



## **Maestría en Inteligente Artificial Aplicada (MNA)**

**Pruebas de software y aseguramiento de la Calidad**

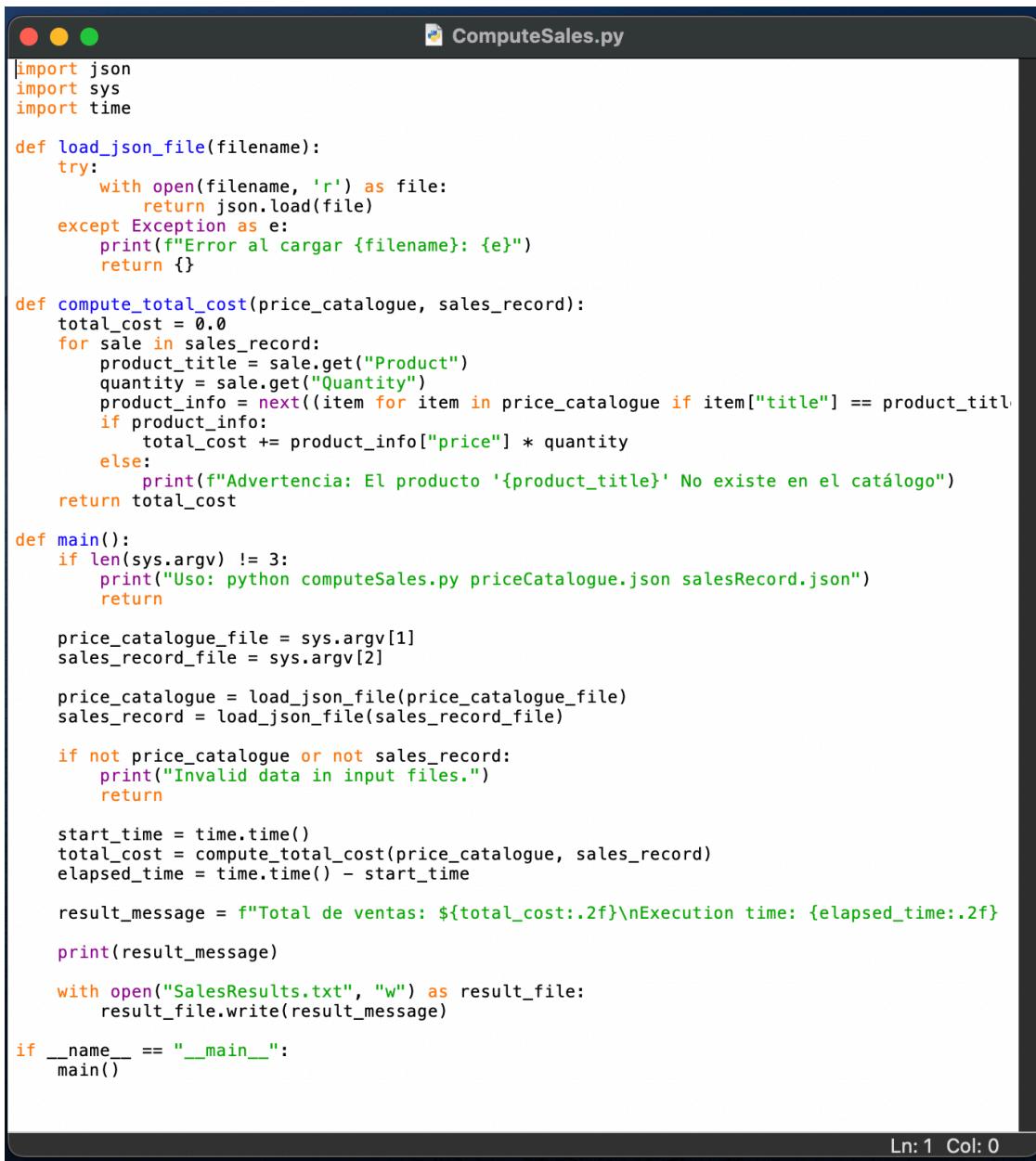
**Ejercicio de programación 2 y análisis estático.**

