Pruebas de software y aseguramiento de la calidad
Erick de Jesus Hernandez Cerecedo
A01066428

México, 11 de febrero de 2024

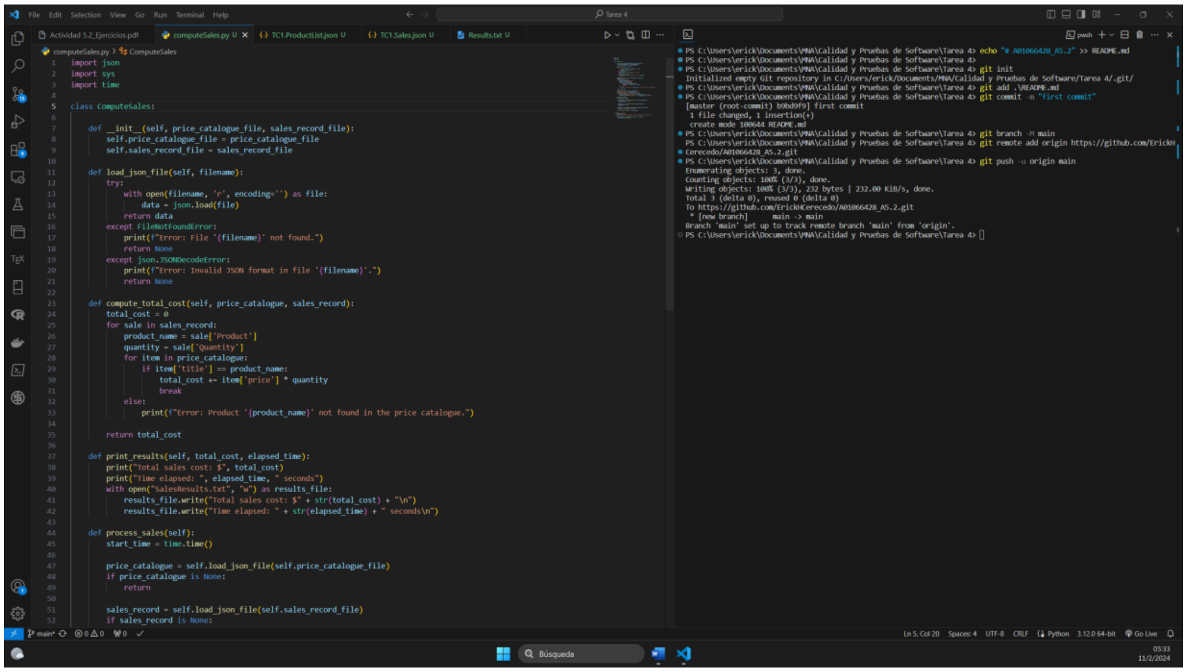
Actividad 5.2 | Ejercicio de programación 2

Enlace al repositorio de GitHub: https://github.com/ErickHCerecedo/A01066428_A5.2

El siguiente archivo contiene la solución implementada en Python al ejercicio de programación 2, en el que se reporta el proceso de creación del ejercicio desde la ejecución de revisión de gramática y estilo usando PyLint, análisis de errores empleando Flake y por último la ejecución de pruebas unitarias.

Problem 1: Compute Sales	
Description	Requirement 1. 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. Requirement 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. Requirement 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. Requirement 4. The name of the program shall be computeSales.py Requirement 5. The minimum format to invoke the program shall be as follows: python computeSales.py priceCatalogue.json salesRecord.json Requirement 6. The program shall manage files having from hundreds of items to thousands of items. Requirement 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. Requirement 8. Be compliant with PEP8.

