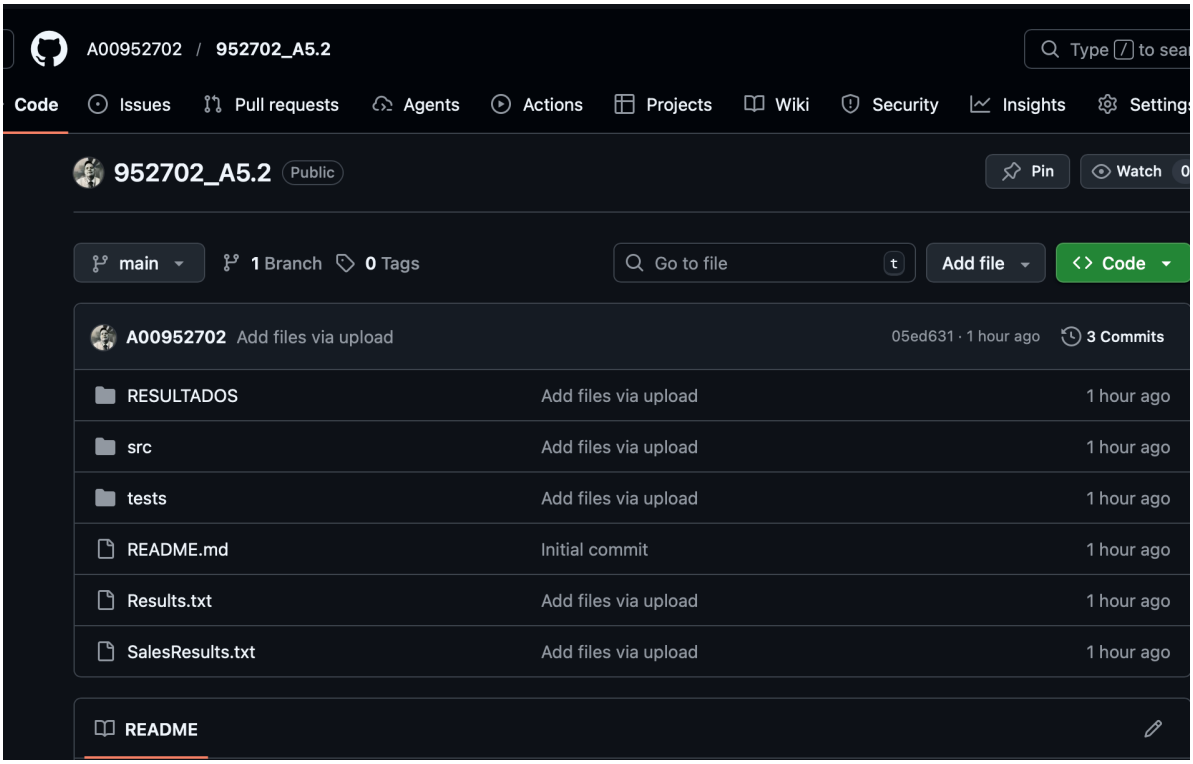
| Ejercicio de programación 2 y análisis estático | Tecnológico de Monterrey |
|---|---|
| Gerardo Solis Hernandez 952702 | Monterrey N.L. a 15 Febrero 2026 |

Repositorio: 952702_A5.2
Lenguaje: Python

## 1. Objetivo de la Actividad

El objetivo de esta actividad fue implementar un programa en Python que calcule el costo total de ventas a partir de un catálogo de productos y un archivo de ventas en formato JSON. Además, se aplicaron buenas prácticas de codificación siguiendo el estándar PEP 8 y se utilizó análisis estático con Flake8 y Pylint.

Todo esto se sube aun Repositorio




## 2. Descripción del Programa

El programa computeSales.py recibe dos archivos como parámetros:
• priceCatalogue.json – Contiene la lista de productos con su precio.
• salesRecord.json – Contiene los registros de ventas.

El programa realiza las siguientes acciones:
• Lee ambos archivos JSON.
• Valida los datos.
• Calcula el costo total de las ventas.
• Considera cantidades negativas (devoluciones).
• Muestra los resultados en consola.
• Genera un archivo SalesResults.txt con el resumen.
• Mide el tiempo de ejecución.

```
Please make sure you have the correct access rights
and the repository exists.
(.venv) (base) gerardosolis@MacBook-Pro-de-GERARDO-2 952702_A5.2 % python src/computeSales.py tests/TC1.ProductList.json tests
/TC1.Sales.json
```

## 3. Pruebas Realizadas

Se ejecutaron tres casos de prueba (TC1, TC2 y TC3) utilizando los archivos proporcionados en la carpeta tests. Se verificó que los resultados coincidieran con los valores esperados por el profesor.

En el caso TC2, fue necesario considerar las cantidades negativas en las ventas, ya que representan devoluciones o ajustes, y deben restarse del total.

```
(.venv) (base) gerardosolis@MacBook-Pro-de-GERARDO-2 952702_A5.2 % python src/computeSales.py tests/TC1.ProductList.json tests
/TC1.Sales.json
SALES SUMMARY
=============
VALID_SALES_ROWS: 46
- SALE_ID=1 | DATE=01/12/23 | PRODUCT='Rustic breakfast' | QTY=1 | UNIT=21.32 | LINE_TOTAL=21.32
- SALE_ID=1 | DATE=01/12/23 | PRODUCT='Sandwich with salad' | QTY=2 | UNIT=22.48 | LINE_TOTAL=44.96
- SALE_ID=1 | DATE=01/12/23 | PRODUCT='Raw legums' | QTY=1 | UNIT=17.11 | LINE_TOTAL=17.11
- SALE_ID=2 | DATE=01/12/23 | PRODUCT='Fresh stawberry' | QTY=1 | UNIT=28.59 | LINE_TOTAL=28.59
```

```
TOTAL_COST: 2481.86
ELAPSED_SECONDS: 0.000226
```

```
TOTAL_COST: 166568.23
ELAPSED_SECONDS: 0.000219
```

```
TOTAL_COST: 165235.37
ELAPSED_SECONDS: 0.000331
(.venv) (base) gerardosolis@MacBook-Pro-de-GERARDO-2 952702_A5.2 %
```

## 4. Cumplimiento con PEP 8

Se aseguró que el código cumpliera con el estándar PEP 8:
• Líneas menores a 79 caracteres.
• Indentación de 4 espacios.

• Eliminación de espacios en blanco innecesarios.
• Nombres de variables descriptivos.
• Uso correcto de docstrings.

## 5. Análisis Estático

Se utilizó Flake8 para detectar errores de estilo y complejidad ciclomática.
Se utilizó Pylint para identificar problemas de calidad del código.

Después de corregir todos los problemas detectados, el código alcanzó calificación perfecta (10.00/10) en Pylint.