Programa

"""calculateSales.py

Automatically generated by Colab.

Original file is located at

https://colab.research.google.com/drive/1okpv7m4Y_MEyEsfSoB5GnSkEcN7SYIeE

Calcula el costo total de ventas tomando como base el cat\u00e1logo de precios.

Este programa toma dos archivos JSON como entrada:

1. priceCatalogue.json - Contiene información de los precios de productos.
2. salesRecord.json - Contiene un registro de las ventas realizadas.

El resultado se muestra en pantalla y se guarda en SalesResults.txt.

"""

```
import json

import sys

import time

import os


from google.colab import drive

drive.mount('/content/drive', force_remount=True)
```

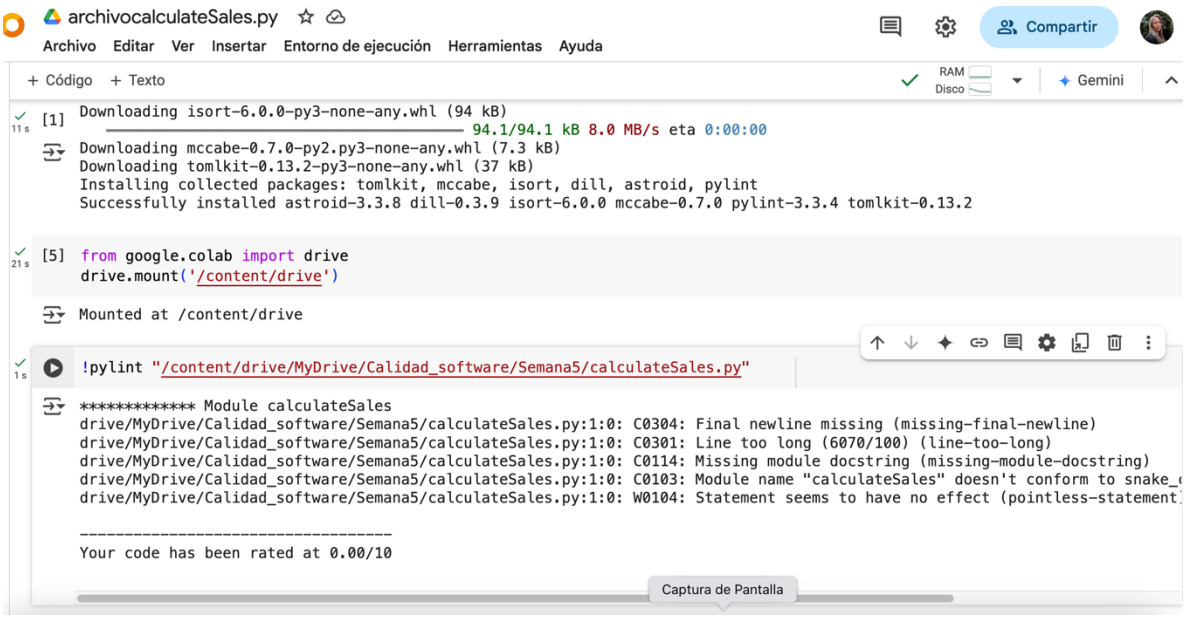
```
# Definir constantes para las rutas de archivos
PRICE_FILE = (
    "/content/drive/MyDrive/Calidad_software/Semana5/priceCatalogue.json"
)
SALES_FILE = (
    "/content/drive/MyDrive/Calidad_software/Semana5/salesRecord.json"
)
RESULT_FILE = (
    "/content/drive/MyDrive/Calidad_software/Semana5/SalesResults.txt"
)


def load_json(file_path):
    """Carga un archivo JSON y maneja errores."""
    try:
        with open(file_path, "r", encoding="utf-8") as file:
            return json.load(file)
    except FileNotFoundError:
        print(f"Error: No se encontr\u00f3 el archivo {file_path}.")
    except json.JSONDecodeError:
        print(f"Error: {file_path} no formato JSON v\u00e1lido.")
    return None


def calculate_total_sales(price_catalogue, sales_record):
    """Calcula costo total de ventas basado en el cat\u00e1logo precios."""
    total_sales = 0.0
    detailed_sales = []
    errors = []
```

```
for sale in sales_record:
    product_id = sale.get("product_id")
    quantity = sale.get("quantity")

    if product_id not in price_catalogue:
        errors.append(
            f"Producto no encontrado en cat\u00e1logo: {product_id}"
        )
        continue

    if not isinstance(quantity, (int, float)) or quantity <= 0:
        errors.append(
            f"Cantidad inv\u00e1lida producto {product_id}: {quantity}"
        )
        continue

    price = price_catalogue[product_id]
    total_cost = price * quantity
    total_sales += total_cost
    detailed_sales.append(
        f"Producto: {product_id}, Cantidad: {quantity}, "
        f"Precio unitario: ${price:.2f}, Costo total: ${total_cost:.2f}"
    )

return total_sales, detailed_sales, errors
```

