



Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey

Escuela de Ingeniería y Ciencias
Maestría en Inteligencia Artificial Aplicada

Actividad 5.2 Ejercicio de programación 2 y análisis estático

Pruebas de software y aseguramiento de la calidad

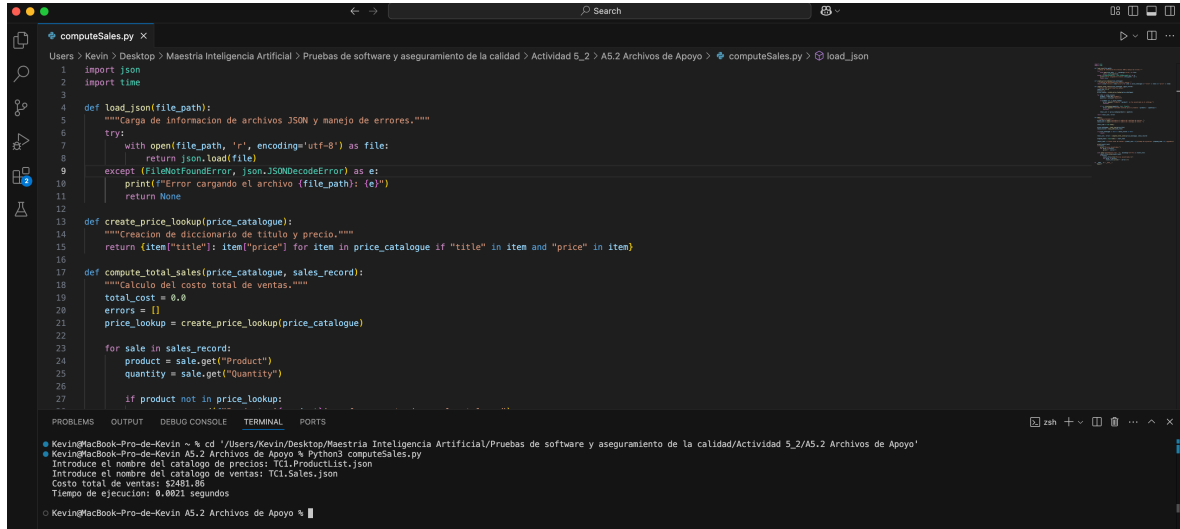
Kevin Gutiérrez Pérez A01795692

9 de febrero del 2025

Problema 1: ComputeSales.py

Evidencia de ejecución de código con resultados esperados:

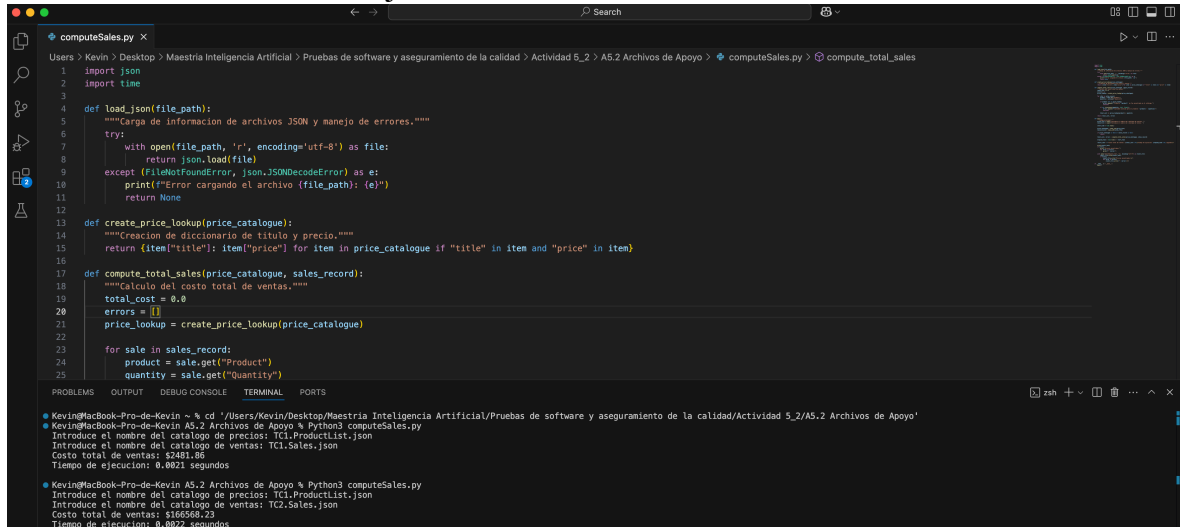
Archivo de ventas TC1.Sales.json:



```
computeSales.py
1 import json
2 import time
3
4 def load_json(file_path):
5     """Carga de informacion de archivos JSON y manejo de errores."""
6     try:
7         with open(file_path, 'r', encoding='utf-8') as file:
8             return json.load(file)
9     except (FileNotFoundError, json.JSONDecodeError) as e:
10         print(f"Error cargando el archivo {file_path}: {e}")
11         return None
12
13 def create_price_lookup(price_catalogue):
14     """Creacion de diccionario de titulo y precio."""
15     return {item["title"]: item["price"] for item in price_catalogue if "title" in item and "price" in item}
16
17 def compute_total_sales(price_catalogue, sales_record):
18     """Calculo del costo total de ventas."""
19     total_cost = 0.0
20     errors = []
21     price_lookup = create_price_lookup(price_catalogue)
22
23     for sale in sales_record:
24         product = sale.get("Product")
25         quantity = sale.get("Quantity")
26
27         if product not in price_lookup:
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
KevinMacBook-Pro-de-Kevin ~ % cd '/Users/Kevin/Desktop/Maestria Inteligencia Artificial/Pruebas de software y aseguramiento de la calidad/Actividad 5.2/A5.2 Archivos de Apoyo'
KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo % Python3 computeSales.py
Introduce el nombre del catalogo de precios: TC1.ProductList.json
Introduce el nombre del catalogo de ventas: TC1.Sales.json
Costo total de ventas: $2481.86
Tiempo de ejecucion: 0.0021 segundos
KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo %
```

