



Instituto Tecnológico y de Estudios

Superiores de Monterrey

Maestría en Inteligencia Artificial Aplicada (MNA)

Pruebas de software y aseguramiento de la calidad (Grupo 10)

Tarea Individual - Actividad 5.2

Ejercicios de Programación 2

Estudiante:

Ana Cristina Torres Cordero

A00831285

3 de febrero del 2025

Monterrey. Nuevo León

Análisis de requisitos

1. Introducción

En el presente trabajo, se realizó un programa en Python llamado `computeSales.py` que calcula el costo total de las ventas a partir de dos archivos: un catálogo de precios de productos y las ventas de la compañía en formato JSON. El programa identifica errores en los archivos y devuelve el tiempo de ejecución total.

2. Compute Sales

Se creó el programa con los requerimientos especificados en la actividad. Se adaptó el código para que sea capaz de recibir varios archivos de ventas y calcular el costo a partir de un solo catálogo de productos, utilizando los archivos JSON que se adjuntan en la actividad.

Inicialmente, el programa consideraba las ventas negativas como errores. Por ejemplo, el producto Green smoothie tiene una cantidad de ventas de -123 en el archivo `TC2.Sales.json`; por lo que el programa marcaba un error y no lo contaba en el precio total. Sin embargo, al ver el resultado esperado del programa, se ajustó el código para considerarlo como una devolución y contarlo en el cálculo.

Resultados esperados:

	TOTAL
TC1	2481.86
TC2	166568.23
TC3	165235.37

Resultados obtenidos (Almacenados en SalesResults.txt):

```
%run computeSales.py TC1.ProductList.json TC1.Sales.json TC2.Sales.json TC3.Sales.json

Tiempo de ejecución: 0.0127 segundos

Archivo: TC1.Sales.json
Costo total de las ventas: 2481.86 unidades monetarias

Archivo: TC2.Sales.json
Costo total de las ventas: 166568.23 unidades monetarias

Archivo: TC3.Sales.json
Costo total de las ventas: 165235.37 unidades monetarias
Errores detectados:
Producto no encontrado en catálogo: Elotes
Producto no encontrado en catálogo: Frijoles
```

Como se observa, el programa indica los errores detectados en cada uno de los archivos después del resultado.

Flake 8

La primera vez que se corrió la prueba con Flake8 se obtuvieron los siguientes resultados:

```
!flake8 computeSales.py

computeSales.py:5:1: E302 expected 2 blank lines, found 1
computeSales.py:14:1: E302 expected 2 blank lines, found 1
computeSales.py:27:1: E302 expected 2 blank lines, found 1
computeSales.py:28:80: E501 line too long (101 > 79 characters)
computeSales.py:33:1: W293 blank line contains whitespace
computeSales.py:43:80: E501 line too long (90 > 79 characters)
computeSales.py:46:80: E501 line too long (112 > 79 characters)
computeSales.py:47:1: W293 blank line contains whitespace
computeSales.py:51:1: E302 expected 2 blank lines, found 1
computeSales.py:54:80: E501 line too long (106 > 79 characters)
computeSales.py:64:80: E501 line too long (91 > 79 characters)
computeSales.py:66:1: W293 blank line contains whitespace
computeSales.py:68:1: W293 blank line contains whitespace
computeSales.py:74:1: W293 blank line contains whitespace
computeSales.py:76:80: E501 line too long (82 > 79 characters)
computeSales.py:78:1: W293 blank line contains whitespace
computeSales.py:81:1: W293 blank line contains whitespace
computeSales.py:84:80: E501 line too long (115 > 79 characters)
computeSales.py:87:1: W293 blank line contains whitespace
computeSales.py:89:1: W293 blank line contains whitespace
computeSales.py:93:1: E305 expected 2 blank lines after class or function definition, found 1
computeSales.py:94:11: W292 no newline at end of file
```

Se corrigieron todos los errores

```
!flake8 computeSales.py
```

PyLint

Se corrigieron los errores con PyLint y se obtuvo la calificación de 9.85/10 por el nombre del archivo:

```
!pylint computeSales.py
***** Module computeSales
computeSales.py:1:0: C0103: Module name "computeSales" doesn't conform to snake_case naming style (invalid-name)

-----
Your code has been rated at 9.85/10 (previous run: 9.71/10, +0.15)
```

El nombre computeSales.py es parte del requisito del programa, por lo que se deshabilitó esa corrección de PyLint para llegar al 10/10:

```
!pylint --disable=invalid-name computeSales.py

-----
Your code has been rated at 10.00/10 (previous run: 9.85/10, +0.15)
```