

5.2 Ejercicio de programación 2 y análisis estático

Juan Felipe Toro Salgado

a01794247

Descripción General

Este script, `compute_sales.py`, procesa transacciones de ventas basadas en un catálogo de productos proporcionado. Calcula el costo total de las ventas, registra errores y añade los resultados a un archivo. Además, realiza verificaciones de calidad de código usando `pylint` y `flake8`.

Características

- Carga un catálogo de productos desde un archivo JSON.
- Procesa un registro de ventas desde otro archivo JSON.
- Calcula el costo total de las ventas.
- Registra errores para productos faltantes o inválidos.
- Agrega resultados a `SalesResults.txt` con una marca de tiempo.
- Ejecuta `pylint` y `flake8` para verificar la calidad del código.

Instalación y Configuración

1. Asegúrate de tener Python instalado.
2. Instala las dependencias necesarias:

```
pip install flake8 pylint
```

3. Coloca `priceCatalogue.json` y `salesRecord.json` en el directorio de trabajo.

Uso

Ejecuta el script desde la línea de comandos:

```
python compute_sales.py priceCatalogue.json salesRecord.json
```

Salida Esperada

Los resultados se agregarán al archivo `SalesResults.txt` con el siguiente formato:

```
=====
Informe de Ventas - Ejecutado en YYYY-MM-DD HH:MM:SS
=====
Costo Total de Ventas: $XXXX.XX
```

Tiempo Transcurrido: X.XXXX segundos
Errores:
- Producto inválido: ejemplo_producto

Manejo de Errores

- Si el producto en el archivo de ventas no existe en el catálogo, se registrará un error.
- Si un archivo JSON está ausente o mal formateado, el script mostrará un error y se detendrá.

Verificación de Calidad del Código

El script ejecuta `pylint` y `flake8` automáticamente después de la ejecución. Los resultados se almacenan en:

- `pylint_results.txt`
- `flake8_results.txt`