



Pruebas de software y aseguramiento de la calidad

TC4017.10

5.2 Ejercicio de programación 2 y análisis estático

Profesor titular: Dr. Gerardo Padilla Zárate

Profesor asistente: Mtra. Viridiana Rodríguez González

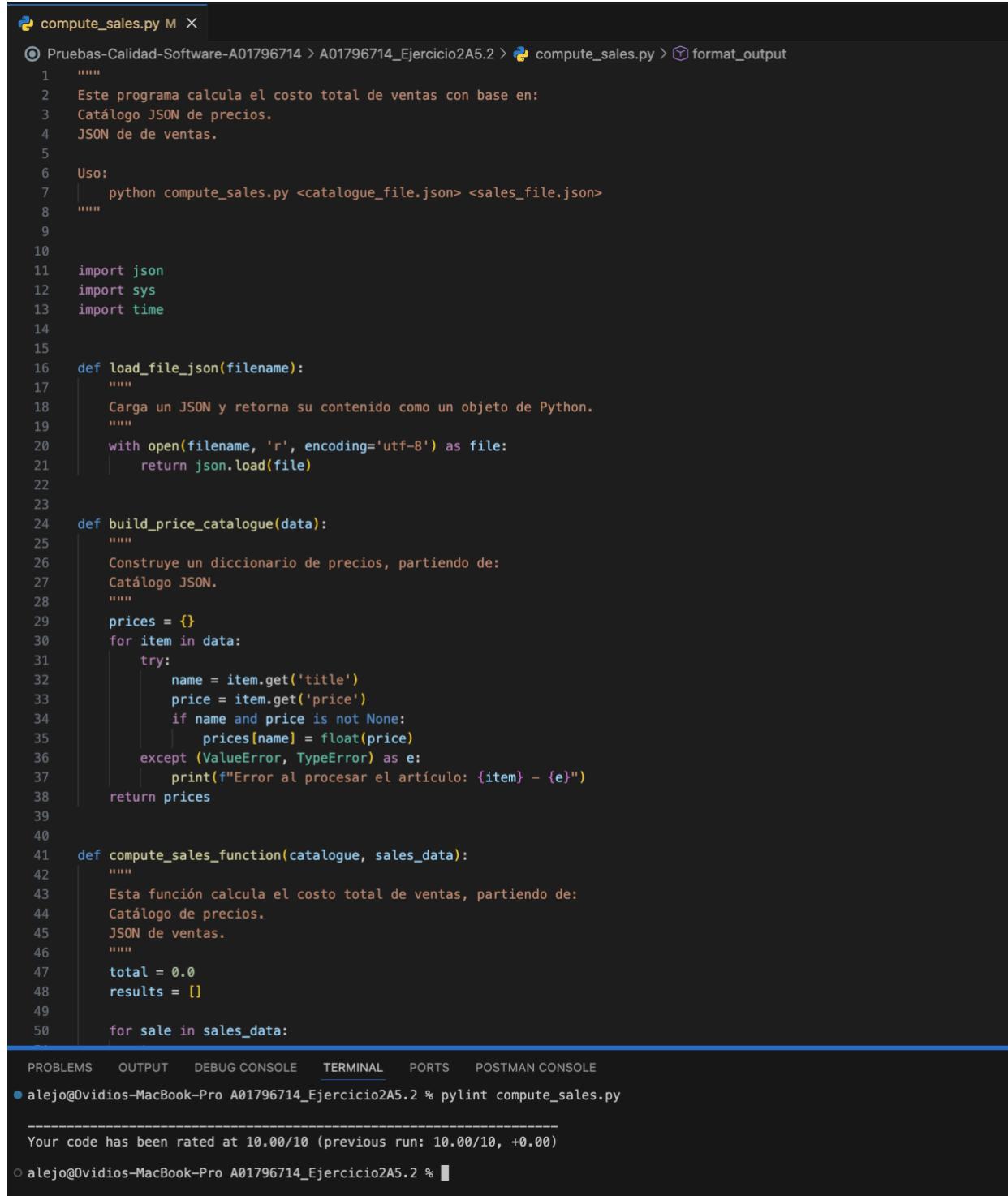
Alumno

A01796714 Ovidio Alejandro Hernández Ruano

Febrero 15, 2026

Ejercicio: Compute Sales

Evidencia de la correcta ejecución de *pylint*, aplicado al programa compute_sales.py



```
compute_sales.py M X
④ Pruebas-Calidad-Software-A01796714 > A01796714_Ejercicio2A5.2 > 🐍 compute_sales.py > ⌂ format_output

1      """
2 Este programa calcula el costo total de ventas con base en:
3 Catálogo JSON de precios.
4 JSON de ventas.
5
6 Uso:
7     python compute_sales.py <catalogue_file.json> <sales_file.json>
8 """
9
10
11 import json
12 import sys
13 import time
14
15
16 def load_file_json(filename):
17     """
18     Carga un JSON y retorna su contenido como un objeto de Python.
19     """
20     with open(filename, 'r', encoding='utf-8') as file:
21         return json.load(file)
22
23
24 def build_price_catalogue(data):
25     """
26     Construye un diccionario de precios, partiendo de:
27     Catálogo JSON.
28     """
29     prices = {}
30     for item in data:
31         try:
32             name = item.get('title')
33             price = item.get('price')
34             if name and price is not None:
35                 prices[name] = float(price)
36             except (ValueError, TypeError) as e:
37                 print(f"Error al procesar el artículo: {item} - {e}")
38     return prices
39
40
41 def compute_sales_function(catalogue, sales_data):
42     """
43     Esta función calcula el costo total de ventas, partiendo de:
44     Catálogo de precios.
45     JSON de ventas.
46     """
47     total = 0.0
48     results = []
49
50     for sale in sales_data:
51
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
● alejo@Ovidios-MacBook-Pro A01796714_Ejercicio2A5.2 % pylint compute_sales.py
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
