Calidad_software/convertir_Numeros.py:1:0: F0001: No module named drive/MyDrive/Calidad_software/convertir_Nur
```

## Código del programa2

```
# -*- coding: utf-8 -*-
```

```
"""convertir_Numeros.py
```

Automatically generated by Colab.

Original file is located at

<https://colab.research.google.com/drive/1u6kJS28VycgYZDy9hKUuj8yqo1vui5CW>

```
"""
```

```
'''
```

Programa que lee varios archivos y entrega

un archivo de salida con la conversion a

binario y hexadecimal

```
'''
```

```
from google.colab import drive
```

```
drive.mount('/content/drive', force_remount=True)
```

```
import os
```

```
import time
```

```
# Directorios en Google Drive
```

```
CARPETA_DRIVE = "/content/drive/MyDrive/Calidad_software/Datos2/"
```

```
CARPETA_SALIDA = "/content/drive/MyDrive/Calidad_software/"
```

```
def convertir_a_binario(numero):
```

```
    '''
```

Función para convertir un número a binario sin usar bin()

```
    '''
```

```
    if numero == 0:
```

```

        return "0"
    binario = ""
    while numero > 0:
        binario = str(numero % 2) + binario
        numero //= 2
    return binario

def convertir_a_hexadecimal(numero):
    """
    Función para convertir un número a hexadecimal sin usar hex()
    """
    if numero == 0:
        return "0"
    hex_chars = "0123456789ABCDEF"
    hexadecimal = ""
    while numero > 0:
        hexadecimal = hex_chars[numero % 16] + hexadecimal
        numero //= 16
    return hexadecimal

def procesar_archivo(ruta_archivo, archivo_entrada):
    """
    Una funcion en donde se realiza la conversion
    del numero en binario y hexadecimal
    """
    if not os.path.isfile(ruta_archivo):
        print(f'Error: El archivo '{ruta_archivo}' no existe.')
        return [], [f'Error: El archivo '{ruta_archivo}' no existe.']

    resultados = []
    errores = []

```

```

try:
    with open(ruta_archivo, 'r', encoding="utf-8") as file:
        lineas = file.readlines()
except FileNotFoundError as error:
    print(f"Error al leer el archivo: {error}")
    return [], [f"Error al leer el archivo: {error}"]

if not lineas:
    print("El archivo está vacío.")
    return [], ["Error: El archivo está vacío."]

item = 1
for linea in lineas:
    elemento = linea.strip()
    try:
        numero = int(elemento)
        binario = convertir_a_binario(numero)
        hexadecimal = convertir_a_hexadecimal(numero)
        resultados.append(f"{item}\t{numero}\t{archivo_entrada}\t{binario}\t{hexadecimal}")
        item += 1
    except ValueError:
        error_msg = f"Error: '{elemento}' no es un número válido."
        errores.append(error_msg)
        print(error_msg)

return resultados, errores

def main():
    """
    Se realiza todo el procesamiento de cada uno de los archivos
    y genera los resultados
    """

```

```

archivos = ["TC1.txt", "TC2.txt", "TC3.txt", "TC4.txt"]
# Lista de los archivos que deseas procesar
ruta_salida = os.path.join(CARPETA_SALIDA, "ConversionResults.txt")

# Abrir archivo de salida en modo escritura
with open(ruta_salida, 'w', encoding="utf-8") as salida:
    # Escribir los encabezados
    salida.write("Item\tNúmero\tArchivo\tBinario\tHexadecimal\n")

for archivo_entrada in archivos:
    ruta_archivo = os.path.join(CARPETA_DRIVE, archivo_entrada)

    print(f"Procesando archivo: {archivo_entrada}...")
    inicio_tiempo = time.time()

    resultados, errores = procesar_archivo(ruta_archivo, archivo_entrada)

    fin_tiempo = time.time()
    tiempo_transcurrido = fin_tiempo - inicio_tiempo

    # Imprimir resultados en consola
    if resultados:
        for resultado in resultados:
            print(resultado)

    if errores:
        for error in errores:
            print(error)

    print(f"Tejecución {archivo_entrada}: {tiempo_transcurrido:.4f} s")

# Guardar resultados en archivo de salida

```

```

    if resultados:
        salida.write(f"\nResultados para {archivo_entrada}:\n")
        for resultado in resultados:
            salida.write(resultado + '\n')
    if errores:
        salida.write(f"Errores en {archivo_entrada}:\n")
        for error in errores:
            salida.write(error + '\n')
    salida.write(f"Tejecución para {archivo_entrada}: {tiempo_transcurrido:.4f} s\n")

