Tecnológico de Monterrey

Pruebas de software y aseguramiento de la calidad

Trimestre Enero-Marzo 2026

Ejercicio de Programación 1: Actividad 4.2

Jorge Muñoz Estrada

Fecha de entrega: 15 de Febrero de 2026

Repositorio Git: `https://github.com/a01213938/a01213938_A5.2`

1. Análisis Estático de Código

Como parte de las prácticas de aseguramiento de calidad, se utilizaron herramientas de análisis estático para verificar el cumplimiento del estándar **PEP-8** y detectar posibles errores lógicos o de complejidad sin necesidad de ejecutar el programa.

Resultados de Pylint y Flake8

- **Pylint:** Se alcanzó una calificación de **10.00/10**, asegurando que el código no presenta problemas de convención, refactorización o diseño.
- **Flake8:** La ejecución no arrojó advertencias ni errores (códigos E, W, F o C90), lo que confirma una estructura de código limpia y una complejidad ciclomática controlada.

```
(base) PS C:\Users\Jorge\Documents\Maestria\a01213938_A5.2\P1\source> flake8 comp
uteSales.py
(base) PS C:\Users\Jorge\Documents\Maestria\a01213938_A5.2\P1\source> python -m p
ylint computeSales.py

--------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 9.74/10, +0.26)
```

2. Pruebas de Ejecución Dinámica

Se realizaron pruebas dinámicas para validar la funcionalidad del programa **computeSales.py** frente a diferentes volúmenes de datos y escenarios de error. El sistema calcula el costo total cruzando un catálogo de precios en formato JSON con registros de ventas.

Caso de Prueba 1: Validación Base

- **Archivo:** TC1.ProductList.json / TC1.Sales.json
- **Resultado:** $2,481.86
- **Tiempo:** 0.0000 segundos
- **Estatus:** Exitoso.

```
(base) PS C:\Users\Jorge\Documents\Maestria\a01213938_A5.2\P1\source> python comp
uteSales.py TC1.ProductList.json TC1.Sales.json
--- TOTAL SALES REPORT ---
Total Cost: $2,481.86
Execution Time: 0.0000 seconds
-------------------------
```

Caso de Prueba 2: Volumen Intermedio

- **Archivo:** TC1.ProductList.json / TC2.Sales.json
- **Resultado:** $166,568.23
- **Estatus:** Exitoso.

```
(base) PS C:\Users\Jorge\Documents\Maestria\a01213938_A5.2\P1\source> python comp
uteSales.py TC1.ProductList.json TC2.Sales.json
--- TOTAL SALES REPORT ---
Total Cost: $166,568.23
Execution Time: 0.0000 seconds
```

Caso de Prueba 3: Volumen Masivo y Manejo de Errores

Este caso valida el **Requerimiento 3** de la actividad, que exige la capacidad de gestionar datos inválidos sin interrumpir la ejecución.

- **Incidencias detectadas:** Productos "Elotes" y "Frijoles" no encontrados en el catálogo.
- **Resultado Total:** $165,235.37
- **Estatus:** Exitoso.

```
(base) PS C:\Users\Jorge\Documents\Maestria\a01213938_A5.2\P1\source> python comp
uteSales.py TC1.ProductList.json TC3.Sales.json
Producto no encontrado: Elotes
Producto no encontrado: Frijoles
--- TOTAL SALES REPORT ---
Total Cost: $165,235.37
Execution Time: 0.0010 seconds
```

3. Conclusiones Técnicas

La integración de herramientas de análisis estático (Flake8, Pylint) permitió reducir la deuda técnica y asegurar que el código sea mantenible. Por otro lado, la ejecución de los casos de prueba confirmó que los algoritmos de búsqueda y cálculo son eficientes, incluso al procesar miles de registros en fracciones de segundo.