○ alejo@Ovidios-MacBook-Pro A01796714_Ejercicio2A5.2 %
```

Evidencia de la correcta ejecución de *flake8*, aplicado al programa compute_sales.py

The screenshot shows a terminal window with the following content:

```
compute_sales.py M ×
● Pruebas-Calidad-Software-A01796714 > A01796714_Ejercicio2A5.2 > ⚡ compute_sales.py > ✎ format_output
1  """
2  Este programa calcula el costo total de ventas con base en:
3  Catálogo JSON de precios.
4  JSON de ventas.
5
6  Uso:
7  |   python compute_sales.py <catalogue_file.json> <sales_file.json>
8  """
9
10
11 import json
12 import sys
13 import time
14
15
16 def load_file_json(filename):
17     """
18     Carga un JSON y retorna su contenido como un objeto de Python.
19     """
20     with open(filename, 'r', encoding='utf-8') as file:
21         return json.load(file)
22
23
24 def build_price_catalogue(data):
25     """
26     Construye un diccionario de precios, partiendo de:
27     Catálogo JSON.
28     """
29     prices = {}
30     for item in data:
31         try:
32             name = item.get('title')
33             price = item.get('price')
34             if name and price is not None:
35                 prices[name] = float(price)
36         except (ValueError, TypeError) as e:
37             print(f"Error al procesar el artículo: {item} - {e}")
38     return prices
39
40
41 def compute_sales_function(catalogue, sales_data):
42     """
43     Esta función calcula el costo total de ventas, partiendo de:
44     Catálogo de precios.
45     JSON de ventas.
46     """
47     total = 0.0
48     results = []
49
50     for sale in sales_data:
51         ...
52
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE
● alejo@Ovidios-MacBook-Pro A01796714_Ejercicio2A5.2 % pylint compute_sales.py
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
● alejo@Ovidios-MacBook-Pro A01796714_Ejercicio2A5.2 % flake8 compute_sales.py
○ alejo@Ovidios-MacBook-Pro A01796714_Ejercicio2A5.2 %
```

Escenario de prueba TC1

Tiempo de ejecución del programa, para procesar el escenario de prueba TC1.

