

# Reporte LifeStore - 01

---

HÉCTOR GERARDO PAREDES CASTILLO

13 DE FEBRERO DE 2022

# Índice

---

1

Introducción

2

Código

3

Análisis adicionales

4

Resultados

5

Conclusiones

# INTRODUCCIÓN

---

# Introducción

---

Lifestore es una tienda de artículos de tecnología que desea conocer sus resultados anuales por mes, producto, categoría y el total anual. Se analizarán los siguientes rubros:

- Productos más vendidos
- Productos más buscados
- Productos con major y peor reseña
- Análisis de ventas por categoría
- Ventas mensuales
- Ventas anuales
- Productos con devoluciones
- Stock y demanda de productos

# CÓDIGO

---

LINK GITHUB:[HECTORGPCAS/EMTECH\\_HECTORPAREDES](https://github.com/HECTORGPCAS/EMTECH_HECTORPAREDES)  
[\(GITHUB.COM\)](https://github.com/HECTORGPCAS/EMTECH_HECTORPAREDES)

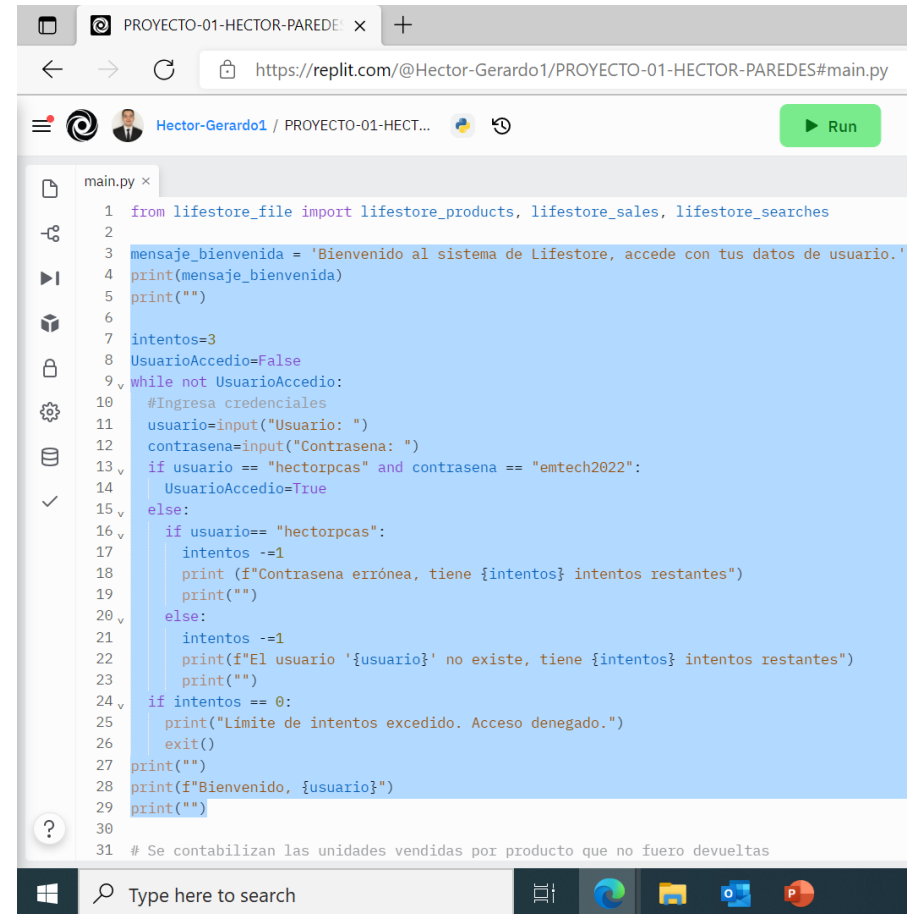
# Validación de usuario

Se creó un ciclo while de máximo 3 intentos para acceder correctamente el nombre de usuario y la contraseña para acceder al Sistema LifeStore.