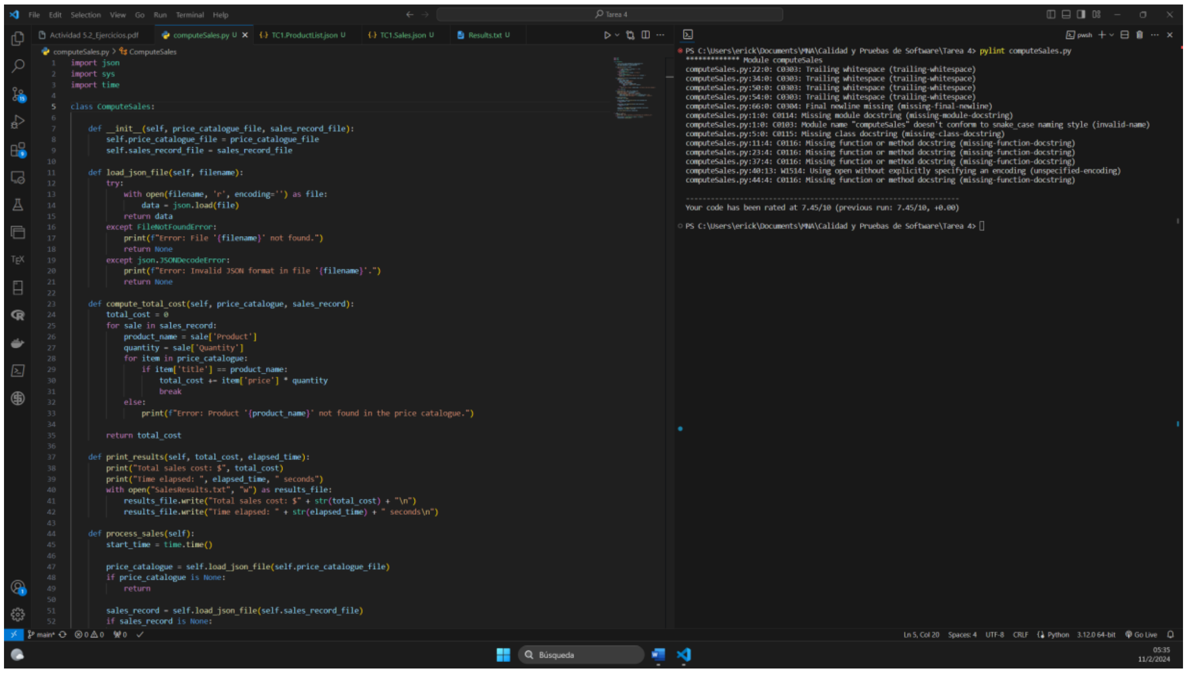
1. Implementación inicial del código:



```python
1  import json
2  import sys
3  import time
4
5  class ComputeSales:
6
7      def __init__(self, price_catalogue_file, sales_record_file):
8          self.price_catalogue_file = price_catalogue_file
9          self.sales_record_file = sales_record_file
10
11     def load_json_file(self, filename):
12         try:
13             with open(filename, 'r', encoding='') as file:
14                 data = json.load(file)
15             return data
16         except FileNotFoundError:
17             print(f"Error: File '{filename}' not found.")
18             return None
19         except json.JSONDecodeError:
20             print(f"Error: Invalid JSON format in file '{filename}'.")
21             return None
22
23     def compute_total_cost(self, price_catalogue, sales_record):
24         total_cost = 0
25         for sale in sales_record:
26             product_name = sale['Product']
27             quantity = sale['Quantity']
28             for item in price_catalogue:
29                 if item['title'] == product_name:
30                     total_cost += item['price'] * quantity
31                     break
32             else:
33                 print(f"Error: Product '{product_name}' not found in the price catalogue.")
34
35         return total_cost
36
37     def print_results(self, total_cost, elapsed_time):
38         print("Total sales cost: $", total_cost)
39         print("Time elapsed: ", elapsed_time, " seconds")
40         with open("SalesResults.txt", "w") as results_file:
41             results_file.write("Total sales cost: $" + str(total_cost) + "\n")
42             results_file.write("Time elapsed: " + str(elapsed_time) + " seconds\n")
43
44     def process_sales(self):
45         start_time = time.time()
46
47         price_catalogue = self.load_json_file(self.price_catalogue_file)
48         if price_catalogue is None:
49             return
50
51         sales_record = self.load_json_file(self.sales_record_file)
52         if sales_record is None:
```

```
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> echo "# A01066428_A5.2" >> README.md
