



## **Pruebas de software y aseguramiento de la calidad**

### **5.2 Ejercicio de programación**

#### **Maestría en Inteligencia Artificial Aplicada (MNA-V)**

**Francisco Medellin Zertuche A01794044**

**Febrero 10 de 2024**

Problema Compute Sales:

A continuación se muestra en la parte izquierda los diferentes archivos del repositoria y en la derecha el resultado usando el primero archivo TC1 al ejecutar el código

Sort

View

Details

Name	Date modified	Type	Size
.pytest_cache	2/10/2024 5:39 PM	File folder	
__pycache__	2/10/2024 6:17 PM	File folder	
TC1	2/9/2024 6:29 PM	File folder	
TC2	2/10/2024 5:56 PM	File folder	
TC3	2/10/2024 5:56 PM	File folder	
compute_sales	2/11/2024 1:25 AM	Python Source File	3 KB
flake8_1	2/11/2024 1:23 AM	PNG File	22 KB
flake8_2	2/11/2024 1:26 AM	PNG File	37 KB
pylint_1	2/10/2024 7:06 PM	PNG File	96 KB
pylint_2	2/11/2024 1:22 AM	PNG File	12 KB
PYTEST	2/11/2024 1:22 AM	PNG File	49 KB
TC1.SalesResults	2/10/2024 3:46 PM	Text Document	1 KB
TC2.SalesResults	2/10/2024 5:56 PM	Text Document	1 KB
TC3.SalesResults	2/10/2024 5:57 PM	Text Document	1 KB
test_compute_sales	2/11/2024 1:26 AM	Python Source File	3 KB

Total cost of all sales: \$2481.86

Execution time: 0.0156 seconds

Resultado en la ejecucion TC2:

TC2.SalesResults

File Edit View

Total cost of all sales: \$166568.23

Execution time: 0.0010 seconds

Resultado de la ejecucion TC3:

TC3.SalesResults

File Edit View

Total cost of all sales: \$165235.37

Execution time: 0.0351 seconds

## Analisis de pruebas estaticas Pylint

En la siguiente imagen podemos observar los cambios que pylint recomienda realizar en nuestro codigo.

```
:\\ase) C:\\Users\\v-fzertuche\\Documents\\Actividad_5_2>pylint computeSales.py C:\\
***** Module
C:\\__init__.py:1:0: F0010: error while code parsing: Unable to load file C:\\__init__.py:
[Errno 2] No such file or directory: 'C:\\__init__.py' (parse-error)
***** Module computeSales
computeSales.py:26:8: C0303: Trailing whitespace (trailing-whitespace)
computeSales.py:46:0: C0301: Line too long (101/100) (line-too-long)
computeSales.py:48:0: C0303: Trailing whitespace (trailing-whitespace)
computeSales.py:83:0: C0304: Final newline missing (missing-final-newline)
computeSales.py:1:0: C0114: Missing module docstring (missing-module-docstring)
computeSales.py:1:0: C0103: Module name "computeSales" doesn't conform to snake_case naming style (invalid-name)
computeSales.py:11:13: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
computeSales.py:21:0: R1710: Either all return statements in a function should return an expression, or none of them should. (inconsistent-return-statements)
computeSales.py:73:9: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)

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

El mensaje de error generado por Pylint indica problemas que pueden mejorarse en el código para cumplir con las convenciones y estándares de Python. Aquí hay algunas sugerencias para abordar los problemas señalados:

- Trailing newlines (trailing-newlines): Este aviso se refiere a tener líneas en blanco adicionales al final del archivo. Se necesita eliminar estas líneas en blanco al final del archivo.
- Missing module docstring (missing-module-docstring): Se debe agregar una cadena de documentación al comienzo del archivo para describir el propósito y la funcionalidad del módulo.
- Module name "computeSales" doesn't conform to snake\_case naming style (invalid-name): Es necesario cambiar el nombre del módulo a "compute\_Sales" para seguir la convención de nombres en snake\_case.

Una vez corregidos estos puntos tenemos el siguiente resultado:

```
(base) C:\\Users\\v-fzertuche\\Documents\\Actividad_5_2>pylint compute_sales.py

-----
Your code has been rated at 10.00/10 (previous run: 0.00/10, +10.00)
```

## Analisis de pruebas estaticas Flake8

En la siguiente imagen podemos observar los cambios que flake8 recomienda realizar en nuestro codigo.