**Pérez del Razo, Mireya Isabel A01795608**

**Febrero 9, 2025.**

## Código ComputeSales.py

El código se capturó en Visual Studio Code y se ejecutó desde la línea de comandos en la opción “terminal” de la MAC.



```
import json
import sys
import time

def load_json_file(filename):
    try:
        with open(filename, 'r') as file:
            return json.load(file)
    except Exception as e:
        print(f"Error al cargar {filename}: {e}")
        return {}

def compute_total_cost(price_catalogue, sales_record):
    total_cost = 0.0
    for sale in sales_record:
        product_title = sale.get("Product")
        quantity = sale.get("Quantity")
        product_info = next((item for item in price_catalogue if item["title"] == product_title), None)
        if product_info:
            total_cost += product_info["price"] * quantity
        else:
            print(f"Advertencia: El producto '{product_title}' No existe en el catálogo")
    return total_cost

def main():
    if len(sys.argv) != 3:
        print("Uso: python computeSales.py priceCatalogue.json salesRecord.json")
        return

    price_catalogue_file = sys.argv[1]
    sales_record_file = sys.argv[2]

    price_catalogue = load_json_file(price_catalogue_file)
    sales_record = load_json_file(sales_record_file)

    if not price_catalogue or not sales_record:
        print("Invalid data in input files.")
        return

    start_time = time.time()
    total_cost = compute_total_cost(price_catalogue, sales_record)
    elapsed_time = time.time() - start_time

    result_message = f"Total de ventas: ${total_cost:.2f}\nExecution time: {elapsed_time:.2f}"
    print(result_message)

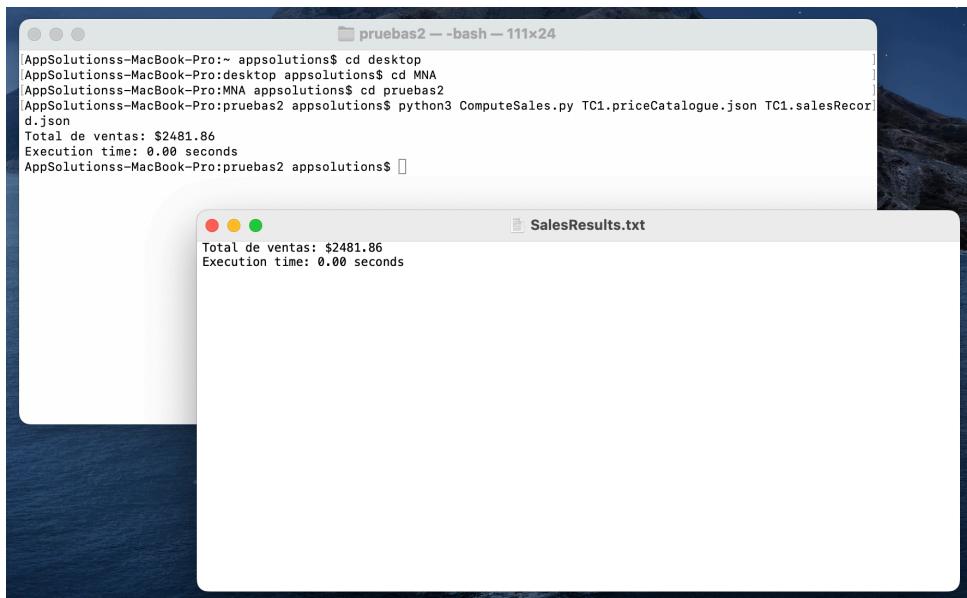
    with open("SalesResults.txt", "w") as result_file:
        result_file.write(result_message)

if __name__ == "__main__":
    main()
```