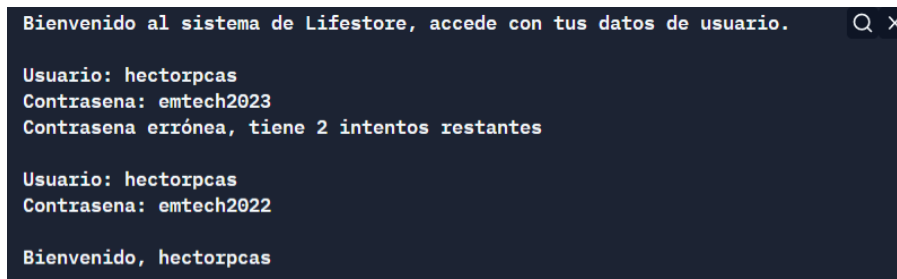
Usuario: hectorpcas

Contraseña: emtech2022

Una vez validadas las credenciales, se le da la bienvenida al usuario.



```
1 from lifestore_file import lifestore_products, lifestore_sales, lifestore_searches
2
3 mensaje_bienvenida = 'Bienvenido al sistema de Lifestore, accede con tus datos de usuario.'
4 print(mensaje_bienvenida)
5 print("")
6
7 intentos=3
8 UsuarioAccedio=False
9 while not UsuarioAccedio:
10     #Ingresa credenciales
11     usuario=input("Usuario: ")
12     contraseña=input("Contraseña: ")
13     if usuario == "hectorpcas" and contraseña == "emtech2022":
14         UsuarioAccedio=True
15     else:
16         if usuario=="hectorpcas":
17             intentos -=1
18             print(f"Contraseña errónea, tiene {intentos} intentos restantes")
19             print("")
20         else:
21             intentos -=1
22             print(f"El usuario '{usuario}' no existe, tiene {intentos} intentos restantes")
23             print("")
24     if intentos == 0:
25         print("Límite de intentos excedido. Acceso denegado.")
26         exit()
27 print("")
28 print(f"Bienvenido, {usuario}")
29 print("")
30
31 # Se contabilizan las unidades vendidas por producto que no fueron devueltas
```



```
Bienvenido al sistema de Lifestore, accede con tus datos de usuario.
Usuario: hectorpcas
Contraseña: emtech2023
Contraseña errónea, tiene 2 intentos restantes

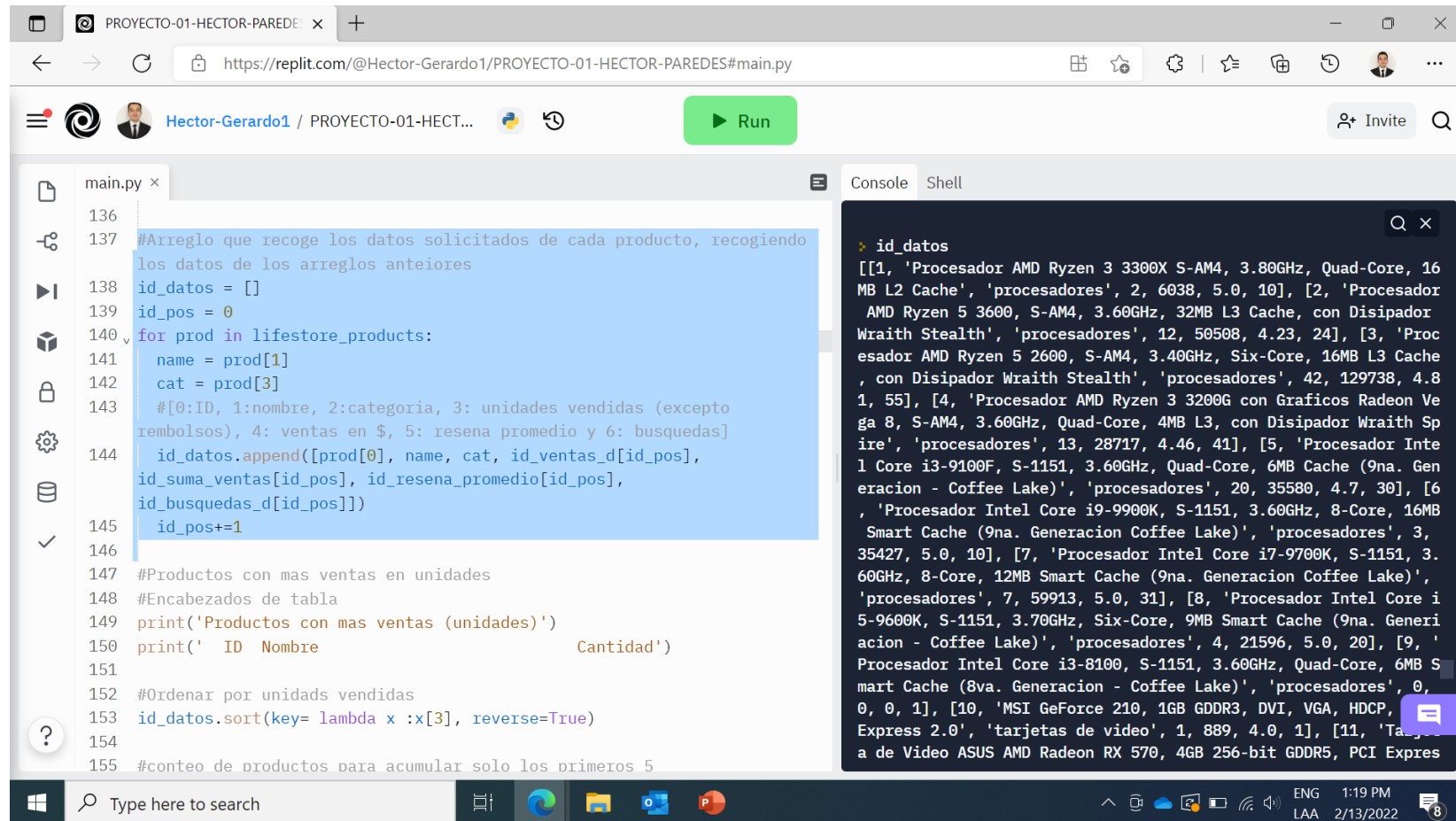
Usuario: hectorpcas
Contraseña: emtech2022

Bienvenido, hectorpcas
```