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4>
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> git init
Initialized empty Git repository in C:/Users/erick/Documents/MNA/Calidad y Pruebas de Software/Tarea 4/.git/
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> git add .\README.md
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> git commit -m "first commit"
[master (root-commit) b9bd9f9] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> git branch -M main
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> git remote add origin https://github.com/ErickH
Cerecedo/A01066428_A5.2.git
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> git push -u origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 232 bytes | 232.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/ErickHCerecedo/A01066428_A5.2.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> 
```

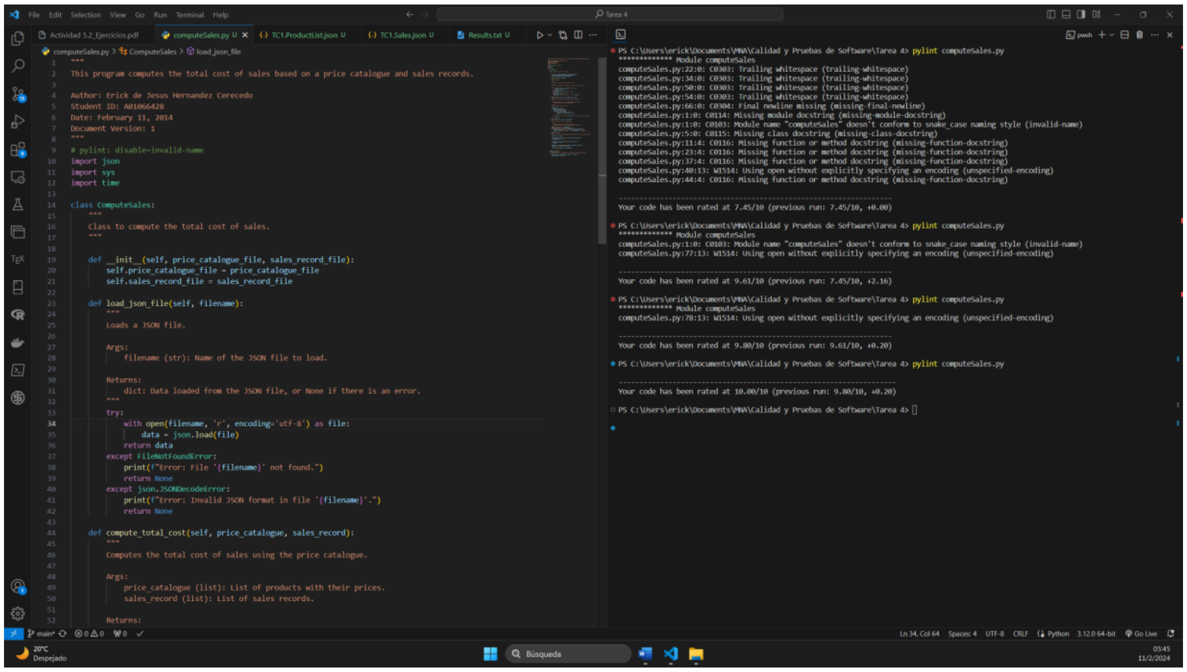
2. Análisis de errores de PyLint (PEP8):



```python
1  import json
2  import sys
3  import time
4
5  class ComputeSales:
6
7      def __init__(self, price_catalogue_file, sales_record_file):
8          self.price_catalogue_file = price_catalogue_file
9          self.sales_record_file = sales_record_file
10
11     def load_json_file(self, filename):
12         try:
13             with open(filename, 'r', encoding='') as file:
14                 data = json.load(file)
15             return data
16         except FileNotFoundError:
17             print(f"Error: File '{filename}' not found.")
18             return None
19         except json.JSONDecodeError:
20             print(f"Error: Invalid JSON format in file '{filename}'.")
21             return None
22
23     def compute_total_cost(self, price_catalogue, sales_record):
24         total_cost = 0
25         for sale in sales_record:
26             product_name = sale['Product']
27             quantity = sale['Quantity']
28             for item in price_catalogue:
29                 if item['title'] == product_name:
30                     total_cost += item['price'] * quantity
31                     break
32             else:
33                 print(f"Error: Product '{product_name}' not found in the price catalogue.")