0.0014 segundos.

Total

\$2481.86

TC1: Evidencia de resultado en consola.

The screenshot shows a terminal window with the following details:

- Editor Area:** Shows the file `compute_sales.py` with the following content:

```
Este programa calcula el costo total de ventas con base en:  
1. ****  
2. Este programa calcula el costo total de ventas con base en:  
3. ****
```
- Terminal Output:** Shows the command run and the resulting output:

```
alejo@Ovidios-MacBook-Pro ~% python compute_sales.py ./TCList/TC1.ProductList.json ./TC1/TC1.Sales.json  
COSTO TOTAL DE VENTAS  
Tiempo: 0.0014 segundos
```
- Output Content:** The output displays a table of sales data with columns: Item, Quantity (x), Price (\$), and Total (\$). The table includes items like Rustic breakfast, Sandwich with salad, Raw legums, Fresh strawberry, Pears juice, Green smoothie, Cuban sandwich, Hazelnut in black ceramic bowl, Tacos, Plums, Fresh blueberries, Green smoothie, Corn, French fries, Ground beef meat burger, Hamburger in black ceramic bowl, Cherry, Homemade bread, Smoothie with chia seeds, Coxinha, Peaches on branch, Pesto with basil, Plums, Fresh blueberries, Green smoothie, Corn, French fries, Ground beef meat burger, Hazelnut in black ceramic bowl, Cherry, Homemade bread, Smoothie with chia seeds, Corn, Rustic breakfast, Sandwich with salad, Raw legums, Fresh strawberry, Pears juice, Green smoothie, Cuban sandwich, Hamburger in black ceramic bowl, Tortas, Plums, Fresh blueberries, Green smoothie, and Corn.
- Bottom of Output:** Shows the total cost and time taken:

```
TOTAL: $2481.86  
Tiempo: 0.0014 segundos
```

TC1: Evidencia de resultado en archivo .txt

```
1 -----  
2 COSTO TOTAL DE VENTAS  
3 -----  
4 Tiempo: 0.0014 segundos  
5 -----  
6  
7 Rustic breakfast      1   x   $ 21.32 =   $ 21.32  
8 Sandwich with salad   2   x   $ 22.48 =   $ 44.96  
9 Raw legums             1   x   $ 17.11 =   $ 17.11  
10 Fresh strawberry      1   x   $ 28.59 =   $ 28.59  
11 Pears juice            2   x   $ 19.49 =   $ 38.98  
12 Green smoothie         3   x   $ 17.68 =   $ 53.04  
13 Cuban sandwich         2   x   $ 18.50 =   $ 37.00  
14 Hazelnut in black ceramic bowl 2   x   $ 27.35 =   $ 54.70  
15 Tomatoes               1   x   $ 26.03 =   $ 26.03  
