Instituto Tecnológico y de Estudios Superiores de Monterrey

Pruebas de software y aseguramiento de la calidad

Actividad 5.2 Ejercicio de programación 2 y análisis estático

Susana Pérez Carranza

Matrícula: A01796151

El siguiente reporte contiene un análisis de los resultados obtenidos en la ejecución del programa realizado para el ejercicio planteado.

Ejercicio Compute sales.

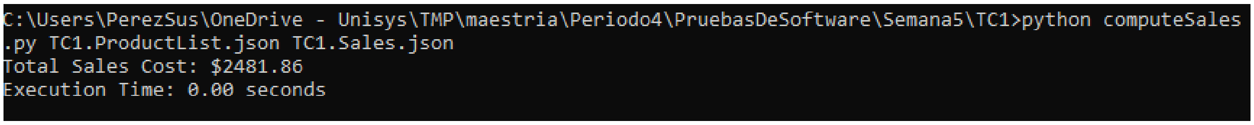
Se plantean los siguientes requisitos:

Req1. The program shall be invoked from a command line. The program shall receive two files as parameters. The first file will contain information in a JSON format about a catalogue of prices of products. The second file will contain a record for all sales in a company.
Req 2. The program shall compute the total cost for all sales included in the second JSON archive. The results shall be print on a screen and on a file named SalesResults.txt. The total cost should include all items in the sale considering the cost for every item in the first file. The output must be human readable, so make it easy to read for the user.
Req 3. The program shall include the mechanism to handle invalid data in the file. Errors should be displayed in the console and the execution must continue.
Req 4. The name of the program shall be
computeSales.py
Req 5. The minimum format to invoke the program shall be as follows:
python computeSales.py priceCatalogue.json salesRecord.json
Req 6. The program shall manage files having from hundreds of items to thousands of items.
Req 7. The program should include at the end of the execution the time elapsed for the execution and calculus of the data. This number shall be included in the results file and on the screen.
Req 8. Be compliant with PEP8.

Se implementa el programa siguiendo la guía del standard PEP8.

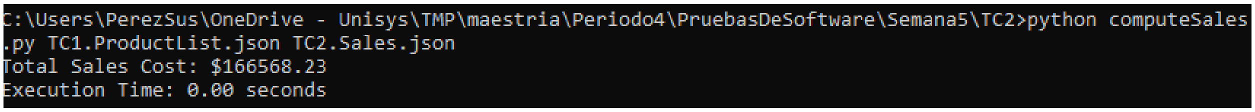
Resultados iniciales

Test Case 1:



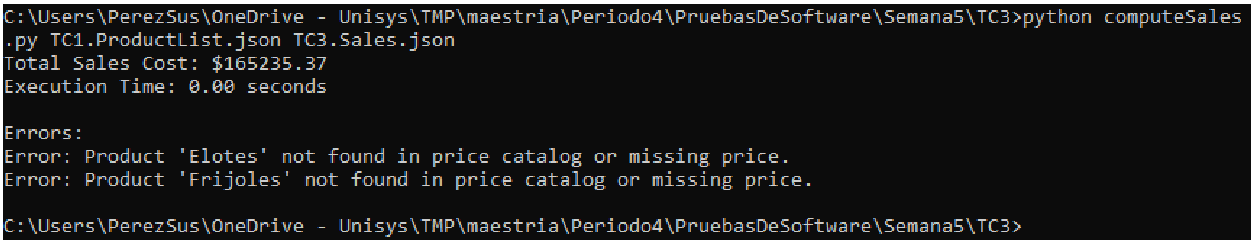
```
C:\Users\PerezSus\OneDrive - Unisys\TMP\maestria\Periodo4\PruebasDeSoftware\Semana5\TC1>python computeSales
.py TC1.ProductList.json TC1.Sales.json
Total Sales Cost: $2481.86
Execution Time: 0.00 seconds
```

Test Case 2:



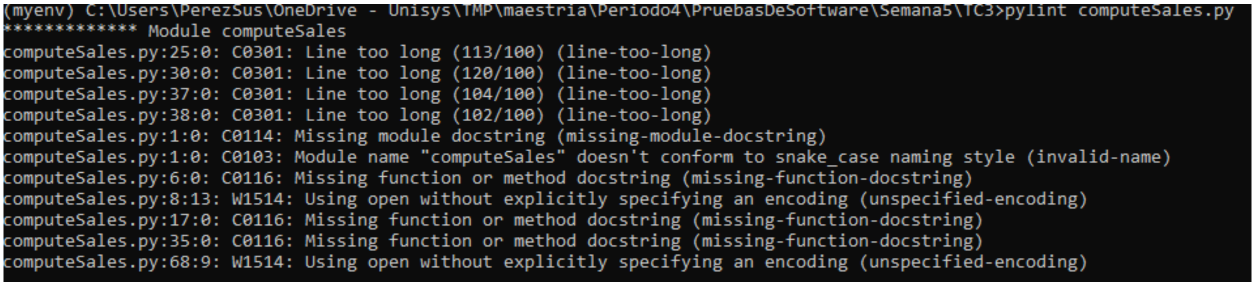
```
C:\Users\PerezSus\OneDrive - Unisys\TMP\maestria\Periodo4\PruebasDeSoftware\Semana5\TC2>python computeSales
.py TC1.ProductList.json TC2.Sales.json
Total Sales Cost: $166568.23
Execution Time: 0.00 seconds
```

Test Case 3:



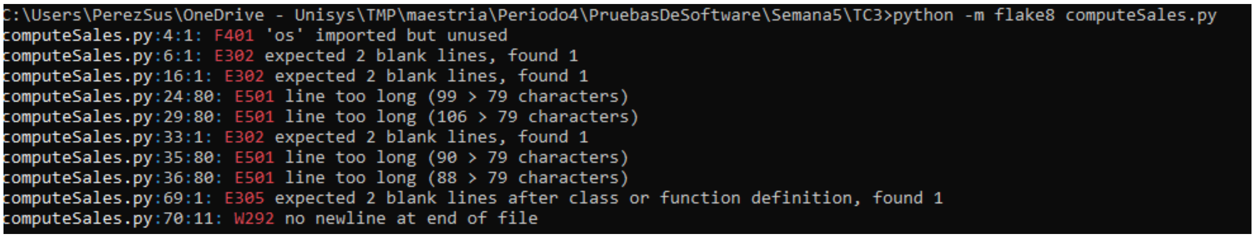
```
C:\Users\PerezSus\OneDrive - Unisys\TMP\maestria\Periodo4\PruebasDeSoftware\Semana5\TC3>python computeSales
.py TC1.ProductList.json TC3.Sales.json
Total Sales Cost: $165235.37
Execution Time: 0.00 seconds

Errors:
Error: Product 'Elotes' not found in price catalog or missing price.
Error: Product 'Frijoles' not found in price catalog or missing price.

C:\Users\PerezSus\OneDrive - Unisys\TMP\maestria\Periodo4\PruebasDeSoftware\Semana5\TC3>
```

Resultado inicial al correr pylint:



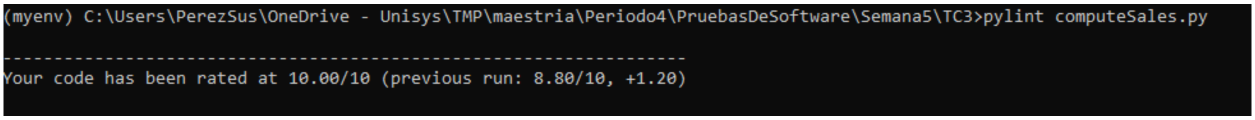
```
(myenv) C:\Users\PerezSus\OneDrive - Unisys\TMP\maestria\Periodo4\PruebasDeSoftware\Semana5\TC3>pylint computeSales.py
************* Module computeSales
computeSales.py:25:0: C0301: Line too long (113/100) (line-too-long)
computeSales.py:30:0: C0301: Line too long (120/100) (line-too-long)
computeSales.py:37:0: C0301: Line too long (104/100) (line-too-long)
computeSales.py:38:0: C0301: Line too long (102/100) (line-too-long)
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:6:0: C0116: Missing function or method docstring (missing-function-docstring)
computeSales.py:8:13: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
computeSales.py:17:0: C0116: Missing function or method docstring (missing-function-docstring)
computeSales.py:35:0: C0116: Missing function or method docstring (missing-function-docstring)
computeSales.py:68:9: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
```

Resultado inicial al correr flake8:



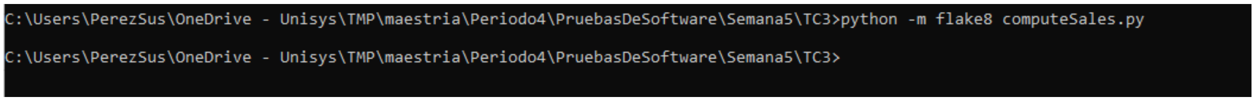
```
C:\Users\PerezSus\OneDrive - Unisys\TMP\maestria\Periodo4\PruebasDeSoftware\Semana5\TC3>python -m flake8 computeSales.py
computeSales.py:4:1: F401 'os' imported but unused
computeSales.py:6:1: E302 expected 2 blank lines, found 1
computeSales.py:16:1: E302 expected 2 blank lines, found 1
computeSales.py:24:80: E501 line too long (99 > 79 characters)
computeSales.py:29:80: E501 line too long (106 > 79 characters)
computeSales.py:33:1: E302 expected 2 blank lines, found 1
computeSales.py:35:80: E501 line too long (90 > 79 characters)
computeSales.py:36:80: E501 line too long (88 > 79 characters)
computeSales.py:69:1: E305 expected 2 blank lines after class or function definition, found 1
computeSales.py:70:11: W292 no newline at end of file
```

Resultados al aplicar las sugerencias de pylint:



```
(myenv) C:\Users\PerezSus\OneDrive - Unisys\TMP\maestria\Periodo4\PruebasDeSoftware\Semana5\TC3>pylint computeSales.py

-------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 8.80/10, +1.20)
```

Resultados al aplicar las sugerencias de flake8:



```
C:\Users\PerezSus\OneDrive - Unisys\TMP\maestria\Periodo4\PruebasDeSoftware\Semana5\TC3>python -m flake8 computeSales.py

C:\Users\PerezSus\OneDrive - Unisys\TMP\maestria\Periodo4\PruebasDeSoftware\Semana5\TC3>
```

Referencias:

1. Van Rossum, G., Warsaw, B. & Coghlan, Alyssa (2013). *PEP 8 – Style Guide for Python Cod. Convención de codificación de Python - PEP8*

2. Python Code Quality Authority. (n.d.). *Flake8: Your tool for style guide enforcement*. Flake8 7.1.0 documentation. https://flake8.pycqa.org/en/latest/