

## TC1

The screenshot shows a VS Code editor with a Python script named `computeSales.py` and its output in the terminal. The script processes a list of products and their quantities, calculating the total sales and the execution time.

```
def main():
    titulo = "REPORTE DE VENTAS".center(130)
    reporte = [
        separador,
        titulo,
        separador,
        df_ventas.to_string(index=False),
        separador,
        f"Total: {total:,4f}",
        separador
    ]
    if not df_ventas_invalidas.empty:
        reporte.append("\nPRODUCTOS INVALIDOS:")
        reporte.append(df_ventas_invalidas.to_string(index=False))
        reporte.append(separador)
    reporte.append(f"Tiempo de ejecucion: {tiempo_ejecucion:4f} segundos")
    reporte.append(separador)
    reporte_final = "\n".join(reporte)
    print(reporte_final)
    try:
        with open("SalesResults.txt", "w", encoding="utf-8") as archivo:
            archivo.write(reporte_final + "\n")
    except IOError as err:
        print(f"Error al guardar el archivo: {err}")
    if __name__ == "__main__":
        main()
```

The terminal output shows the following data:

DATE	PRODUCT	QUANTITY	PRICE	TOTAL
02/12/23	Cherry	2	14.35	28.70
02/12/23	Homemade bread	5	17.48	87.40
02/12/23	Smoothie with chia seeds	2	25.26	50.52
02/12/23	Corn	18	13.55	243.90
02/12/23	Rustic breakfast	2	21.32	42.64
02/12/23	Sandwich with salad	2	22.48	44.96
02/12/23	Raw legums	28	17.11	479.08
02/12/23	Fresh strawberry	4	26.59	106.36
02/12/23	Pears juice	1	19.49	19.49
02/12/23	Green smoothie	8	17.68	141.44
02/12/23	Cuban sandwich	9	18.59	167.31
02/12/23	Hazelnut in black ceramic bowl	2	27.35	54.70
02/12/23	Tomatoes	1	26.63	26.63
02/12/23	Plums	2	19.18	38.36
02/12/23	Fresh blueberries	3	21.01	63.03
02/12/23	Green smoothie	5	17.68	88.40
02/12/23	Corn	4	13.55	54.20

Total: 2,481.8600  
Tiempo de ejecucion: 0.0104 segundos

## TC2

The screenshot shows a VS Code editor with a Python script named `computeSales.py` and its output in the terminal. The script processes a list of products and their quantities, calculating the total sales and the execution time.

```
def main():
    titulo = "REPORTE DE VENTAS".center(130)
    reporte = [
        separador,
        titulo,
        separador,
        df_ventas.to_string(index=False),
        separador,
        f"Total: {total:,4f}",
        separador
    ]
    if not df_ventas_invalidas.empty:
        reporte.append("\nPRODUCTOS INVALIDOS:")
        reporte.append(df_ventas_invalidas.to_string(index=False))
        reporte.append(separador)
    reporte.append(f"Tiempo de ejecucion: {tiempo_ejecucion:4f} segundos")
    reporte.append(separador)
    reporte_final = "\n".join(reporte)
    print(reporte_final)
    try:
        with open("SalesResults.txt", "w", encoding="utf-8") as archivo:
            archivo.write(reporte_final + "\n")
    except IOError as err:
        print(f"Error al guardar el archivo: {err}")
    if __name__ == "__main__":
        main()
```

The terminal output shows the following data:

DATE	PRODUCT	QUANTITY	PRICE	TOTAL
02/12/23	Smoothie with chia seeds	2	25.26	50.52
02/12/23	Corn	5	13.55	67.75
02/12/23	Plums	445	19.18	8535.10
02/12/23	Fresh blueberries	35	21.01	735.35
02/12/23	Green smoothie	2	17.68	35.36
02/12/23	Corn	18	13.55	243.90
02/12/23	French fries	28	18.32	513.76
02/12/23	Ground beef meat burger	4	11.73	46.92
02/12/23	Hazelnut in black ceramic bowl	1	27.35	27.35
02/12/23	Sweet fresh strawberry	131	29.43	3856.43
02/12/23	Homemade bread	9	17.48	157.32
02/12/23	Smoothie with chia seeds	13	25.26	328.38
02/12/23	Corn	678	13.55	9186.90
02/12/23	Plums	334	19.18	6406.12
02/12/23	Fresh blueberries	3445	21.01	72379.45
02/12/23	Green smoothie	-123	17.68	-2174.64
02/12/23	Corn	445	13.55	6029.75