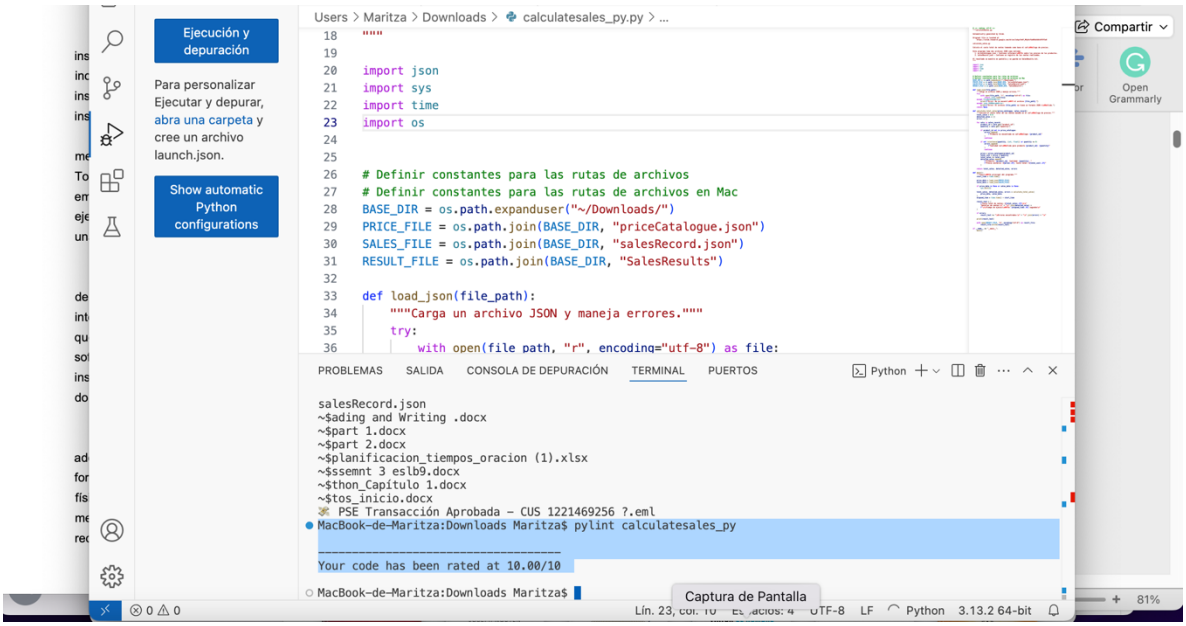
```
def main():
    """Funci\u00f3n principal del programa."""
    start_time = time.time()

    price_data = load_json(PRICE_FILE)
    sales_data = load_json(SALES_FILE)

    if price_data is None or sales_data is None:
        sys.exit(1)

    total_sales, detailed_sales, errors = calculate_total_sales(
        price_data, sales_data
    )
    elapsed_time = time.time() - start_time

    result_text = (
        f"Costo total de ventas: ${total_sales:.2f}\n\n"
        "Detalles de ventas:\n" + "\n".join(detailed_sales) +
        f"\n\nTiempo de ejecuci\u00f3n: {elapsed_time:.4f} segundos\n"
    )

    if errors:
        result_text += "\nErrores encontrados:\n" + "\n".join(errors) + "\n"

    print(result_text)

    with open(RESULT_FILE, "w", encoding="utf-8") as result_file:
```

```
    result_file.write(result_text)


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

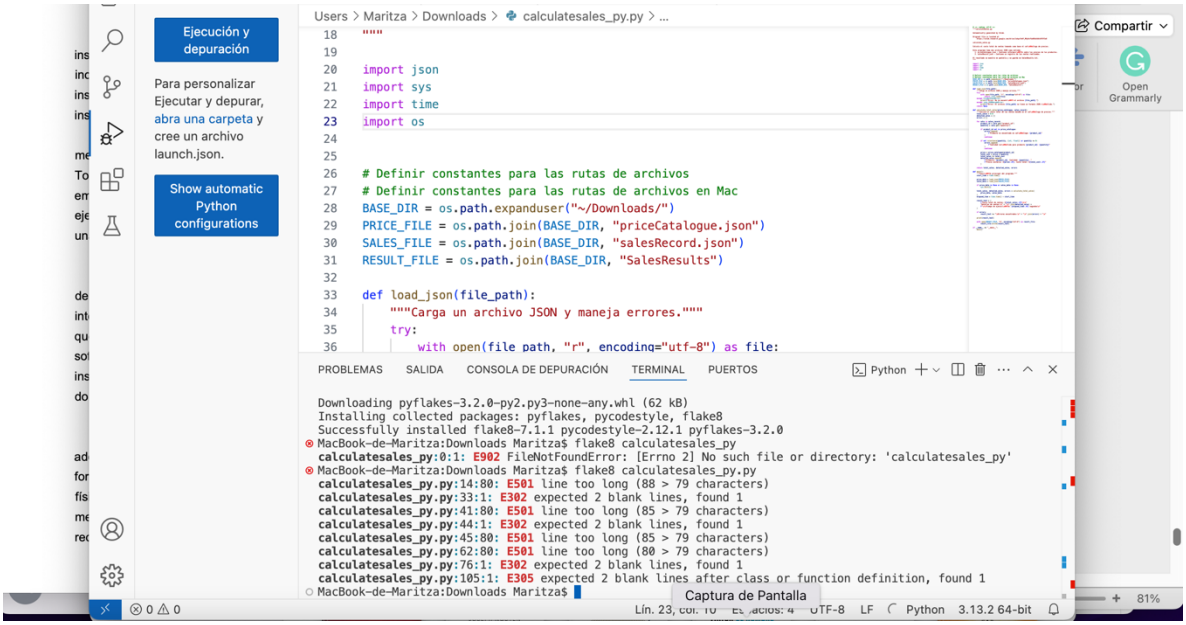
Al Ejecutar el comando Pylint para las validaciones dinámicas se encontró:



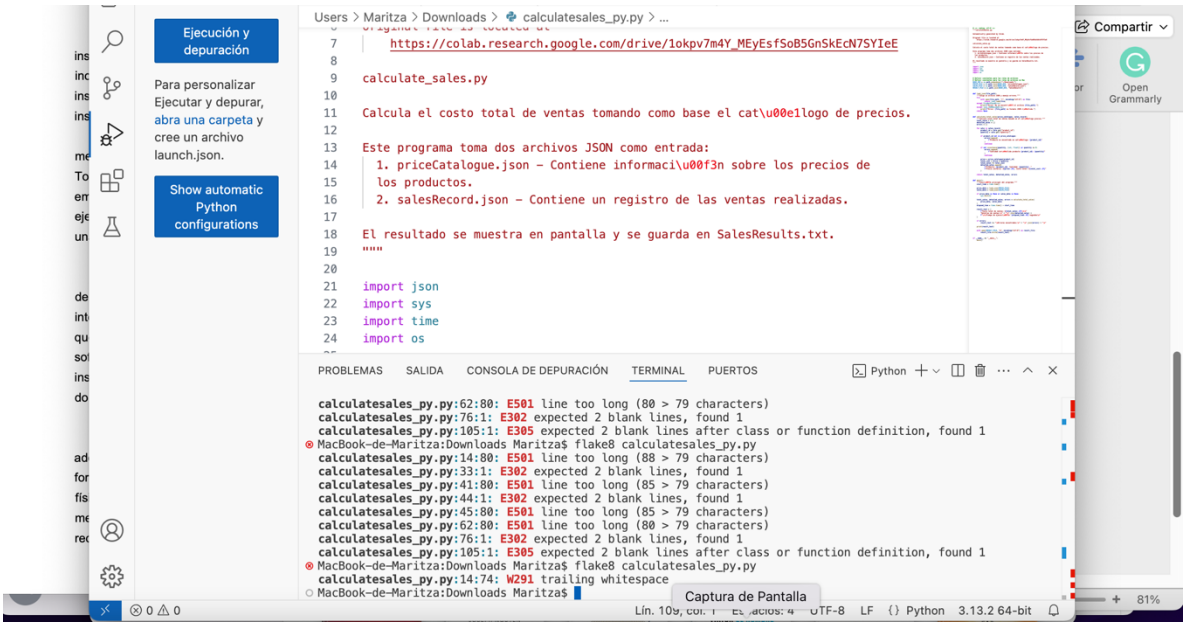
En colab no me ejecuta correctamente, lo realice en Visual y me salió todo ok



Al ejecutar Flake8 salio por visual:



Se ajustaron cosas se subsanaron, pero sale otro error



Ultima ejecución eliminando espacios y reduciendo el texto