Ln: 1 Col: 0

## Pruebas de ejecución

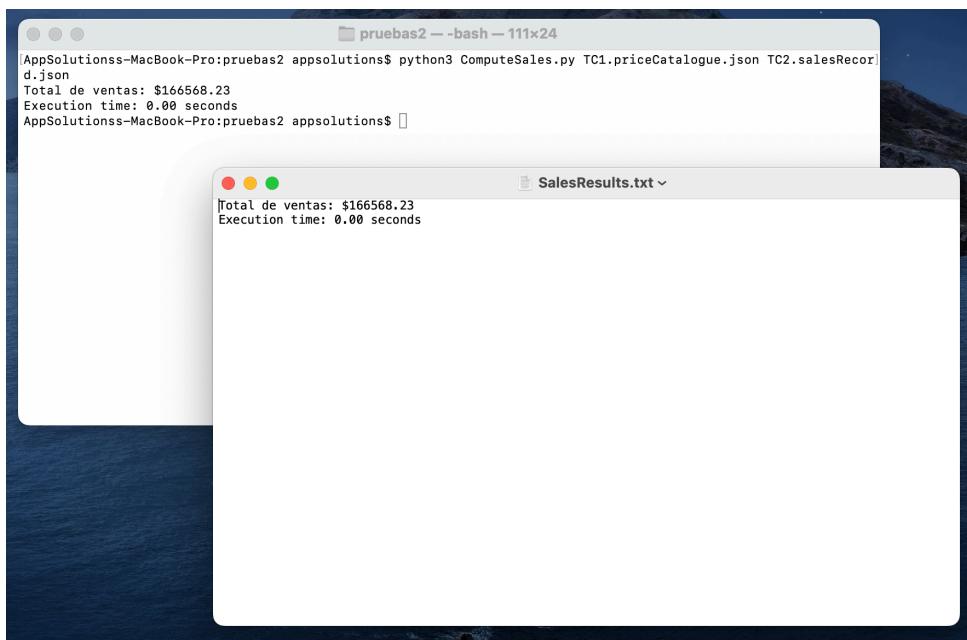
1. Ejecución del programa con el registro de ventas de TC1.



```
pruebas2 -- bash - 11x24
AppSolutionss-MacBook-Pro:~ appssolutions$ cd desktop
AppSolutionss-MacBook-Pro:desktop appssolutions$ cd MNA
AppSolutionss-MacBook-Pro:MNA appssolutions$ cd pruebas2
AppSolutionss-MacBook-Pro:pruebas2 appssolutions$ python3 ComputeSales.py TC1.priceCatalogue.json TC1.salesRecords.json
Total de ventas: $2481.86
Execution time: 0.00 seconds
AppSolutionss-MacBook-Pro:pruebas2 appssolutions$
```

```
SalesResults.txt
Total de ventas: $2481.86
Execution time: 0.00 seconds
```

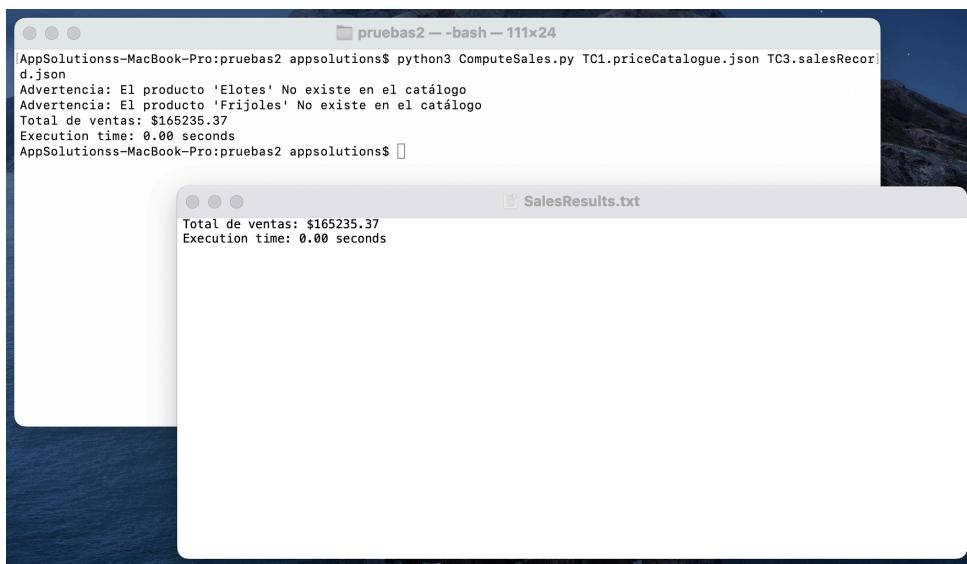
2. Ejecución del programa con el registro de ventas de TC2.



```
pruebas2 -- bash - 11x24
AppSolutionss-MacBook-Pro:pruebas2 appssolutions$ python3 ComputeSales.py TC1.priceCatalogue.json TC2.salesRecords.json
Total de ventas: $166568.23
Execution time: 0.00 seconds
AppSolutionss-MacBook-Pro:pruebas2 appssolutions$
```