16 Plums                  1   x   $ 19.18 =   $ 19.18  
17 Fresh blueberries      2   x   $ 21.01 =   $ 42.02  
18 Green smoothie          2   x   $ 17.68 =   $ 35.36  
19 Corn                   3   x   $ 13.55 =   $ 40.65  
20 French fries            1   x   $ 18.32 =   $ 18.32  
21 Ground beef meat burger 1   x   $ 11.73 =   $ 11.73  
22 Hazelnut in black ceramic bowl 2   x   $ 27.35 =   $ 54.70  
23 Cherry                 2   x   $ 14.35 =   $ 28.70  
24 Homemade bread          1   x   $ 17.48 =   $ 17.48  
25 Smoothie with chia seeds 1   x   $ 25.26 =   $ 25.26  
26 Corn                   1   x   $ 13.55 =   $ 13.55  
27 Peaches on branch       2   x   $ 25.62 =   $ 51.24  
28 Pesto with basil        2   x   $ 18.19 =   $ 36.38  
29 Plums                  1   x   $ 19.18 =   $ 19.18  
30 Fresh blueberries       2   x   $ 21.01 =   $ 42.02  
31 Green smoothie          4   x   $ 17.68 =   $ 70.72  
32 Corn                   1   x   $ 13.55 =   $ 13.55  
33 French fries            2   x   $ 18.32 =   $ 36.64  
34 Ground beef meat burger 3   x   $ 11.73 =   $ 35.19  
35 Hazelnut in black ceramic bowl 1   x   $ 27.35 =   $ 27.35  
36 Cherry                 2   x   $ 14.35 =   $ 28.70  
37 Homemade bread          5   x   $ 17.48 =   $ 87.40  
38 Smoothie with chia seeds 2   x   $ 25.26 =   $ 50.52  
39 Corn                   10  x   $ 13.55 =   $ 135.50  
40 Rustic breakfast         2   x   $ 21.32 =   $ 42.64  
41 Sandwich with salad     3   x   $ 22.48 =   $ 67.44  
42 Raw legums              20  x   $ 17.11 =   $ 342.20  
43 Fresh strawberry         4   x   $ 28.59 =   $ 114.36  
44 Pears juice              1   x   $ 19.49 =   $ 19.49  
45 Green smoothie           8   x   $ 17.68 =   $ 141.44  
46 Cuban sandwich           9   x   $ 18.50 =   $ 166.50  
47 Hazelnut in black ceramic bowl 2   x   $ 27.35 =   $ 54.70  
48 Tomatoes                1   x   $ 26.03 =   $ 26.03  
49 Plums                   2   x   $ 19.18 =   $ 38.36  
50 Fresh blueberries        3   x   $ 21.01 =   $ 63.03  
51 Green smoothie           5   x   $ 17.68 =   $ 88.40  
52 Corn                     4   x   $ 13.55 =   $ 54.20  
53  
54 -----  
55 TOTAL: $2481.86  
56 -----  
57 Tiempo: 0.0014 segundos  
58 -----  
59 -----
```