34
35         return total_cost
36
37     def print_results(self, total_cost, elapsed_time):
38         print("Total sales cost: $", total_cost)
39         print("Time elapsed: ", elapsed_time, " seconds")
40         with open("SalesResults.txt", "w") as results_file:
41             results_file.write("Total sales cost: $" + str(total_cost) + "\n")
42             results_file.write("Time elapsed: " + str(elapsed_time) + " seconds\n")
43
44     def process_sales(self):
45         start_time = time.time()
46
47         price_catalogue = self.load_json_file(self.price_catalogue_file)
48         if price_catalogue is None:
49             return
50
51         sales_record = self.load_json_file(self.sales_record_file)
52         if sales_record is None:
```

```
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> pylint computeSales.py
************* Module computeSales
computeSales.py:22:0: C0303: Trailing whitespace (trailing-whitespace)
computeSales.py:34:0: C0303: Trailing whitespace (trailing-whitespace)
computeSales.py:50:0: C0303: Trailing whitespace (trailing-whitespace)
computeSales.py:54:0: C0303: Trailing whitespace (trailing-whitespace)
computeSales.py:66:0: C0304: Final newline missing (missing-final-newline)
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: C0115: Missing class docstring (missing-class-docstring)
computeSales.py:11:4: C0116: Missing function or method docstring (missing-function-docstring)
computeSales.py:23:4: C0116: Missing function or method docstring (missing-function-docstring)
computeSales.py:37:4: C0116: Missing function or method docstring (missing-function-docstring)
computeSales.py:40:13: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
computeSales.py:44:4: C0116: Missing function or method docstring (missing-function-docstring)

------------------------------------------------------------------
Your code has been rated at 7.45/10 (previous run: 7.45/10, +0.00)

PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> 
```

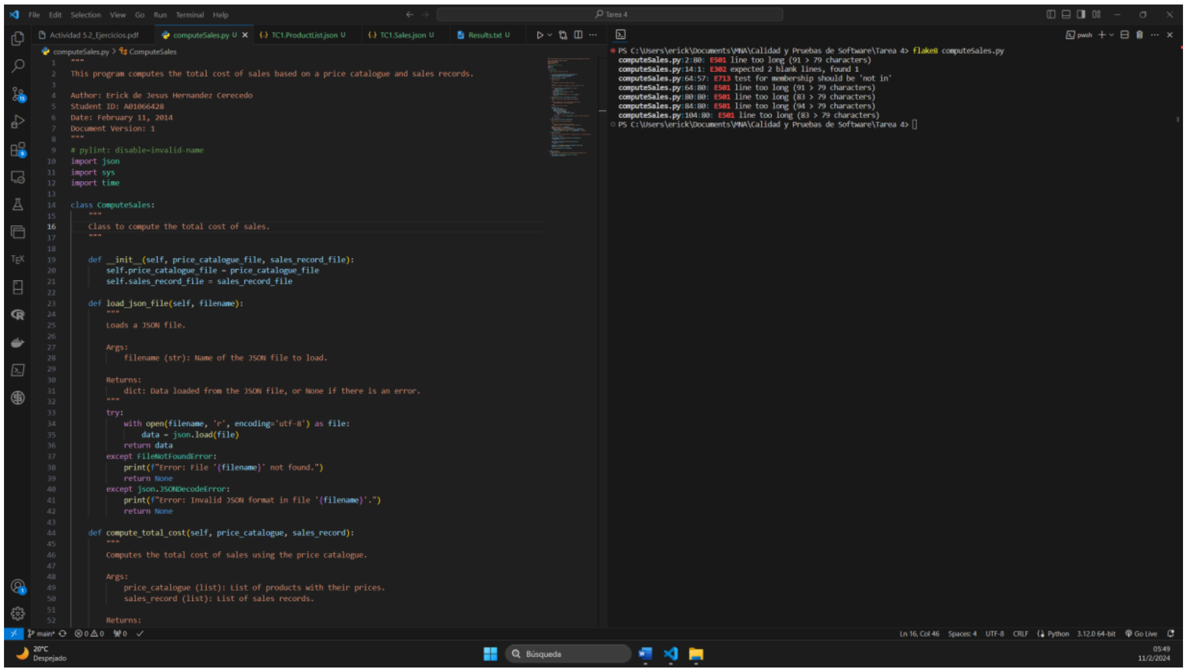
Corrección de errores:



```python
"""
This program computes the total cost of sales based on a price catalogue and sales records.

Author: Erick de Jesus Hernandez Cerecedo
Student ID: A01066428
Date: February 11, 2014
Document Version: 1
"""
# pylint: disable=invalid-name
import json
import sys
import time

class ComputeSales:
    """
    Class to compute the total cost of sales.
    """

    def __init__(self, price_catalogue_file, sales_record_file):
        self.price_catalogue_file = price_catalogue_file
        self.sales_record_file = sales_record_file

    def load_json_file(self, filename):
        """
        Loads a JSON file.

        Args:
            filename (str): Name of the JSON file to load.

        Returns:
            dict: Data loaded from the JSON file, or None if there is an error.