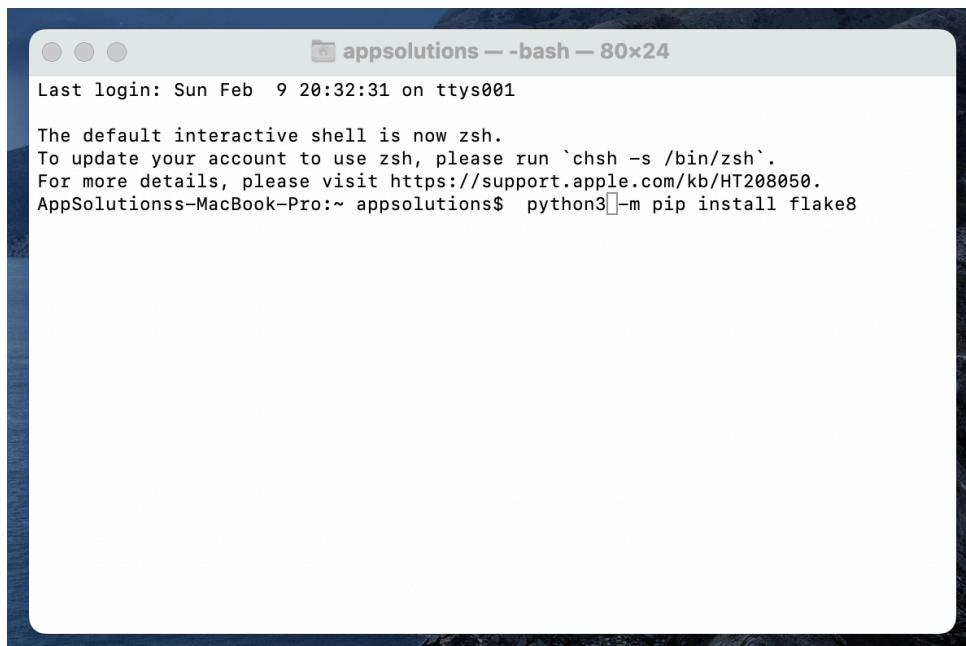
```
SalesResults.txt
Total de ventas: $166568.23
Execution time: 0.00 seconds
```