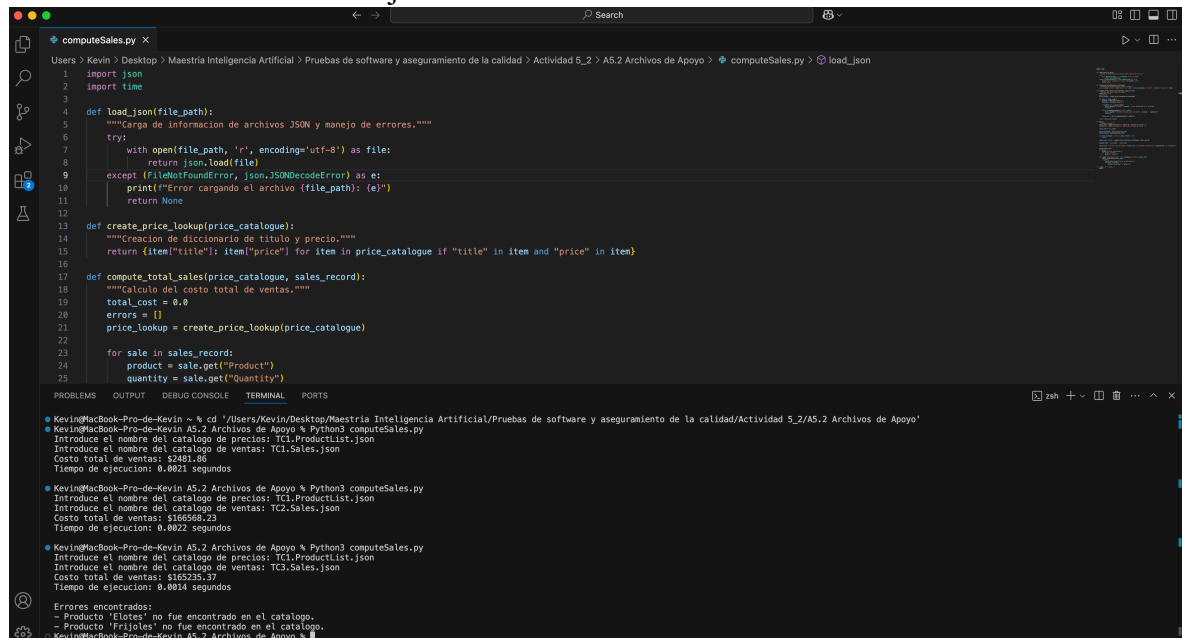
Archivo de ventas TC2.Sales.json:



```
computeSales.py
1 import json
2 import time
3
4 def load_json(file_path):
5     """Carga de informacion de archivos JSON y manejo de errores."""
6     try:
7         with open(file_path, 'r', encoding='utf-8') as file:
8             return json.load(file)
9     except (FileNotFoundError, json.JSONDecodeError) as e:
10         print(f"Error cargando el archivo {file_path}: {e}")
11         return None
12
13 def create_price_lookup(price_catalogue):
14     """Creacion de diccionario de titulo y precio."""
15     return {item["title"]: item["price"] for item in price_catalogue if "title" in item and "price" in item}
16
17 def compute_total_sales(price_catalogue, sales_record):
18     """Calculo del costo total de ventas."""
19     total_cost = 0.0
20     errors = []
21     price_lookup = create_price_lookup(price_catalogue)
22
23     for sale in sales_record:
24         product = sale.get("Product")
25         quantity = sale.get("Quantity")
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
KevinMacBook-Pro-de-Kevin ~ % cd '/Users/Kevin/Desktop/Maestria Inteligencia Artificial/Pruebas de software y aseguramiento de la calidad/Actividad 5.2/A5.2 Archivos de Apoyo'
KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo % Python3 computeSales.py
Introduce el nombre del catalogo de precios: TC1.ProductList.json
Introduce el nombre del catalogo de ventas: TC1.Sales.json
Costo total de ventas: $2481.86
Tiempo de ejecucion: 0.0021 segundos
KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo % Python3 computeSales.py
Introduce el nombre del catalogo de precios: TC1.ProductList.json
Introduce el nombre del catalogo de ventas: TC2.Sales.json
Costo total de ventas: $160568.23
Tiempo de ejecucion: 0.0022 segundos
```