        """
        try:
            with open(filename, 'r', encoding='utf-8') as file:
                data = json.load(file)
            return data
        except FileNotFoundError:
            print(f"Error: File '{filename}' not found.")
            return None
        except json.JSONDecodeError:
            print(f"Error: Invalid JSON format in file '{filename}'.")
            return None

    def compute_total_cost(self, price_catalogue, sales_record):
        """
        Computes the total cost of sales using the price catalogue.

        Args:
            price_catalogue (list): List of products with their prices.
            sales_record (list): List of sales records.

        Returns:
```

```
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> pylint computeSales.py
************* Module computeSales
computeSales.py:22:0: C0303: Trailing whitespace (trailing-whitespace)
computeSales.py:34:0: C0303: Trailing whitespace (trailing-whitespace)
computeSales.py:50:0: C0303: Trailing whitespace (trailing-whitespace)
computeSales.py:54:0: C0303: Trailing whitespace (trailing-whitespace)
computeSales.py:66:0: C0304: Final newline missing (missing-final-newline)
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: C0115: Missing class docstring (missing-class-docstring)
computeSales.py:11:4: C0116: Missing function or method docstring (missing-function-docstring)
computeSales.py:23:4: C0116: Missing function or method docstring (missing-function-docstring)
computeSales.py:37:4: C0116: Missing function or method docstring (missing-function-docstring)
computeSales.py:40:13: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)
computeSales.py:44:4: C0116: Missing function or method docstring (missing-function-docstring)

-------------------------------------------------------------------
Your code has been rated at 7.45/10 (previous run: 7.45/10, +0.00)

PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> pylint computeSales.py
************* Module computeSales
computeSales.py:1:0: C0103: Module name "computeSales" doesn't conform to snake_case naming style (invalid-name)
computeSales.py:77:13: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)

-------------------------------------------------------------------
Your code has been rated at 9.61/10 (previous run: 7.45/10, +2.16)

PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> pylint computeSales.py
************* Module computeSales
computeSales.py:78:13: W1514: Using open without explicitly specifying an encoding (unspecified-encoding)

-------------------------------------------------------------------
Your code has been rated at 9.80/10 (previous run: 9.61/10, +0.20)

PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> pylint computeSales.py

--------------------------------------------------------------------
Your code has been rated at 10.00/10 (previous run: 9.80/10, +0.20)

PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> 
```

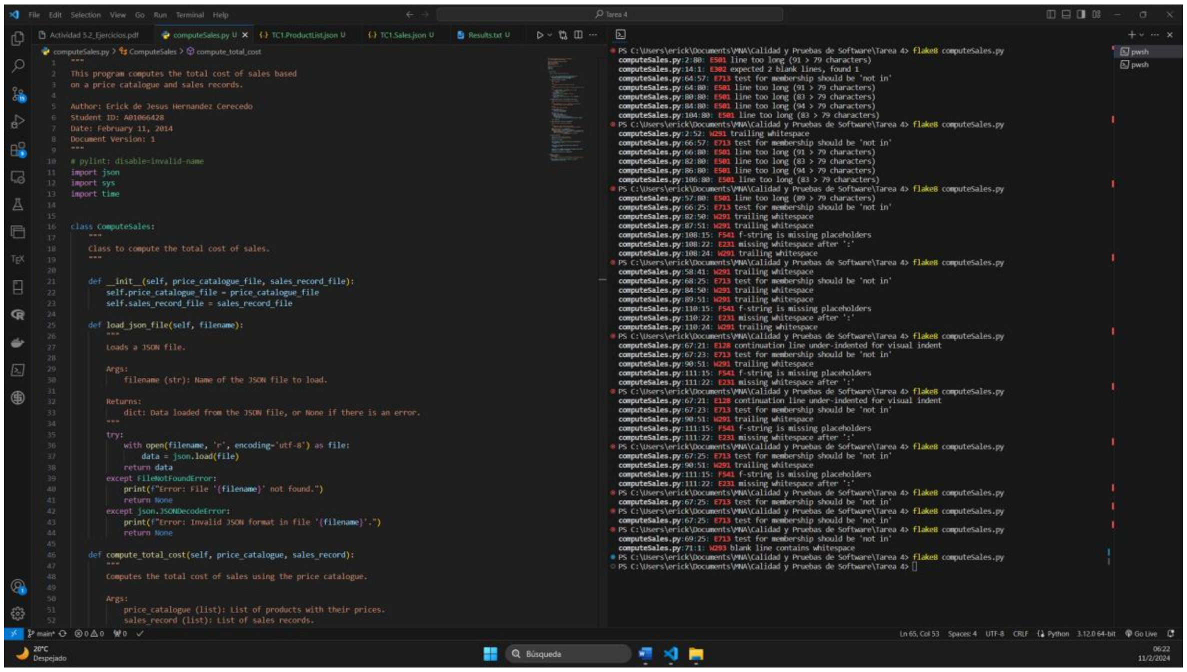
3. Análisis de Errores Flake:



```python
"""
This program computes the total cost of sales based on a price catalogue and sales records.