Total: 166,568.2300  
Tiempo de ejecucion: 0.0126 segundos

# TC3

The screenshot shows a VS Code editor with the following components:

- EXPLORER:**
  - Root: MNA-A01796560-PRUEBAS-...
  - 4.2
  - P1
    - results
    - source
      - computeStatistics.py
    - tests
      - TC1.txt
      - TC2.txt
      - TC3.txt
      - TC4.txt
      - TC5.txt
      - TC6.txt
      - TC7.txt
    - P2
    - P3
      - 5.2/P1
        - results
          - TC1.SalesResults.txt
          - TC2.SalesResults.txt
        - source
          - computeSales.py
      - priceCatalogue.json
      - salesRecord.json
      - SalesResults.txt
      - tests
        - TC1.ProductList.json
        - TC1.Sales.json
        - TC2.Sales.json
        - TC3.Sales.json
      - README.md
  - EDITOR:**
    - `computeSales.py` (line 59):
 

```
def main():
    titulo = "REPORTE DE VENTAS".center(130)
    reporte = [
        separador,
        titulo,
        separador,
        df_ventas.to_string(index=False),
        separador,
        f"Total: {total:,4f}",
        separador
    ]
    if not df_ventas_invalidas.empty:
        reporte.append("\nPRODUCTOS INVALIDOS:")
        reporte.append(df_ventas_invalidas.to_string(index=False))
        reporte.append(separador)
    reporte.append(f"Tiempo de ejecucion: {tiempo_ejecucion:4f} segundos")
    reporte.append(separador)

    reporte_final = "\n".join(reporte)

    print(reporte_final)

    try:
        with open("SalesResults.txt", "w", encoding="utf-8") as archivo:
            archivo.write(reporte_final + "\n")
    except IOError as err:
        print(f"Error al guardar el archivo: {err}")

    if __name__ == "__main__":
        main()
```
    - `salesRecord.json` (line 1):
 

```
{
  "SALE_ID": 1,
  "SALE_Date": "01/12/23",
  "Product": "Rustic breakfast",
  "Quantity": 200
},
{
  "SALE_ID": 1,
  "SALE_Date": "01/12/23",
  "Product": "Sandwich with salad",
  "Quantity": 23
},
{
  "SALE_ID": 1,
  "SALE_Date": "01/12/23",
  "Product": "Maw legums",
  "Quantity": 11
},
{
  "SALE_ID": 2,
  "SALE_Date": "01/12/23",
  "Product": "Fresh strawberry",
  "Quantity": 221
},
{
  "SALE_ID": 2,
  "SALE_Date": "01/12/23",
  "Product": "Maw legums",
  "Quantity": 2
},
{
  "SALE_ID": 3,
  "SALE_Date": "01/12/23",
  "Product": "Green smoothie",
```
  - TERMINAL:**

```
@robertobazua ~ - /MNA-A01796560-pruebas-de-software/5.2/P1/source (feature-5) $ python3 computeSales.py priceCatalogue.json salesRecord.json
9 02/12/23 Frijoles 100 NaN NaN
9 02/12/23 Ground beef meat burger 4 NaN NaN
9 02/12/23 Hazelnut in black ceramic bowl 1 Hazelnut in black ceramic bowl 27.35 27.35
9 02/12/23 Sweet fresh strawberry 133 Sweet fresh strawberry 28.45 3837.95
9 02/12/23 Homemade bread 9 Homemade bread 17.48 157.32
10 02/12/23 Smoothie with chia seeds 13 Smoothie with chia seeds 25.26 328.38
10 02/12/23 Corn 678 Corn 13.55 9186.98
10 02/12/23 Plums 324 Plums 19.18 6460.12
10 02/12/23 Fresh blueberries 3445 Fresh blueberries 21.01 72379.45
10 02/12/23 Green smoothie -123 Green smoothie 17.68 -2174.64
10 02/12/23 Corn 445 Corn 13.55 6025.75

Total: 165,235,3700

PRODUCTOS INVALIDOS:
SALE_ID SALE_Date Product Quantity title price
6 01/12/23 f1otes 100 NaN NaN
9 02/12/23 Frijoles 100 NaN NaN

Tiempo de ejecucion: 0.0108 segundos
```