Escenario de prueba TC2

Tiempo de ejecución del programa, para procesar el escenario de prueba TC2.

0.0004 segundos.

Total

\$166568.23

TC2: Evidencia de resultado en consola.

The screenshot shows a terminal window with the following command history:

```
Este programa calcula el costo total de ventas con base en:
1 Catalogo JSON de precios.
2 JSON de de ventas.
3 Uso:
4 python compute_sales.py <catalogue_file.json> <sales_file.json>
```

Below the command history, there is a table titled "COSTO TOTAL DE VENTAS" with the following data:

Item	Cantidad	Precio Unitario	Total
Rustic breakfast	200	\$ 21.32	= \$ 4264.00
Smoothie with salad	1	\$ 21.32	= \$ 21.32
Raw legume	11	\$ 17.11	= \$ 188.21
Fresh strawberry	221	\$ 28.59	= \$ 6318.39
Raw meat burger	2	\$ 17.11	= \$ 34.22
Green smoothie	400	\$ 17.68	= \$ 7072.00
Cold sandwich	2	\$ 18.30	= \$ 36.60
Hazelnut in black ceramic bowl	2	\$ 27.35	= \$ 54.70
Tomato	1	\$ 26.83	= \$ 26.83
Juice	250	\$ 21.81	= \$ 5450.00
Fresh blueberries	334	\$ 21.81	= \$ 7377.34
Green smoothie	308	\$ 17.68	= \$ 5390.00
Gof	69	\$ 17.55	= \$ 1214.00
French fries	33	\$ 18.32	= \$ 604.56
Ground meat burger	70	\$ 17.11	= \$ 1197.70
Hazelnut in black ceramic bowl	2	\$ 27.35	= \$ 54.70
Sweet Fresh strawberry	87	\$ 29.45	= \$ 2562.15
Homemade bread	1	\$ 17.48	= \$ 17.48
Smoothie with chia seeds	46	\$ 25.26	= \$ 1161.96
Corn	1	\$ 13.55	= \$ 13.55
Plums	2	\$ 19.18	= \$ 38.36
Fresh blueberries	46	\$ 21.81	= \$ 996.46
Green smoothie	1	\$ 17.68	= \$ 17.68
Corn	2	\$ 13.55	= \$ 27.10
French fries	2	\$ 18.32	= \$ 36.64
Ground beef meat burger	64	\$ 11.73	= \$ 756.72
Hazelnut in black ceramic bowl	2	\$ 27.35	= \$ 54.70
Sweet Fresh strawberry	131	\$ 29.45	= \$ 3857.95
Homemade bread	1	\$ 17.48	= \$ 17.48
Smoothie with chia seeds	13	\$ 25.26	= \$ 328.38
Corn	5	\$ 13.55	= \$ 67.75
Plums	645	\$ 19.18	= \$ 12371.18
Fresh blueberries	33	\$ 21.81	= \$ 733.33
Green smoothie	2	\$ 17.68	= \$ 35.36
Corn	46	\$ 13.55	= \$ 630.70
French fries	28	\$ 18.32	= \$ 366.48
Ground beef meat burger	4	\$ 11.73	= \$ 46.92
Hazelnut in black ceramic bowl	1	\$ 27.35	= \$ 27.35
Sweet Fresh strawberry	131	\$ 29.45	= \$ 3857.95
Homemade bread	1	\$ 17.48	= \$ 17.48
Smoothie with chia seeds	13	\$ 25.26	= \$ 328.38
Corn	678	\$ 13.55	= \$ 9186.90
Juice	254	\$ 21.81	= \$ 5508.32
Fresh blueberries	3445	\$ 21.81	= \$ 72379.45
Green smoothie	-123	\$ 17.68	= \$ -2174.64
Corn	445	\$ 13.55	= \$ 6029.75

