



Actividad 5.2

Alumna:

Guadalupe Esmeralda González Maldonado (Ao1795767)

Maestría en Inteligencia Artificial Aplicada

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

Profesor Titular: Dr. Gerardo Padilla Zárate

Febrero 2025

Contenido

ANÁLISIS DE FLAKE8.....3

Conclusión de Flake8:.....3

EJERCICIO DE PROGRAMACIÓN 2.....3

 TC1.....3

 TC2.....4

 TC3.....5

CONCLUSIÓN5

Análisis de Flake8

Para asegurar la calidad del código, se ejecutó Flake8 como herramienta de análisis estático. A lo largo del desarrollo, se encontraron y corrigieron los siguientes errores:

Errores de espaciado y líneas en blanco (E302, E303, E305):

Se ajustaron los espacios entre funciones y estructuras para cumplir con PEP8.

Errores de formato (W292, W391):

Se corrigieron líneas en blanco al final del archivo y la falta de una nueva línea final.

Llamadas a Flake8 dentro del código:

Se implementó la opción `--lint` para permitir que el usuario analice el código con Flake8 de manera automática.

Conclusión de Flake8:

El código final cumple con los estándares de PEP8, lo que garantiza su claridad, mantenibilidad y conformidad con buenas prácticas de desarrollo en Python.

Ejercicio de programación 2

TC1

- **Costo total de ventas:** 2481.86
- **Tiempo de ejecución:** 0.001986 segundos
- **Errores detectados:** Ninguno
- **Observaciones:**
 - El programa procesó correctamente los datos sin errores.
 - Todos los productos en `salesRecord.json` coincidieron con los de `priceCatalogue.json`.

```

b2 C:/n2e2/258ad/Donwio9q2/v2's vlcμllo2 qe vbo2o>
j1e2b2o qe j1e2c2c2j2o: 0'00j28e 2e2b2u2q2o
c2o2j2o f2o2f2j qe 2e2u2f2e2: 5q8j'8e
ou
cμllo2 qe vbo2o/1c2j/b2j2e2c2f2e2j2o2b2e'j2o2u' "C:/n2e2/258ad/Donwio9q2/v2's vlcμllo2 qe vbo2o/1c2j/2e2j2e22e2c2o2q'j2e
b2 C:/n2e2/258ad/Donwio9q2/v2's vlcμllo2 qe vbo2o> b2f2mou c2omb2n2f2e22e2j2e2'2l "C:/n2e2/258ad/Donwio9q2/v2's vlc
n2o: b2f2mou c2omb2n2f2e22e2j2e2'2l b2j2e2c2f2e2j2o2b2e'j2o2u 2e2j2e22e2c2o2q'j2o2u
\b2f2mou232 e2q\b2f2mou'e2e' "C:/n2e2/258ad/Donwio9q2/v2's vlcμllo2 qe vbo2o\c2omb2n2f2e22e2j2e2'2l"
b2 C:/n2e2/258ad/Donwio9q2/v2's vlcμllo2 qe vbo2o> g "C:\b2o2e2u2w f2j2e2 (x8e)\j2j2c2o2o2f2e2 j2e2n2e2j 2f2n2q2j2o2j2u2e2q
b2o2b2f2e2w2 1 ON2b2N2L DEB2N2e C2o2j2o2f2e2 j2e2u2w2v2f b2o2j2e2
b2f2mou + ^ □ □ □ ... v x
53 j2f2 b2o2q2n2c2f'2u2w2e 2u2q j2e2j2u2f2u2c2e(b2j2e2' (j2u2f' f2j2o2f)):
55 b2j2e2 = j2f2e2w'8e2f("b2j2e2")
57 b2o2q2n2c2f'2u2w2e = j2f2e2w'8e2f("f2j2f2e2")
59 f2o2u j2f2e2w j2u b2j2e2'c2f2e2j2o2b2e2:
j2o b2j2e2'q2j2c2f = {}
j28 ""C2o2u2j2e2f2e2 j2e2 j2e2f2e2 qe b2o2q2n2c2f2o2 2u n2u q2j2c2j2o2u2e2j2o2 c2o2u b2e2c2j2o2' ""
j23 qe2f c2e2f2e2' b2j2e2'q2j2c2f(b2j2e2'c2f2e2j2o2b2e2):
j2e
j2e
j24 l2e2f2u2w j2u2o2e
j23 b2j2u2f(j2e2f2e2l2o2 2j j2e2e2 {f2j2e2u2w2e}): {e2}
j25 e2x2c2e2f (f2j2e2j2o2f2o2n2u2q2e2l2o2' j2e2o2u'j2e2o2w2e2c2o2q2e2f2e2l2o2) 2e2 e2:
j27 l2e2f2u2w j2e2o2u'j2o2q2 (f2j2e2)
j29 m2j2f2u o2b2e2u(f2j2e2u2w2e' '2' ' 2u2c2o2q2j2u2e2=,n2f2e2-8,) 2e2 f2j2e2:
j2o f2u2l:
j28 ""j2e2e2 n2u 2e2c2μj2o2 j2e2o2u 2e2 l2e2f2o2u2w 2u c2o2n2f2e2j2o2' ""
j23 qe2f l2e2q'j2e2o2u'f2j2e2(f2j2e2u2w2e):
j2e
j2e
j24 j2u2w2o2u2f o2
j23 j2u2w2o2u2f f2j2u2e
j25 j2u2w2o2u2f j2e2o2u
j27 j2u2w2o2u2f 222e
c2omb2n2f2e22e2j2e2'2l > l2e2q'j2e2o2u'2j2e2
c2omb2n2f2e22e2j2e2'2l x {} 2e2j2e22e2c2o2q2j2e2o2u {} b2j2e2'c2f2e2j2o2b2e2j2e2o2u
▶ ^ □ □ □ ...