### 3. Ejecución del programa con el registro de ventas de TC3.



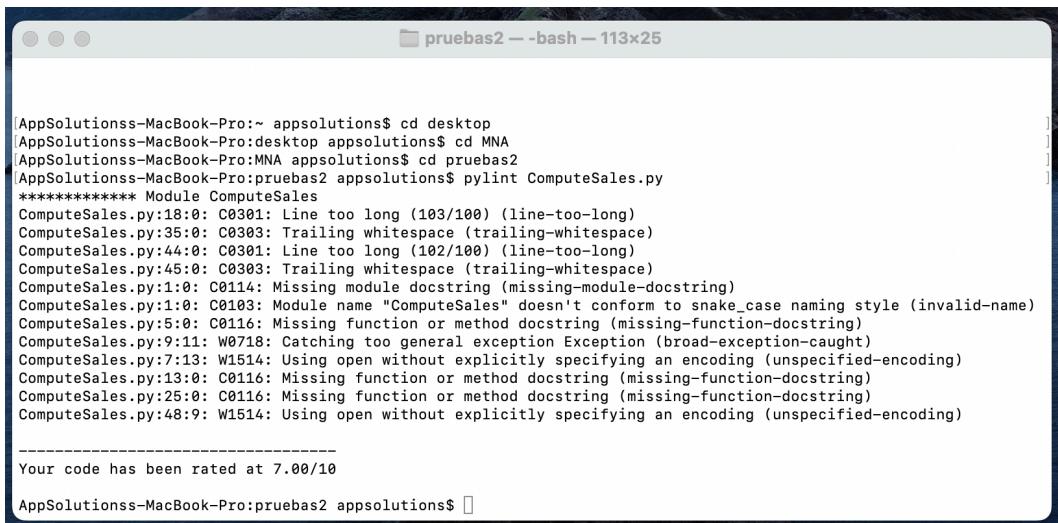
### Instalación de flake8, herramienta de análisis estático.

Se instala la herramienta flake8 con el comando: `python -m pip install flake8`



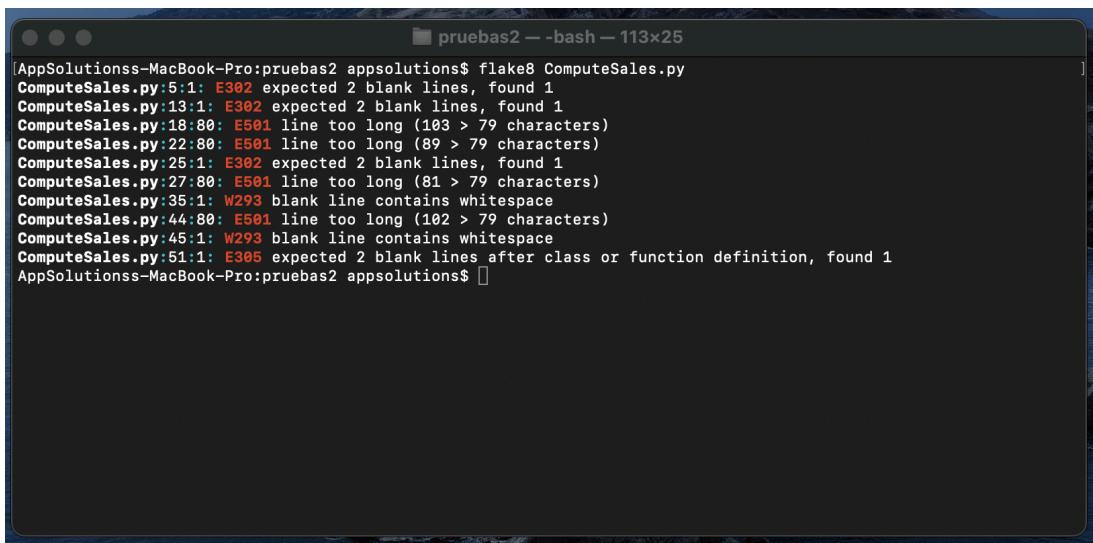
## Ejecución de pylint y flake8.