At the bottom of the terminal, it says "TOTAL: \$166568.23" and "Tiempo: 0.0004 segundos".

TC2: Evidencia de resultado en archivo .txt

The screenshot shows a code editor interface with two main windows. On the left is the Explorer pane, which lists files and folders. On the right is the Editor pane, which displays the contents of a file named 'compute_sales.py'.

Explorer:

- OPEN EDITORS:
 - compute_sales.py A01796714_Ejer... M
 - TC2-SalesResults.txt A01796714_E... U
- UNTITLED (WORKSPACE)
 - Pruebas-Calidad-Software-A017...
 - 4.2 Ejercicio de programación 1
 - A01796714_Ejercicio2A5.2
 - Resultados
 - TC1-SalesResults.txt U
 - TC2-SalesResults.txt U
 - TC1
 - TC1.Sales.json
 - TC2
 - TC2.Sales.json
 - TC3
 - TCLIST
 - TC1.ProductList.json U
 - compute_sales.py M
 - .gitignore

Escenario de prueba TC3

Tiempo de ejecución del programa, para procesar el escenario de prueba TC3.

0.0004 segundos.

Total

\$165235.37

TC3: Evidencia de resultado en consola.

```
EXPLORER          compute_sales.py M
OPEN EDITORS      Pruebas-Calidad-Software-A01796714_Ejercicio2A5.2 > compute_sales.py > format_output
UNTITLED (RECENT) 
compute_sales.py A01796714_Ejercicio2A5.2
    Este programa calcula el costo total de ventas con base en:
    Catalogo JSON de precios.
    JSON de de ventas.
    Uso:
        python compute_sales.py <catalogue_file.json> <sales_file.json>
    Import json
PROBLEMS          OUTPUT      DEBEG CONSOLE   TERMINAL      PORTS      POSTMAN CONSOLE
alejandro@vidrios-MacBook-Pro A01796714_Ejercicio2A5.2 % python compute_sales.py ./TCList/TC1.ProductList.json ./TC3/TC3.Sales.json
  Producto no encontrado en el catalogo: 'Elotes'
  Producto no encontrado en el catalogo: 'Frifijoles'
COSTO TOTAL DE VENTAS
Tiempo: 0.0004 segundos
rustic_breakfast    200  x  $ 21.22 =  $ 4246.00
Sandwich with salad 23  x  $ 22.48 =  $ 517.84
Raw legues          11  x  $ 17.11 =  $ 188.21
Fresh blueberry     223  x  $ 28.35 =  $ 6173.85
Smoothie            4  x  $ 26.83 =  $ 107.32
Cuban sandwich      2  x  $ 17.11 =  $ 34.22
Smoothie            2  x  $ 28.35 =  $ 56.70
Hazelnut in black ceramic bowl 2  x  $ 27.35 =  $ 54.70
French fries        100  x  $ 13.55 =  $ 1355.00
Plum                258  x  $ 19.18 =  $ 4795.80
Fresh blueberries   335  x  $ 21.41 =  $ 7101.35
Green smoothie      380  x  $ 17.68 =  $ 5384.00
Corn                68  x  $ 13.55 =  $ 921.40
French fries        13  x  $ 18.55 =  $ 240.95
Ground beef meat burger 78  x  $ 11.73 =  $ 914.94
Homemade bread      2  x  $ 27.35 =  $ 54.70
Sweet Fresh strawberry 87  x  $ 29.45 =  $ 2562.15
Homemade bread      1  x  $ 17.48 =  $ 17.48
Smoothie with chia seeds 40  x  $ 13.55 =  $ 1155.00
Corn                1  x  $ 13.55 =  $ 13.55
Fruit               2  x  $ 19.18 =  $ 38.36
Green smoothie      2  x  $ 17.68 =  $ 35.36
Corn                2  x  $ 13.55 =  $ 27.10
French fries        64  x  $ 11.73 =  $ 756.72
Ground beef meat burger 5  x  $ 27.35 =  $ 136.75
Homemade bread      3  x  $ 29.45 =  $ 88.35
Sweet Fresh strawberry 5  x  $ 29.45 =  $ 147.25
Homemade bread      456  x  $ 17.48 =  $ 7978.88
Smoothie with chia seeds 2  x  $ 23.25 =  $ 46.50
Corn                5  x  $ 13.55 =  $ 67.75
Fruit               45  x  $ 19.18 =  $ 858.18
Fresh blueberries   35  x  $ 21.41 =  $ 735.35
Green smoothie      2  x  $ 17.68 =  $ 35.36
Corn                40  x  $ 13.55 =  $ 542.00
Ground beef meat burger 4  x  $ 11.73 =  $ 46.92
Hazelnut in black ceramic bowl 1  x  $ 27.35 =  $ 27.35
Sweet Fresh strawberry 131  x  $ 29.45 =  $ 3857.95
Homemade bread      9  x  $ 17.48 =  $ 157.32
Smoothie with chia seeds 13  x  $ 23.25 =  $ 302.25
Corn                678  x  $ 13.55 =  $ 9186.90
Fruit               35  x  $ 19.18 =  $ 669.30
Fresh blueberries   3445  x  $ 21.41 =  $ 72379.45
Green smoothie      -123  x  $ 17.68 =  $ -2174.64
Corn                445  x  $ 13.55 =  $ 6029.75
TOTAL: $165235.37
Tiempo: 0.0004 segundos
```