Author: Erick de Jesus Hernandez Cerecedo
Student ID: A01066428
Date: February 11, 2014
Document Version: 1
"""
# pylint: disable=invalid-name
import json
import sys
import time

class ComputeSales:
    """
    Class to compute the total cost of sales.
    """

    def __init__(self, price_catalogue_file, sales_record_file):
        self.price_catalogue_file = price_catalogue_file
        self.sales_record_file = sales_record_file

    def load_json_file(self, filename):
        """
        Loads a JSON file.

        Args:
            filename (str): Name of the JSON file to load.

        Returns:
            dict: Data loaded from the JSON file, or None if there is an error.
        """
        try:
            with open(filename, 'r', encoding='utf-8') as file:
                data = json.load(file)
            return data
        except FileNotFoundError:
            print(f"Error: File '{filename}' not found.")
            return None
        except json.JSONDecodeError:
            print(f"Error: Invalid JSON format in file '{filename}'.")
            return None

    def compute_total_cost(self, price_catalogue, sales_record):
        """
        Computes the total cost of sales using the price catalogue.

        Args:
            price_catalogue (list): List of products with their prices.
            sales_record (list): List of sales records.

        Returns:
```

```
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> flake8 computeSales.py
computeSales.py:2:80: E501 line too long (91 > 79 characters)
computeSales.py:14:1: E302 expected 2 blank lines, found 1
computeSales.py:64:57: E713 test for membership should be 'not in'
computeSales.py:64:80: E501 line too long (91 > 79 characters)
computeSales.py:80:80: E501 line too long (83 > 79 characters)
computeSales.py:84:80: E501 line too long (94 > 79 characters)
computeSales.py:104:80: E501 line too long (83 > 79 characters)
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> 
```

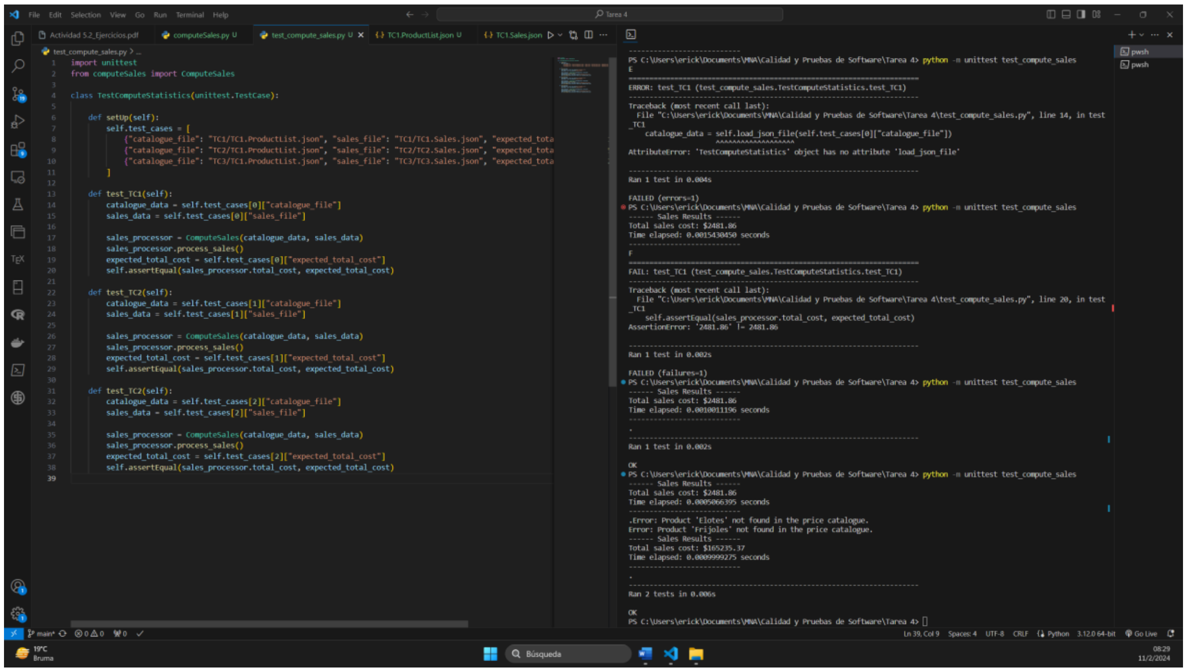
Corrección de errores con Flake8



```python
"""
This program computes the total cost of sales based
on a price catalogue and sales records.