```
(base) C:\Users\v-fzertuche\Documents\Actividad_5_2>flake8 compute_sales.py
compute_sales.py:49:80: E501 line too long (95 > 79 characters)
compute_sales.py:58:80: E501 line too long (94 > 79 characters)
compute_sales.py:62:80: E501 line too long (88 > 79 characters)
```

El mensaje de error que se esta recibiendo al ejecutar Flake8 indica que algunas líneas del archivo compute\_sales.py son demasiado largas según el estándar de estilo definido (probablemente PEP8). El límite de ancho de línea predeterminado en PEP8 es de 79 caracteres.

Para resolver estos problemas de líneas demasiado largas, hay varias opciones:

- Dividir las líneas largas
- Ignorar líneas específicas

Al momento de corregir los puntos dados por flake8 y volver a ejecutar la prueba obtenemos el siguiente resultado.

```
(base) C:\Users\v-fzertuche\Documents\Actividad_5_2>flake8 compute_sales.py
```

Podemos observar que todos los errores ya no se muestran.

## Analisis de pruebas Dinamicas

Para este ejercicio se tienen diferentes tipos de pares de archivos de ejemplo para realizar tests ejecutando nuestro codigo y verificando los resultados. Los archivos en las carpetas TC1, TC2 y TC3 hacen referencia a los diferentes tests que se realizaron en el programa.

Como ya se menciona anteriormente los resultados fueron exactamente los mismos proporcionados por el maestro.

El archivo test\_compute\_sales contiene el codigo que realizo las pruebas dinamicas del programa.

TC2	2/10/2024 5:56 PM	File folder	
TC3	2/10/2024 5:56 PM	File folder	
compute_sales	2/11/2024 1:25 AM	Python Source File	3 KB
flake8_1	2/11/2024 1:23 AM	PNG File	22 KB

Codigo test\_compute\_sales:

```
test_compute_sales.py X
test_compute_sales.py > ...
11 },
12 "TC2": {
13     "price_catalogue": PATH + "/TC2/TC2.Productlist.json",
14     "sales_record": PATH + "/TC2/TC2.Sales.json",
15 },
16 "TC3": {
17     "price_catalogue": PATH + "/TC3/TC3.Productlist.json",
18     "sales_record": PATH + "/TC3/TC3.Sales.json",
19 }
20 }
21 RESULTS = {
22     "TC1": 2481.86,
23     "TC2": 166568.23,
24     "TC3": 165235.37
25 }
26
27
28 @pytest.mark.parametrize(
29     "filename",
30     [
31         (TCFILES["TC1"]["price_catalogue"],
32         TCFILES["TC2"]["price_catalogue"],
33         TCFILES["TC3"]["price_catalogue"])
34     ]
35 )
36 def test_load_json_file_existing(filename):
37     """
38     Prueba para verificar la carga de un archivo JSON existente
39     """
40     data = compute_sales.load_json_file(filename)
41     assert data is not None
42
43
44 def test_load_json_file_invalid_format():
45     """Prueba para verificar el manejo de un archivo JSON con formato incorrecto"""
46     data = compute_sales.load_json_file("invalid_format.json")
47     assert data is None
48
49 @pytest.mark.parametrize(
50     "price_catalogue,sales_record,result",
51     [
52         (TCFILES["TC1"]["price_catalogue"],TCFILES["TC1"]["sales_record"], RESULTS["TC1"]) ,
53         (TCFILES["TC2"]["price_catalogue"],TCFILES["TC2"]["sales_record"], RESULTS["TC2"]) ,
54         (TCFILES["TC3"]["price_catalogue"],TCFILES["TC3"]["sales_record"], RESULTS["TC3"])
55     ]
56 )
```

A continuacion se muestran los resultados de los tests,obteniendo resultados satisfactorios:

```
PS C:\Users\v-fzertuche\Documents\Actividad_5_2> & C:/ProgramData/anaconda3/python.exe c:/Users/v-fzertuche/
Documents/Actividad_5_2/test_computeSales.py
===== test session starts =====
platform win32 -- Python 3.11.5, pytest-7.4.0, pluggy-1.0.0
rootdir: C:\Users\v-fzertuche\Documents\Actividad_5_2
plugins: anyio-3.5.0
collected 7 items

test_computeSales.py ..... [100%]

===== 7 passed in 0.03s =====
```