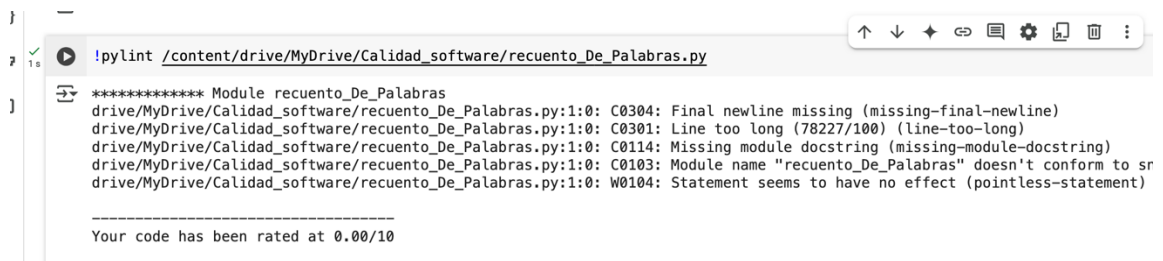
print(f"Resultados guardados en: {ruta_salida}")

if __name__ == "__main__":
    main()

```

### Programa3

#### Primera ejecución comando pylint



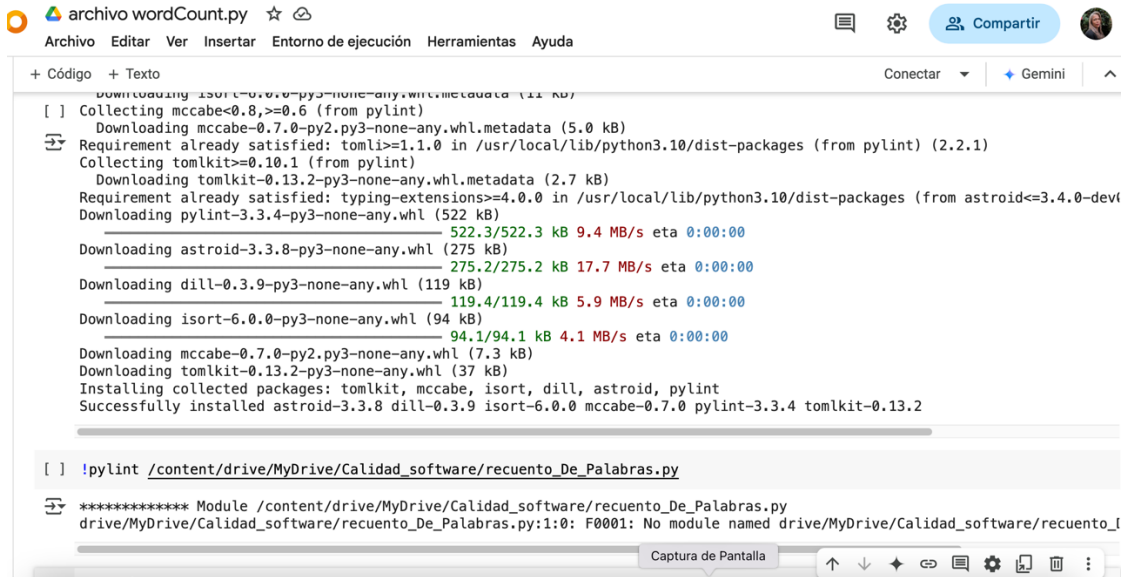
```

!pylint /content/drive/MyDrive/Calidad_software/recuento_De_Palabras.py
***** Module recuento_De_Palabras
drive/MyDrive/Calidad_software/recuento_De_Palabras.py:1:0: C0304: Final newline missing (missing-final-newline)
drive/MyDrive/Calidad_software/recuento_De_Palabras.py:1:0: C0301: Line too long (78227/100) (line-too-long)
drive/MyDrive/Calidad_software/recuento_De_Palabras.py:1:0: C0114: Missing module docstring (missing-module-docstring)
drive/MyDrive/Calidad_software/recuento_De_Palabras.py:1:0: C0103: Module name "recuento_De_Palabras" doesn't conform to sr
drive/MyDrive/Calidad_software/recuento_De_Palabras.py:1:0: W0104: Statement seems to have no effect (pointless-statement)

Your code has been rated at 0.00/10

```

Después de realizar los mismos cambios del programa 1, al ejecutar el comando pylint, la salida ya salió limpia