Author: Erick de Jesus Hernandez Cerecedo
Student ID: A01066428
Date: February 11, 2014
Document Version: 1
"""
# pylint: disable=invalid-name
import json
import sys
import time


class ComputeSales:
    """
    Class to compute the total cost of sales.
    """

    def __init__(self, price_catalogue_file, sales_record_file):
        self.price_catalogue_file = price_catalogue_file
        self.sales_record_file = sales_record_file

    def load_json_file(self, filename):
        """
        Loads a JSON file.

        Args:
            filename (str): Name of the JSON file to load.

        Returns:
            dict: Data loaded from the JSON file, or None if there is an error.
        """
        try:
            with open(filename, 'r', encoding='utf-8') as file:
                data = json.load(file)
            return data
        except FileNotFoundError:
            print(f"Error: File '{filename}' not found.")
            return None
        except json.JSONDecodeError:
            print(f"Error: Invalid JSON format in file '{filename}'.")
            return None

    def compute_total_cost(self, price_catalogue, sales_record):
        """
        Computes the total cost of sales using the price catalogue.

        Args:
            price_catalogue (list): List of products with their prices.
            sales_record (list): List of sales records.
```

```
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> flake8 computeSales.py
computeSales.py:2:80: E501 line too long (91 > 79 characters)
computeSales.py:14:1: E302 expected 2 blank lines, found 1
computeSales.py:64:57: E713 test for membership should be 'not in'
computeSales.py:64:80: E501 line too long (91 > 79 characters)
computeSales.py:80:80: E501 line too long (83 > 79 characters)
computeSales.py:84:80: E501 line too long (94 > 79 characters)
computeSales.py:104:80: E501 line too long (83 > 79 characters)
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> flake8 computeSales.py
computeSales.py:2:52: W291 trailing whitespace
computeSales.py:66:57: E713 test for membership should be 'not in'
computeSales.py:66:80: E501 line too long (91 > 79 characters)
computeSales.py:82:80: E501 line too long (83 > 79 characters)
computeSales.py:86:80: E501 line too long (94 > 79 characters)
computeSales.py:106:80: E501 line too long (83 > 79 characters)
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> flake8 computeSales.py
computeSales.py:57:80: E501 line too long (89 > 79 characters)
computeSales.py:66:25: E713 test for membership should be 'not in'
computeSales.py:82:50: W291 trailing whitespace
computeSales.py:87:51: W291 trailing whitespace
computeSales.py:108:15: F541 f-string is missing placeholders
computeSales.py:108:22: E231 missing whitespace after ':'
computeSales.py:108:24: W291 trailing whitespace
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> flake8 computeSales.py
computeSales.py:58:41: W291 trailing whitespace
computeSales.py:68:25: E713 test for membership should be 'not in'
computeSales.py:84:50: W291 trailing whitespace
computeSales.py:89:51: W291 trailing whitespace
computeSales.py:110:15: F541 f-string is missing placeholders
computeSales.py:110:22: E231 missing whitespace after ':'
computeSales.py:110:24: W291 trailing whitespace
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> flake8 computeSales.py
computeSales.py:67:21: E128 continuation line under-indented for visual indent
computeSales.py:67:23: E713 test for membership should be 'not in'
computeSales.py:90:51: W291 trailing whitespace
computeSales.py:111:15: F541 f-string is missing placeholders
computeSales.py:111:22: E231 missing whitespace after ':'
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> flake8 computeSales.py
computeSales.py:67:21: E128 continuation line under-indented for visual indent
computeSales.py:67:23: E713 test for membership should be 'not in'
computeSales.py:90:51: W291 trailing whitespace
computeSales.py:111:15: F541 f-string is missing placeholders
computeSales.py:111:22: E231 missing whitespace after ':'
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> flake8 computeSales.py
computeSales.py:67:25: E713 test for membership should be 'not in'
computeSales.py:90:51: W291 trailing whitespace
computeSales.py:111:15: F541 f-string is missing placeholders
computeSales.py:111:22: E231 missing whitespace after ':'
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> flake8 computeSales.py
computeSales.py:67:25: E713 test for membership should be 'not in'