```

TC2

- Costo total de ventas: 169478.22
- Tiempo de ejecución: 0.001067 segundos
- Errores detectados:
 - Cantidad inválida para Fresh blueberries:-35
 - Cantidad inválida para Green smoothie:-123
- Observaciones:
 - El programa identificó correctamente cantidades negativas y reportó los errores.
 - A pesar de los errores, el programa continuó ejecutándose sin interrupciones.

```

PS C:\Users\52899\Downloads\A5.2 Archivos de Apoyo> python computeSales.py PriceCatalogue.json "C:\Users\528
99\Downloads\A5.2 Archivos de Apoyo\TC2\TC2.Sales.json"
Costo total de ventas: 169478.22
Tiempo de Ejecución: 0.001067 segundos
Errores encontrados:
- Cantidad inválida para Fresh blueberries: -35
- Cantidad inválida para Green smoothie: -123
PS C:\Users\52899\Downloads\A5.2 Archivos de Apoyo>

```

TC3

- **Costo total de ventas:** 168145.36
- **Tiempo de ejecución:** 0.002035 segundos
- **Errores detectados:**
 - Producto no encontrado: Elotes
 - Producto no encontrado: Frijoles
 - Cantidad inválida para Fresh blueberries:-35
 - Cantidad inválida para Green smoothie:-123
- **Observaciones:**
 - Además de cantidades inválidas, se encontraron productos que no estaban en el catálogo.
 - El programa manejó ambos tipos de errores correctamente y generó un reporte detallado.

```
PS C:\Users\52899\Downloads\A5.2 Archivos de Apoyo> python computeSales.py PriceCatalogue.json "C:\Users\52899\Downloads\A5.2 Archivos de Apoyo\TC3\TC3.Sales.json"
Costo total de ventas: 168145.36
Tiempo de Ejecución: 0.002035 segundos
Errores encontrados:
- Producto no encontrado: Elotes
- Cantidad inválida para Fresh blueberries: -35
- Producto no encontrado: Frijoles
- Cantidad inválida para Green smoothie: -123
PS C:\Users\52899\Downloads\A5.2 Archivos de Apoyo> |
```

Conclusión

1. **Funcionamiento Correcto:**
 - El programa ejecutó con éxito los tres casos de prueba, demostrando su capacidad para calcular ventas y detectar errores sin interrupciones.
2. **Manejo de Errores:**
 - Identificó y reportó cantidades inválidas (negativas) en TC2 y TC3.
 - Detectó productos inexistentes en TC3, lo que asegura una validación adecuada de los datos.
3. **Eficiencia:**
 - El tiempo de ejecución fue muy bajo en todas las pruebas, lo que demuestra que el programa es eficiente incluso con grandes volúmenes de datos.
4. **Sugerencias de Mejora:**

- **Mejor manejo de errores:** En lugar de solo reportar errores, podría implementarse una opción para corregirlos automáticamente.
- **Log de errores:** Guardar errores en un archivo separado podría facilitar su revisión.
- **Validación previa:** Antes de procesar los datos, validar la existencia de productos y valores numéricos podría evitar errores tempranos.