TC3: Evidencia de resultado en archivo .txt

The screenshot shows a terminal window with two tabs: 'compute_sales.py M' and 'TC3-SalesResults.txt U'. The terminal output is as follows:

```
1 -----  
2 COSTO TOTAL DE VENTAS  
3 -----  
4 Tiempo: 0.0004 segundos  
5 -----  
6  
7 Rustic breakfast      200  X   $ 21.32 =   $ 4264.00  
8 Sandwich with salad    23   X   $ 22.48 =   $ 517.04  
9 Raw legums              11   X   $ 17.11 =   $ 188.21  
10 Fresh strawberry       221  X   $ 28.59 =   $ 6318.39  
11 Raw legums              2   X   $ 17.11 =   $ 34.22  
12 Green smoothie         400  X   $ 17.68 =   $ 7072.00  
13 Cuban sandwich         2   X   $ 18.50 =   $ 37.00  
14 Hazelnut in black ceramic bowl 2   X   $ 27.35 =   $ 54.70  
15 Tomatoes                1   X   $ 26.03 =   $ 26.03  
16 Plums                   250  X   $ 19.18 =   $ 4795.00  
17 Fresh blueberries       334  X   $ 21.01 =   $ 7017.34  
18 Green smoothie          300  X   $ 17.68 =   $ 5304.00  
19 Corn                      68   X   $ 13.55 =   $ 921.40  
20 French fries             33   X   $ 18.32 =   $ 604.56  
21 Ground beef meat burger 78   X   $ 11.73 =   $ 914.94  
22 Hazelnut in black ceramic bowl 2   X   $ 27.35 =   $ 54.70  
23 Sweet fresh strawberry   87   X   $ 29.45 =   $ 2562.15  
24 Homemade bread            1   X   $ 17.48 =   $ 17.48  
25 Smoothie with chia seeds 46   X   $ 25.26 =   $ 1161.96  
26 Corn                      1   X   $ 13.55 =   $ 13.55  
27 Plums                   2   X   $ 19.18 =   $ 38.36  
28 Green smoothie           1   X   $ 17.68 =   $ 17.68  
29 Corn                      2   X   $ 13.55 =   $ 27.10  
30 French fries              4   X   $ 18.32 =   $ 73.28  
31 Ground beef meat burger 64   X   $ 11.73 =   $ 750.72  
32 Hazelnut in black ceramic bowl 2   X   $ 27.35 =   $ 54.70  
33 Sweet fresh strawberry   3   X   $ 29.45 =   $ 88.35  
34 Homemade bread            456  X   $ 17.48 =   $ 7970.88  
35 Smoothie with chia seeds 2   X   $ 25.26 =   $ 50.52  
36 Corn                      5   X   $ 13.55 =   $ 67.75  
37 Plums                   645  X   $ 19.18 =   $ 12371.10  
38 Fresh blueberries        -35  X   $ 21.01 =   $ -735.35  
39 Green smoothie           2   X   $ 17.68 =   $ 35.36  
40 Corn                      465  X   $ 13.55 =   $ 6300.75  
41 Ground beef meat burger 4   X   $ 11.73 =   $ 46.92  
42 Hazelnut in black ceramic bowl 1   X   $ 27.35 =   $ 27.35  
43 Sweet fresh strawberry   131  X   $ 29.45 =   $ 3857.95  
44 Homemade bread            9   X   $ 17.48 =   $ 157.32  
45 Smoothie with chia seeds 13   X   $ 25.26 =   $ 328.38  
46 Corn                      678  X   $ 13.55 =   $ 9186.90  
47 Plums                   334  X   $ 19.18 =   $ 6406.12  
48 Fresh blueberries        3445 X   $ 21.01 =   $ 72379.45  
49 Green smoothie           -123 X   $ 17.68 =   $ -2174.64  
50 Corn                      445  X   $ 13.55 =   $ 6029.75  
51 -----  
52 -----  
53 TOTAL: $165235.37  
54 -----  
55 Tiempo: 0.0004 segundos  
56 -----  
57 -----
```