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> flake8 computeSales.py
computeSales.py:67:25: E713 test for membership should be 'not in'
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> flake8 computeSales.py
computeSales.py:69:25: E713 test for membership should be 'not in'
computeSales.py:71:1: W293 blank line contains whitespace
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> flake8 computeSales.py
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4>
```

4. Pruebas unitarias pasadas correctamente:



```python
import unittest
from computeSales import ComputeSales

class TestComputeStatistics(unittest.TestCase):

    def setUp(self):
        self.test_cases = [
            {"catalogue_file": "TC1/TC1.ProductList.json", "sales_file": "TC1/TC1.Sales.json", "expected_tota
            {"catalogue_file": "TC2/TC1.ProductList.json", "sales_file": "TC2/TC2.Sales.json", "expected_tota
            {"catalogue_file": "TC3/TC1.ProductList.json", "sales_file": "TC3/TC3.Sales.json", "expected_tota
        ]

    def test_TC1(self):
        catalogue_data = self.test_cases[0]["catalogue_file"]
        sales_data = self.test_cases[0]["sales_file"]

        sales_processor = ComputeSales(catalogue_data, sales_data)
        sales_processor.process_sales()
        expected_total_cost = self.test_cases[0]["expected_total_cost"]
        self.assertEqual(sales_processor.total_cost, expected_total_cost)

    def test_TC2(self):
        catalogue_data = self.test_cases[1]["catalogue_file"]
        sales_data = self.test_cases[1]["sales_file"]

        sales_processor = ComputeSales(catalogue_data, sales_data)
        sales_processor.process_sales()
        expected_total_cost = self.test_cases[1]["expected_total_cost"]
        self.assertEqual(sales_processor.total_cost, expected_total_cost)

    def test_TC2(self):
        catalogue_data = self.test_cases[2]["catalogue_file"]
        sales_data = self.test_cases[2]["sales_file"]

        sales_processor = ComputeSales(catalogue_data, sales_data)
        sales_processor.process_sales()
        expected_total_cost = self.test_cases[2]["expected_total_cost"]
        self.assertEqual(sales_processor.total_cost, expected_total_cost)
```

```
-------------------------
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> python -m unittest test_compute_sales
E
======================================================================
ERROR: test_TC1 (test_compute_sales.TestComputeStatistics.test_TC1)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4\test_compute_sales.py", line 14, in test
_TC1
    catalogue_data = self.load_json_file(self.test_cases[0]["catalogue_file"])
                     ^^^^^^^^^^^^^^^^^^^
AttributeError: 'TestComputeStatistics' object has no attribute 'load_json_file'

----------------------------------------------------------------------
Ran 1 test in 0.004s

FAILED (errors=1)
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> python -m unittest test_compute_sales
------ Sales Results ------
Total sales cost: $2481.86
Time elapsed: 0.0015430450 seconds
-------------------------
F
======================================================================
FAIL: test_TC1 (test_compute_sales.TestComputeStatistics.test_TC1)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4\test_compute_sales.py", line 20, in test
_TC1
    self.assertEqual(sales_processor.total_cost, expected_total_cost)
AssertionError: '2481.86' != 2481.86

----------------------------------------------------------------------
Ran 1 test in 0.002s

FAILED (failures=1)
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> python -m unittest test_compute_sales
------ Sales Results ------
Total sales cost: $2481.86
Time elapsed: 0.0010011196 seconds
-------------------------
.
----------------------------------------------------------------------
Ran 1 test in 0.002s

OK
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4> python -m unittest test_compute_sales
------ Sales Results ------
Total sales cost: $2481.86
Time elapsed: 0.0005066395 seconds
-------------------------
.Error: Product 'Elotes' not found in the price catalogue.
Error: Product 'Frijoles' not found in the price catalogue.
------ Sales Results ------
Total sales cost: $165235.37
Time elapsed: 0.0009999275 seconds
-------------------------
.
----------------------------------------------------------------------
Ran 2 tests in 0.006s

OK
PS C:\Users\erick\Documents\MNA\Calidad y Pruebas de Software\Tarea 4>
```