Se ejecutan ambas herramientas para llevar a cabo el análisis estático del código.



```
[AppSolutionss-MacBook-Pro:~ appssolutions$ cd desktop
[AppSolutionss-MacBook-Pro:desktop appssolutions$ cd MNA
[AppSolutionss-MacBook-Pro:MNA appssolutions$ cd pruebas2
[AppSolutionss-MacBook-Pro:pruebas2 appssolutions$ pylint ComputeSales.py
*****
Module ComputeSales
-----
ComputeSales.py:18:0: C0301: Line too long (103/100) (line-too-long)
ComputeSales.py:35:0: C0303: Trailing whitespace (trailing-whitespace)
ComputeSales.py:44:0: C0301: Line too long (102/100) (line-too-long)
ComputeSales.py:45:0: C0303: Trailing whitespace (trailing-whitespace)
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:5:0: C0116: Missing function or method docstring (missing-function-docstring)
ComputeSales.py:9:11: W0718: Catching too general exception Exception (broad-exception-caught)
ComputeSales.py:7:13: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
ComputeSales.py:13:0: C0116: Missing function or method docstring (missing-function-docstring)
ComputeSales.py:25:0: C0116: Missing function or method docstring (missing-function-docstring)
ComputeSales.py:48:0: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)

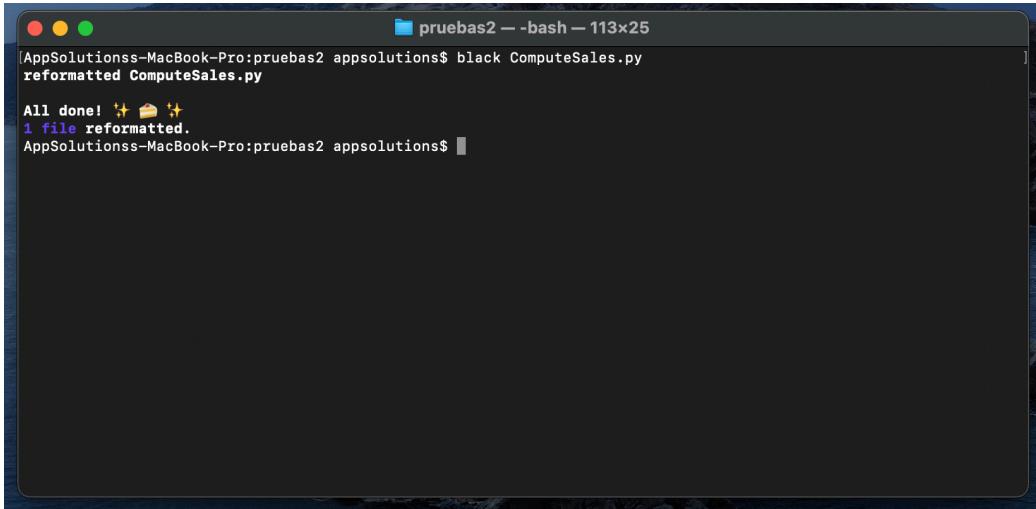
-----
Your code has been rated at 7.00/10
AppSolutionss-MacBook-Pro:pruebas2 appssolutions$ ]
```



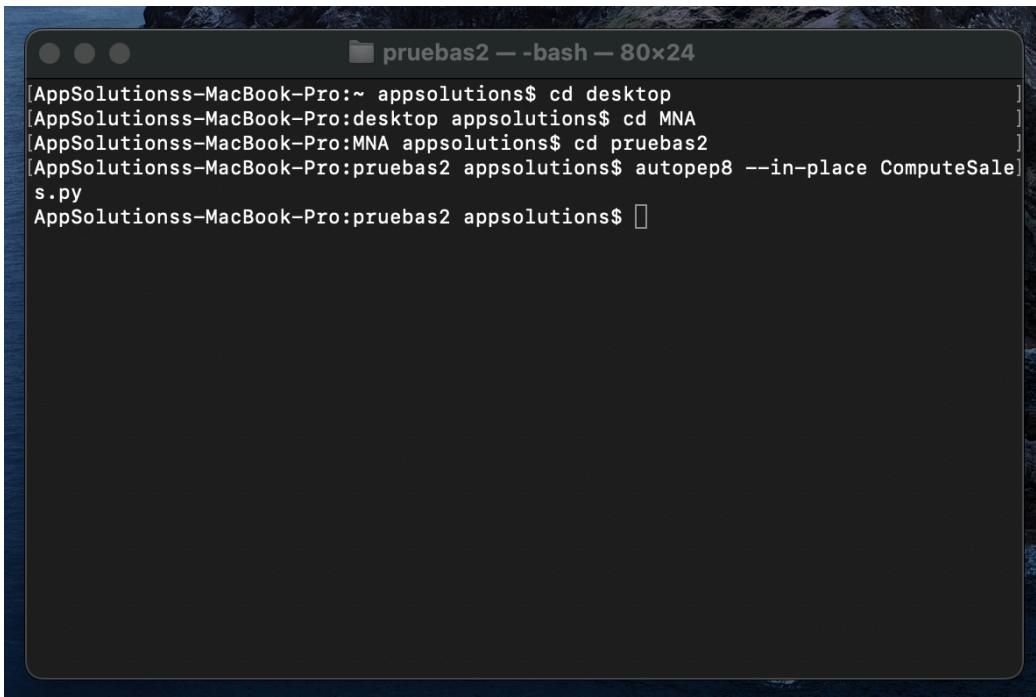
```
[AppSolutionss-MacBook-Pro:pruebas2 appssolutions$ flake8 ComputeSales.py
ComputeSales.py:5:1: E302 expected 2 blank lines, found 1
ComputeSales.py:13:1: E302 expected 2 blank lines, found 1
ComputeSales.py:18:80: E501 line too long (103 > 79 characters)
ComputeSales.py:22:80: E501 line too long (89 > 79 characters)
ComputeSales.py:25:1: E302 expected 2 blank lines, found 1
ComputeSales.py:27:80: E501 line too long (81 > 79 characters)
ComputeSales.py:35:1: W293 blank line contains whitespace
ComputeSales.py:44:80: E501 line too long (102 > 79 characters)
ComputeSales.py:45:1: W293 blank line contains whitespace
ComputeSales.py:51:1: E305 expected 2 blank lines after class or function definition, found 1
AppSolutionss-MacBook-Pro:pruebas2 appssolutions$ ]
```