Commit realizados

Evidencia del commit realizado para subir los cambios correspondientes a la corrección de errores en el código, tras el análisis estático de código realizado con *pylint* y *flake8*.

```
EXPLORER OPEN EDITORS UNTITLED (WORKSPACE) Pruebas-Calidad-Software-A01796714 > A01796714_Ejercicio2A5.2 > compute_sales.py > format_output
41 def compute_sales_function(catalogue, sales_data):
70     except(ValueError, typeError) as e:
71         print("Error al procesar: (sale) - (e)")
72
73     return total, results
74
75
76 def format_output(total, elapsed_time, results):
77
78     """ Función para formatear la salida de los resultados, partiendo de:
79     Costo total de ventas.
80     Tiempo de ejecución.
81     Resultados detallados.
82
83     output = []
84     output.append("-" * 98)
85     output.append("COSTO TOTAL DE VENTAS")
86     output.append("-" * 98)
87     output.append("Tiempo: (elapsed_time:.4f) segundos")
88     output.append("-" * 98)
89     output.append("")
90
91     for product, qty, price, subtotal in results:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS POSTMAN CONSOLE

- alejo@Ovidios-MacBook-Pro Pruebas-Calidad-Software-A01796714 % git add ./A01796714_Ejercicio2A5.2
- alejo@Ovidios-MacBook-Pro Pruebas-Calidad-Software-A01796714 % git status

On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
(use "git restore --staged <file>..." to unstage)
modified: A01796714_Ejercicio2A5.2/Resultados/SalesResults.txt
new file: A01796714_Ejercicio2A5.2/Resultados/TC1-SalesResults.txt
new file: A01796714_Ejercicio2A5.2/Resultados/TC2-SalesResults.txt
new file: A01796714_Ejercicio2A5.2/Resultados/TC3-SalesResults.txt
deleted: A01796714_Ejercicio2A5.2/TC1.Sales.json
deleted: A01796714_Ejercicio2A5.2/TC2.Sales.json
deleted: A01796714_Ejercicio2A5.2/TC3.Sales.json
renamed: A01796714_Ejercicio2A5.2/TCL.ProductList.json => A01796714_Ejercicio2A5.2/TCL.TC1.ProductList.json
modified: A01796714_Ejercicio2A5.2/compute_sales.py

Untracked files:
(use "git add <file>..." to include in what will be committed)
".2 Ejercicio de programación1/SalesResults.txt"

- alejo@Ovidios-MacBook-Pro Pruebas-Calidad-Software-A01796714 % git commit -m "Se realizan las correcciones a los errores detectados por pylint y flake8"
- [main 39f0c64] Se realizan las correcciones a los errores detectados por pylint y flake8
9 files changed, 106 insertions(+), 106 deletions(-)
create mode 100644 A01796714_Ejercicio2A5.2/Resultados/TC1-SalesResults.txt
create mode 100644 A01796714_Ejercicio2A5.2/Resultados/TC2-SalesResults.txt
create mode 100644 A01796714_Ejercicio2A5.2/Resultados/TC3-SalesResults.txt
delete mode 100644 A01796714_Ejercicio2A5.2/TC1.Sales.json
delete mode 100644 A01796714_Ejercicio2A5.2/TC2.Sales.json
delete mode 100644 A01796714_Ejercicio2A5.2/TC3.Sales.json
rename A01796714_Ejercicio2A5.2/I => TCL.ProductList.json (100%)
- alejo@Ovidios-MacBook-Pro Pruebas-Calidad-Software-A01796714 % git push origin main

Enumerating objects: 15, done.
Counting objects: 15, done.
Delta compression using up to 10 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (10/10), 1.55 KB | 1.55 MB/s, done.
Total 10 (delta 6), reused 0 (delta 0) pack-reduce from 8
remote: Resolving deltas: 100% (6/6), completed with 3 local objects.
To https://github.com/OvidioHernandezA01796714/Pruebas-Calidad-Software-A01796714.git
 b3589cc..39f0c64 main --> main
alejo@Ovidios-MacBook-Pro Pruebas-Calidad-Software-A01796714 %

OvidioHernandezA01796714 / Pruebas-Calidad-Software-A01796714 (Public)

Code Issues Pull requests Actions Projects Security Insights

Files

main Go to file

4.2 Ejercicio de programación 1 A01796714_Ejercicio2A5.2 Resultados TC1 TC2 TC3 TCL .DS_Store compute_sales.py .gitignore

Pruebas-Calidad-Software-A01796714 / A01796714_Ejercicio2A5.2 /

OvidioHernandezA01796714 Se realizan las correcciones a los errores detectados por p... 39f0c64 · 2 minutes ago History

Name	Last commit message	Last commit date
..		
Resultados	Se realizan las correcciones a los errores detectados por p...	2 minutes ago
TC1	Primer commit Actividad5.2	2 days ago
TC2	Primer commit Actividad5.2	2 days ago
TC3	Primer commit Actividad5.2	2 days ago
TCL	Se realizan las correcciones a los errores detectados por p...	2 minutes ago
.DS_Store	Primer commit Actividad5.2	2 days ago
compute_sales.py	Se realizan las correcciones a los errores detectados por p...	2 minutes ago