Archivo de ventas TC3.Sales.json:



```
1 import json
2 import time
3
4 def load_json(file_path):
5     """Carga de informacion de archivos JSON y manejo de errores."""
6     try:
7         with open(file_path, 'r', encoding='utf-8') as file:
8             return json.load(file)
9     except (FileNotFoundException, json.JSONDecodeError) as e:
10         print(f"Error cargando el archivo {file_path}: {e}")
11         return None
12
13 def create_price_lookup(price_catalogue):
14     """Creacion de diccionario de titulo y precio."""
15     return {item["title"]: item["price"] for item in price_catalogue if "title" in item and "price" in item}
16
17 def compute_total_sales(price_catalogue, sales_record):
18     """Calculo del costo total de ventas."""
19     total_cost = 0.0
20     errors = []
21     price_lookup = create_price_lookup(price_catalogue)
22
23     for sale in sales_record:
24         product = sale.get("Product")
25         quantity = sale.get("Quantity")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

KevinMacBook-Pro-de-Kevin ~ % cd '/Users/Kevin/Desktop/Maestria Inteligencia Artificial/Pruebas de software y aseguramiento de la calidad/Actividad 5.2/A5.2 Archivos de Apoyo' & python3 computeSales.py

Introduce el nombre del catalogo de precios: TC1.ProductList.json

Introduce el nombre del catalogo de ventas: TC1.Sales.json

Costo total de ventas: \$2481.86

Tiempo de ejecucion: 0.0021 segundos

KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo % python3 computeSales.py

Introduce el nombre del catalogo de precios: TC1.ProductList.json

Introduce el nombre del catalogo de ventas: TC2.Sales.json

Costo total de ventas: \$16568.23

Tiempo de ejecucion: 0.0022 segundos

KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo % python3 computeSales.py

Introduce el nombre del catalogo de precios: TC1.ProductList.json

Introduce el nombre del catalogo de ventas: TC3.Sales.json

Costo total de ventas: \$165233.37

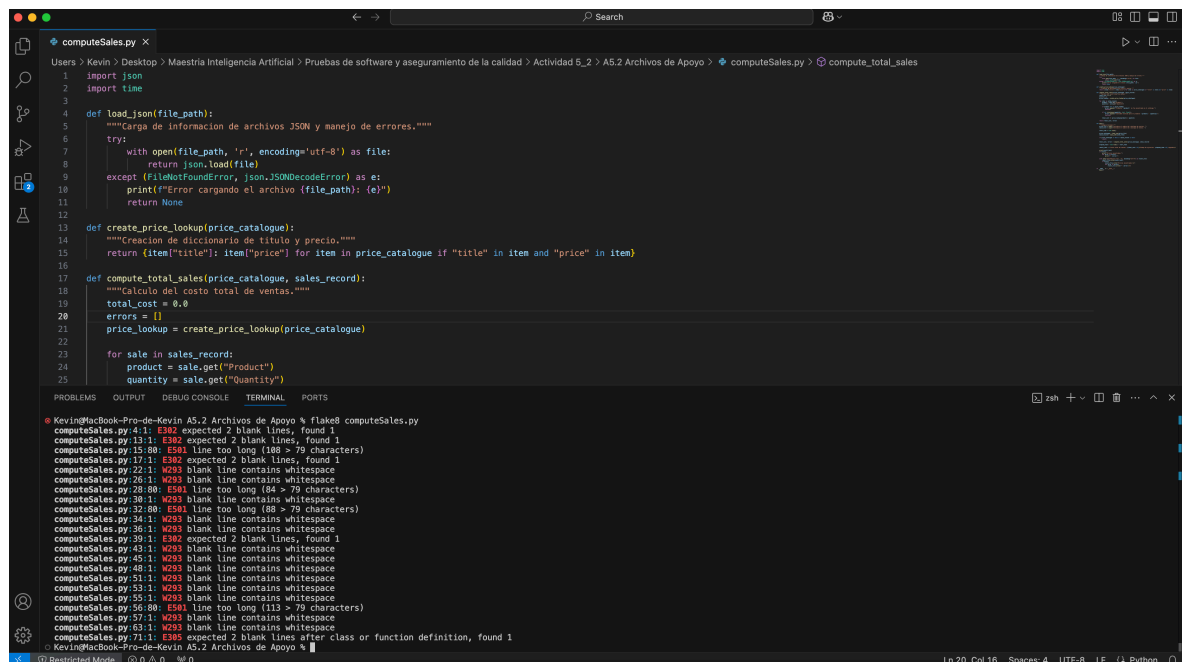
Tiempo de ejecucion: 0.0014 segundos

Errores encontrados:

- Producto "Elotes" no fue encontrado en el catalogo.
- Producto "Frijoles" no fue encontrado en el catalogo.

KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo %

Evidencia de ejecución de código con flake8:



```
1 import json
2 import time
3
4 def load_json(file_path):
5     """Carga de informacion de archivos JSON y manejo de errores."""
6     try:
7         with open(file_path, 'r', encoding='utf-8') as file:
8             return json.load(file)
9     except (FileNotFoundException, json.JSONDecodeError) as e:
10         print(f"Error cargando el archivo {file_path}: {e}")
11         return None
12
13 def create_price_lookup(price_catalogue):
14     """Creacion de diccionario de titulo y precio."""
15     return {item["title"]: item["price"] for item in price_catalogue if "title" in item and "price" in item}
16
17 def compute_total_sales(price_catalogue, sales_record):
18     """Calculo del costo total de ventas."""
19     total_cost = 0.0
20     errors = []
21     price_lookup = create_price_lookup(price_catalogue)
22
23     for sale in sales_record:
24         product = sale.get("Product")
25         quantity = sale.get("Quantity")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo % flake8 computeSales.py

computeSales.py:4:1: E302 expected 2 blank lines, found 1

computeSales.py:13:1: E302 expected 2 blank lines, found 1

computeSales.py:15:80: E501 line too long (188 > 79 characters)

computeSales.py:17:1: E302 expected 2 blank lines, found 1

computeSales.py:22:1: W293 blank line contains whitespace

computeSales.py:26:1: W293 blank line contains whitespace

computeSales.py:28:80: E501 line too long (84 > 79 characters)

computeSales.py:30:1: W293 blank line contains whitespace

computeSales.py:32:80: E501 line too long (88 > 79 characters)

computeSales.py:34:1: W293 blank line contains whitespace

computeSales.py:36:1: W293 blank line contains whitespace

computeSales.py:39:1: E302 expected 2 blank lines, found 1

computeSales.py:43:1: W293 blank line contains whitespace

computeSales.py:45:1: W293 blank line contains whitespace

computeSales.py:48:1: W293 blank line contains whitespace

computeSales.py:51:1: W293 blank line contains whitespace

computeSales.py:53:1: W293 blank line contains whitespace

computeSales.py:55:1: W293 blank line contains whitespace

computeSales.py:56:80: E501 line too long (112 > 79 characters)

computeSales.py:57:1: W293 blank line contains whitespace

computeSales.py:63:1: W293 blank line contains whitespace

computeSales.py:71:1: E302 expected 2 blank lines after class or function definition, found 1

KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo %

Errores mostrados y corregidos:

computeSales.py:4:1: E302 expected 2 blank lines, found 1

computeSales.py:13:1: E302 expected 2 blank lines, found 1

computeSales.py:15:80: E501 line too long (108 > 79 characters)
 computeSales.py:17:1: E302 expected 2 blank lines, found 1
 computeSales.py:22:1: W293 blank line contains whitespace
 computeSales.py:26:1: W293 blank line contains whitespace
 computeSales.py:28:80: E501 line too long (84 > 79 characters)
 computeSales.py:30:1: W293 blank line contains whitespace
 computeSales.py:32:80: E501 line too long (88 > 79 characters)
 computeSales.py:34:1: W293 blank line contains whitespace
 computeSales.py:36:1: W293 blank line contains whitespace
 computeSales.py:39:1: E302 expected 2 blank lines, found 1
 computeSales.py:43:1: W293 blank line contains whitespace
 computeSales.py:45:1: W293 blank line contains whitespace
 computeSales.py:48:1: W293 blank line contains whitespace
 computeSales.py:51:1: W293 blank line contains whitespace
 computeSales.py:53:1: W293 blank line contains whitespace
 computeSales.py:55:1: W293 blank line contains whitespace
 computeSales.py:56:80: E501 line too long (113 > 79 characters)
 computeSales.py:57:1: W293 blank line contains whitespace
 computeSales.py:63:1: W293 blank line contains whitespace
 computeSales.py:71:1: E305 expected 2 blank lines after class or function definition, found 1

Evidencia de funcionamiento de código una vez corregidos los errores:

```

def compute_total_sales(price_catalogue, sales_record):
    """Calculo del costo total de ventas."""
    total_cost = 0.0
    errors = []
    price_lookup = create_price_lookup(price_catalogue)

    for sale in sales_record:
        product = sale.get("Product")
        quantity = sale.get("Quantity")

        if product not in price_lookup:
            errors.append(f"Producto '{product}' no fue encontrado.")
            continue

        if not isinstance(quantity, (int, float)):
            errors.append(f"Cantidad invalida para '{product}': {quantity}")
            continue

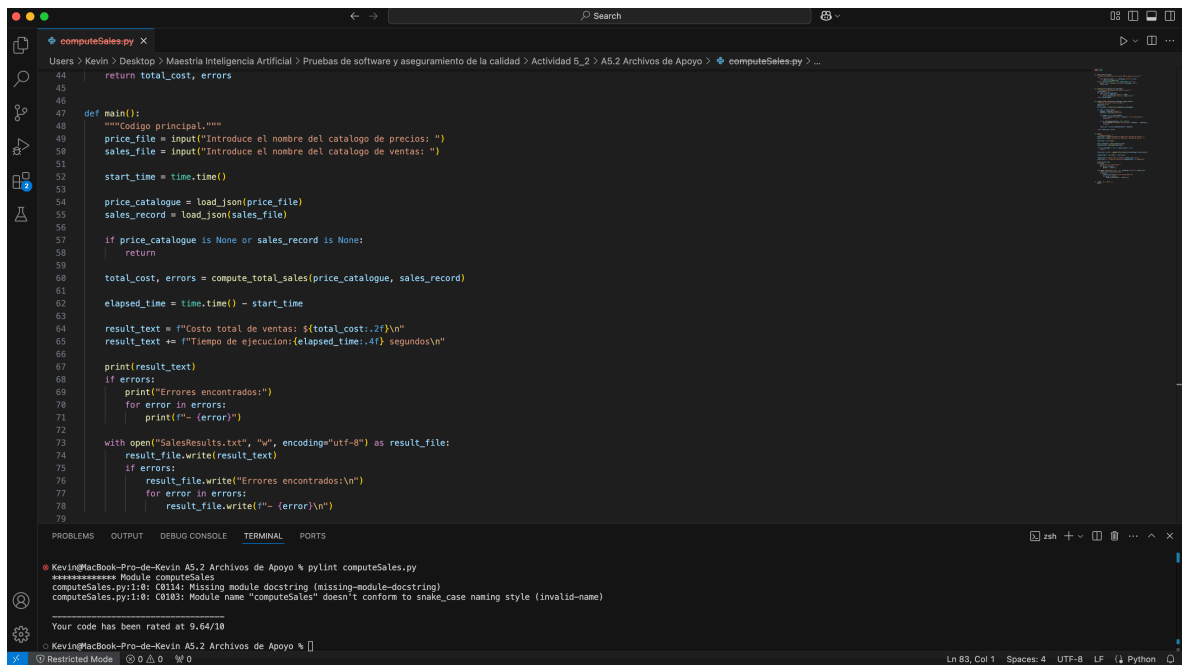
        total_cost += price_lookup[product] * quantity

    return total_cost, errors
  