## Ejecución de black y autopep8

Se ejecutan las herramientas de corrección de código black y autopep8.



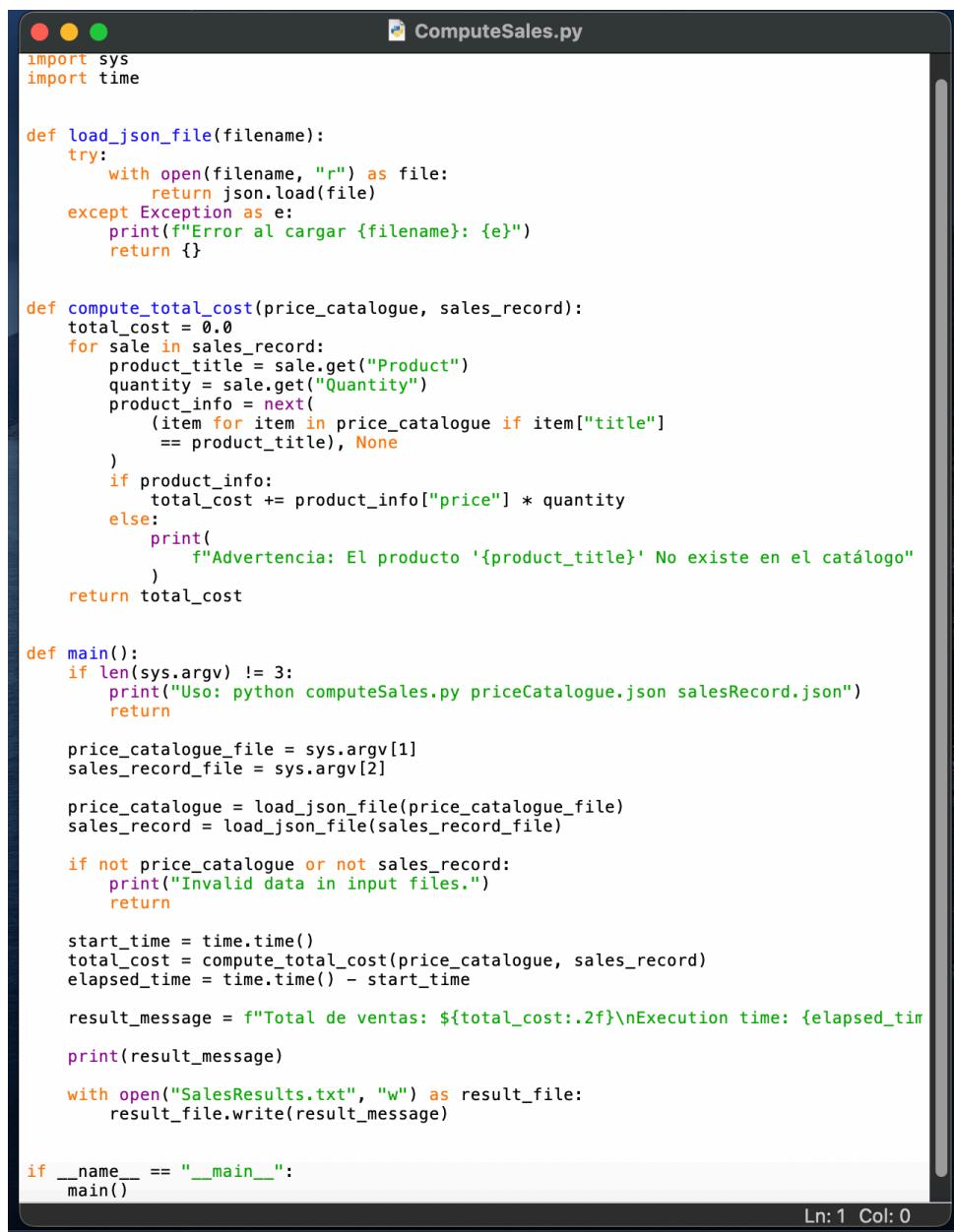
```
[AppSolutionss-MacBook-Pro:pruebas2 appsolutions$ black ComputeSales.py
reformatted ComputeSales.py
]
All done! ✨🍰✨
1 file reformatted.
AppSolutionss-MacBook-Pro:pruebas2 appsolutions$ ]
```



```
[AppSolutionss-MacBook-Pro:~ appsolutions$ cd desktop
[AppSolutionss-MacBook-Pro:desktop appsolutions$ cd MNA
[AppSolutionss-MacBook-Pro:MNA appsolutions$ cd pruebas2
[AppSolutionss-MacBook-Pro:pruebas2 appsolutions$ autopep8 --in-place ComputeSale
s.py
AppSolutionss-MacBook-Pro:pruebas2 appsolutions$ ]]
```