```
archivo wordCount.py ☆ ☁
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda
+ Código + Texto Conectar Gemini
[ ] Collecting mccabe<0.8,>=0.6 (from pylint)
  Downloading mccabe-0.7.0-py2.py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: toml<=1.1.0 in /usr/local/lib/python3.10/dist-packages (from pylint) (2.2.1)
Collecting tomlkit>=0.10.1 (from pylint)
  Downloading tomlkit-0.13.2-py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.10/dist-packages (from astroid<=3.4.0-dev)
  Downloading astroid-3.3.8-py3-none-any.whl (522.3 kB)
  522.3/522.3 kB 9.4 MB/s eta 0:00:00
  Downloading dill-0.3.9-py3-none-any.whl (119 kB)
  119.4/119.4 kB 5.9 MB/s eta 0:00:00
  Downloading isort-6.0.0-py3-none-any.whl (94 kB)
  94.1/94.1 kB 4.1 MB/s eta 0:00:00
  Downloading mccabe-0.7.0-py2.py3-none-any.whl (7.3 kB)
  Downloading tomlkit-0.13.2-py3-none-any.whl (37 kB)
Installing collected packages: tomlkit, mccabe, isort, dill, astroid, pylint
Successfully installed astroid-3.3.8 dill-0.3.9 isort-6.0.0 mccabe-0.7.0 pylint-3.3.4 tomlkit-0.13.2

[ ] !pylint /content/drive/MyDrive/Calidad_software/recuento_De_Palabras.py
***** Module /content/drive/MyDrive/Calidad_software/recuento_De_Palabras.py
drive/MyDrive/Calidad_software/recuento_De_Palabras.py:1:0: F0001: No module named drive/MyDrive/Calidad_software/recuento_I
```

### Código del programa3

```
# -*- coding: utf-8 -*-
```

```
"""recuento_De_Palabras.py
```

Automatically generated by Colab.

Original file is located at

<https://colab.research.google.com/drive/1VNqygi-uGEQiZl1Cu6Pwml9S1LaAXALf>

```
"""
```

```
from google.colab import drive
```

```
drive.mount('/content/drive', force_remount=True)
```

```
import time
```

```
import os
```

```
# Rutas de trabajo
```

```
CARPETA_DRIVE = "/content/drive/MyDrive/Calidad_software/Datos3/"
```

```
CARPETA_SALIDA = "/content/drive/MyDrive/Calidad_software/"
```

```
def contar_palabras(archivo):
```



```

"""
Funcion que cuenta las palabras iguales
"""

conteo_palabras = {}

try:
    with open(archivo, 'r', encoding='utf-8') as file:
        for linea in file:
            palabras = linea.strip().split() # División manual sin usar regex
            for palabra in palabras:
                palabra = ".join([c.lower() for c in palabra if c.isalnum()]) # Normalización
                if palabra:
                    if palabra in conteo_palabras:
                        conteo_palabras[palabra] += 1
                    else:
                        conteo_palabras[palabra] = 1
except FileNotFoundError as error:
    print(f'Error al procesar el archivo {archivo}: {error}')

return conteo_palabras


def guardar_resultados(conteo_palabras, tiempo_ejecucion):
    """
    funcion que guarda el resultado en el archivo
    """

    salida_path = os.path.join(CARPETA_SALIDA, "WordCountResults.txt")

    try:
        with open(salida_path, 'a', encoding='utf-8') as file: # Usamos 'a' para agregar al archivo
            for palabra, frecuencia in conteo_palabras.items():
                file.write(f'{palabra}: {frecuencia}\n')
            file.write(f'\nTiempo de ejecución: {tiempo_ejecucion:.6f} segundos\n')
            file.write("\n" + "-"*50 + "\n") # Línea separadora para claridad entre archivos
            print(f'Resultados guardados en: {salida_path}')
    except FileNotFoundError as error:

```

```

    print(f'Error al escribir el archivo de resultados: {error}')

def main():
    """
    ejecucion de todo el programa
    """
    archivos = ['TC1.txt', 'TC2.txt', 'TC3.txt', 'TC4.txt', 'TC5.txt']

    # Borrar el archivo de resultados previo si existe
    if os.path.exists(os.path.join(CARPETA_SALIDA, "WordCountResults.txt")):
        os.remove(os.path.join(CARPETA_SALIDA, "WordCountResults.txt"))

    for archivo_nombre in archivos:
        archivo = os.path.join(CARPETA_DRIVE, archivo_nombre)
        if not os.path.exists(archivo):
            print(f'Error: El archivo '{archivo}' no existe.")
            continue

        inicio = time.time()
        conteo_palabras = contar_palabras(archivo)
        fin = time.time()
        tiempo_ejecucion = fin - inicio

        # Mostrar resultados en pantalla
        print(f"\nResultados para el archivo {archivo_nombre}:")
        for palabra, frecuencia in conteo_palabras.items():
            print(f"{palabra}: {frecuencia}")
        print(f"\nTiempo de ejecución: {tiempo_ejecucion:.6f} segundos")

        # Guardar resultados en un archivo de salida único
        guardar_resultados(conteo_palabras, tiempo_ejecucion)

if __name__ == "__main__":

```

`main()`