```

```

computeSales.py:31:80: E501 line too long (88 > 79 characters)
computeSales.py:35:80: E501 line too long (88 > 79 characters)
computeSales.py:61:19: E225 missing whitespace around operator
KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo % flake8 computeSales.py
computeSales.py:17:80: E501 line too long (108 > 79 characters)
computeSales.py:31:80: E501 line too long (84 > 79 characters)
computeSales.py:35:80: E501 line too long (88 > 79 characters)
KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo % flake8 computeSales.py
computeSales.py:31:80: E501 line too long (88 > 79 characters)
computeSales.py:35:80: E501 line too long (88 > 79 characters)
KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo % flake8 computeSales.py
computeSales.py:39:80: E501 line too long (88 > 79 characters)
KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo % flake8 computeSales.py
computeSales.py:39:80: E501 line too long (88 > 79 characters)
KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo % Python3 computeSales.py
Introduce el nombre del catalogo de precios: TCI.ProductList.json
Introduce el nombre del catalogo de ventas: TCI.Sales.json
Costo total de ventas: $2481.86
Tiempo de ejecucion:0.0024 segundos
  
```

Evidencia de ejecución de pylint



The screenshot shows a code editor with a Python file named `computeSales.py`. The code defines a `main` function that prompts the user for price and sales file names, loads JSON data, and calculates total cost and execution time. The terminal at the bottom shows the output of a `pylint` command, which reports two errors: `C0114: Missing module docstring` and `C0103: Module name 'computeSales' doesn't conform to snake_case naming style`. The code is currently at line 83, column 1.

```
44     return total_cost, errors
45
46
47 def main():
48     """Codigo principal."""
49     price_file = input("Introduce el nombre del catalogo de precios: ")
50     sales_file = input("Introduce el nombre del catalogo de ventas: ")
51
52     start_time = time.time()
53
54     price_catalogue = load_json(price_file)
55     sales_record = load_json(sales_file)
56
57     if price_catalogue is None or sales_record is None:
58         return
59
60     total_cost, errors = compute_total_sales(price_catalogue, sales_record)
61
62     elapsed_time = time.time() - start_time
63
64     result_text = f"Costo total de ventas: ${total_cost:.2f}\n"
65     result_text += f"Tiempo de ejecucion:{elapsed_time:.4f} segundos\n"
66
67     print(result_text)
68     if errors:
69         print("Errores encontrados:")
70         for error in errors:
71             print(f"- {error}")
72
73     with open("SalesResults.txt", "w", encoding="utf-8") as result_file:
74         result_file.write(result_text)
75         if errors:
76             result_file.write("Errores encontrados:\n")
77             for error in errors:
78                 result_file.write(f"- {error}\n")
79
80
81
82
83
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo % pylint computeSales.py
***** Module computeSales
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)

Your code has been rated at 9.64/10

KevinMacBook-Pro-de-Kevin A5.2 Archivos de Apoyo %
```

Ln 83, Col 1 Spaces: 4 UTF-8 LF Python

Errores mostrados y corregidos:

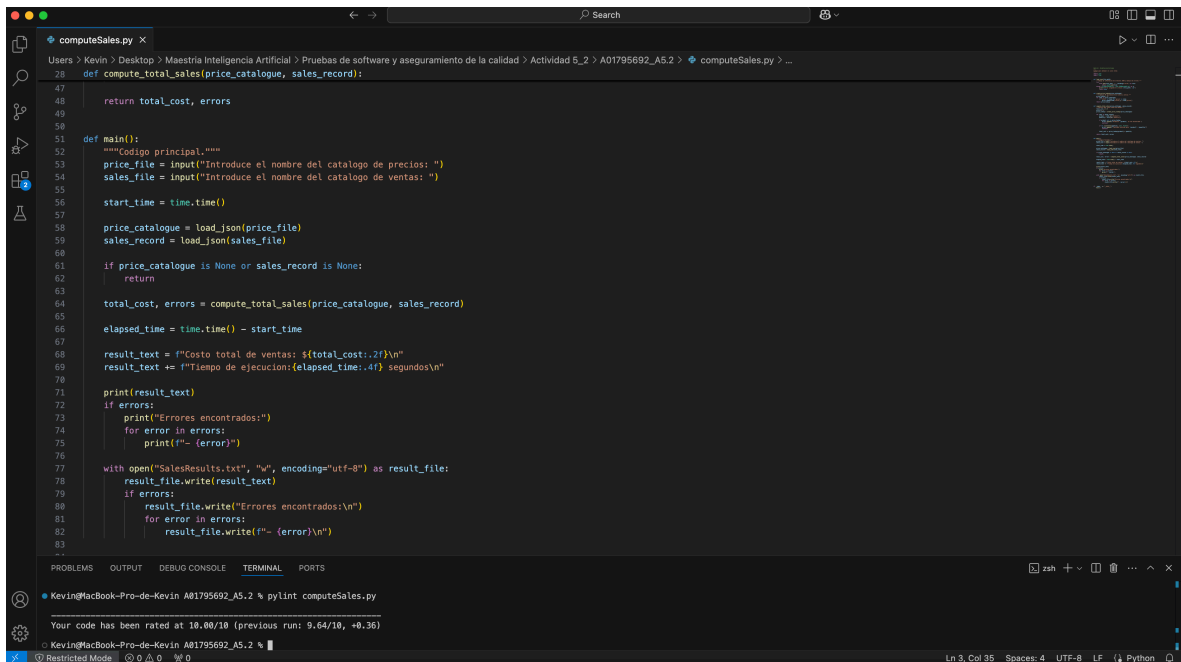
***** Module computeSales

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)

Your code has been rated at 9.64/10

Evidencia de codigo corregido en base a comentarios de pylint:



```
28 def compute_total_sales(price_catalogue, sales_record):
29     return total_cost, errors
30
31 def main():
32     """Codigo principal."""
33     price_file = input("Introduce el nombre del catalogo de precios: ")
34     sales_file = input("Introduce el nombre del catalogo de ventas: ")
35
36     start_time = time.time()
37
38     price_catalogue = load_json(price_file)
39     sales_record = load_json(sales_file)
40
41     if price_catalogue is None or sales_record is None:
42         return
43
44     total_cost, errors = compute_total_sales(price_catalogue, sales_record)
45
46     elapsed_time = time.time() - start_time
47
48     result_text = f"Costo total de ventas: ${total_cost:.2f}\n"
49     result_text += f"Tiempo de ejecucion:{elapsed_time:.4f} segundos\n"
50
51     print(result_text)
52     if errors:
53         print("Errores encontrados:")
54         for error in errors:
55             print(f"~ (error)")
56
57     with open("SalesResults.txt", "w", encoding="utf-8") as result_file:
58         result_file.write(result_text)
59         if errors:
60             result_file.write("Errores encontrados:\n")
61             for error in errors:
62                 result_file.write(f"~ (error)\n")
63
64 ...
65
66 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
67
68 KevinMacBook-Pro-de-Kevin A01795692_A5.2 % pylint computeSales.py
69
70 Your code has been rated at 10.00/10 (previous run: 9.64/10, +0.36)
71
72 KevinMacBook-Pro-de-Kevin A01795692_A5.2 %
```

Nota importante:

El error: computeSales.py:1:0: C0103: Module name "computeSales" doesn't conform to snake_case naming style (invalid-name)

Fue omitido con el codigo: #pylint: disable=invalid-name debido a que las instrucciones solicitaban ese nombre.