## Código final ComputeSales.py

Después de las correcciones pertinentes así quedó el código. Las pruebas de salida quedaron igual, corroborando con ello que efectivamente el análisis estático es independiente de las pruebas de ejecución.



```
ComputeSales.py
import sys
import time

def load_json_file(filename):
    try:
        with open(filename, "r") as file:
            return json.load(file)
    except Exception as e:
        print(f"Error al cargar {filename}: {e}")
        return {}

def compute_total_cost(price_catalogue, sales_record):
    total_cost = 0.0
    for sale in sales_record:
        product_title = sale.get("Product")
        quantity = sale.get("Quantity")
        product_info = next(
            (item for item in price_catalogue if item["title"] == product_title), None
        )
        if product_info:
            total_cost += product_info["price"] * quantity
        else:
            print(
                f"Advertencia: El producto '{product_title}' No existe en el catálogo"
            )
    return total_cost

def main():
    if len(sys.argv) != 3:
        print("Uso: python computeSales.py priceCatalogue.json salesRecord.json")
        return

    price_catalogue_file = sys.argv[1]
    sales_record_file = sys.argv[2]

    price_catalogue = load_json_file(price_catalogue_file)
    sales_record = load_json_file(sales_record_file)

    if not price_catalogue or not sales_record:
        print("Invalid data in input files.")
        return

    start_time = time.time()
    total_cost = compute_total_cost(price_catalogue, sales_record)
    elapsed_time = time.time() - start_time

    result_message = f"Total de ventas: ${total_cost:.2f}\nExecution time: {elapsed_time}"
    print(result_message)

    with open("SalesResults.txt", "w") as result_file:
        result_file.write(result_message)

if __name__ == "__main__":
    main()

Ln: 1 Col: 0
```