# Objetivo del código

El primer objetivo del Código fue fomar una lista de listas (id\_datos) que recoja los siguientes datos para cada producto:

- ID
- Nombre
- Categoría
- Unidades vendidas
- Ventas totales (\$)
- Reseña promedio
- Búsquedas totales



The screenshot shows a web browser window with the URL `https://replit.com/@Hector-Gerardo1/PROYECTO-01-HECTOR-PAREDES#main.py`. The editor displays a Python script in `main.py` with the following code:

```
136
137 #Arreglo que recoge los datos solicitados de cada producto, recogiendo
    los datos de los arreglos anteriores
138 id_datos = []
139 id_pos = 0
140 for prod in lifestore_products:
141     name = prod[1]
142     cat = prod[3]
143     #[0:ID, 1:nombre, 2:categoría, 3: unidades vendidas (excepto
    reembolsos), 4: ventas en $, 5: resena promedio y 6: búsquedas]
144     id_datos.append([prod[0], name, cat, id_ventas_d[id_pos],
    id_suma_ventas[id_pos], id_resena_promedio[id_pos],
    id_búsquedas_d[id_pos]])
145     id_pos+=1
146
147 #Productos con mas ventas en unidades
148 #Encabezados de tabla
149 print('Productos con mas ventas (unidades)')
150 print(' ID Nombre Cantidad')
151
152 #Ordenar por unidades vendidas
153 id_datos.sort(key= lambda x :x[3], reverse=True)
154
155 #conteo de productos para acumular solo los primeros 5
```

The console output shows the `id_datos` list, which contains product information as nested lists. The first few elements are:

```
> id_datos
[[1, 'Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16
MB L2 Cache', 'procesadores', 2, 6038, 5.0, 10], [2, 'Procesador
AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador
Wraith Stealth', 'procesadores', 12, 50508, 4.23, 24], [3, 'Proc
esador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache
, con Disipador Wraith Stealth', 'procesadores', 42, 129738, 4.8
1, 55], [4, 'Procesador AMD Ryzen 3 3200G con Graficos Radeon Ve
ga 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Sp
ire', 'procesadores', 13, 28717, 4.46, 41], [5, 'Procesador Inte
l Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Gen
eracion - Coffee Lake)', 'procesadores', 20, 35580, 4.7, 30], [6
, 'Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB
Smart Cache (9na. Generacion Coffee Lake)', 'procesadores', 3,
35427, 5.0, 10], [7, 'Procesador Intel Core i7-9700K, S-1151, 3.
60GHz, 8-Core, 12MB Smart Cache (9na. Generacion Coffee Lake)',
'procesadores', 7, 59913, 5.0, 31], [8, 'Procesador Intel Core i
5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generi
acion - Coffee Lake)', 'procesadores', 4, 21596, 5.0, 20], [9, '
Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB S
mart Cache (8va. Generacion - Coffee Lake)', 'procesadores', 0,
0, 0, 1], [10, 'MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP,
Express 2.0', 'tarjetas de video', 1, 889, 4.0, 1], [11, 'Tarj
eta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Expres
```

# Unidades vendidas (id\_ventas)

Se realiza un conteo por product y por venta, acudiendo a las listas `lifestore_products` y `lifestore_sales`. Sumando a cada product una unidad por cada venta registrada sin devolución.

Los productos que no registren ventas se rellenan con 0.

Se obtiene una lista con ID de product y unidades vendidas.

Se repite proceso para contabilizar todas las ventas incluyendo las devoluciones.

The screenshot shows a Replit Python environment with a file named `main.py` and a console output. The script processes `lifestore_products` and `lifestore_sales` to calculate units sold per product, including returns.

```
main.py x
30
31 # Se contabilizan las unidades vendidas por producto que no fuero
    devueltas
32 id_ventas = []
33 for prod in lifestore_products:
34     id_prod = prod[0]
35     sub = [id_prod, 0]
36     id_ventas.append(sub)
37
38 # Se obtiene Id y unidades vendidas
39 for sale in lifestore_sales:
40     id_prod = sale[1]
41     indice = id_prod - 1
42     if sale[-1] == 1:
43         continue
44     id_ventas[indice][1] += 1
45
46 # Unicamente reúne los valores de unidades vendidas por id
47 id_ventas_d = []
48 for uds in id_ventas:
49     id_ventas_d.append(uds[1])
50
51 # Se repite proceso de contabilizar unidades vendidas. Se consideran
    tambien las unidades devueltas. Esto se usa para promediar reseñas de
```

Console output:

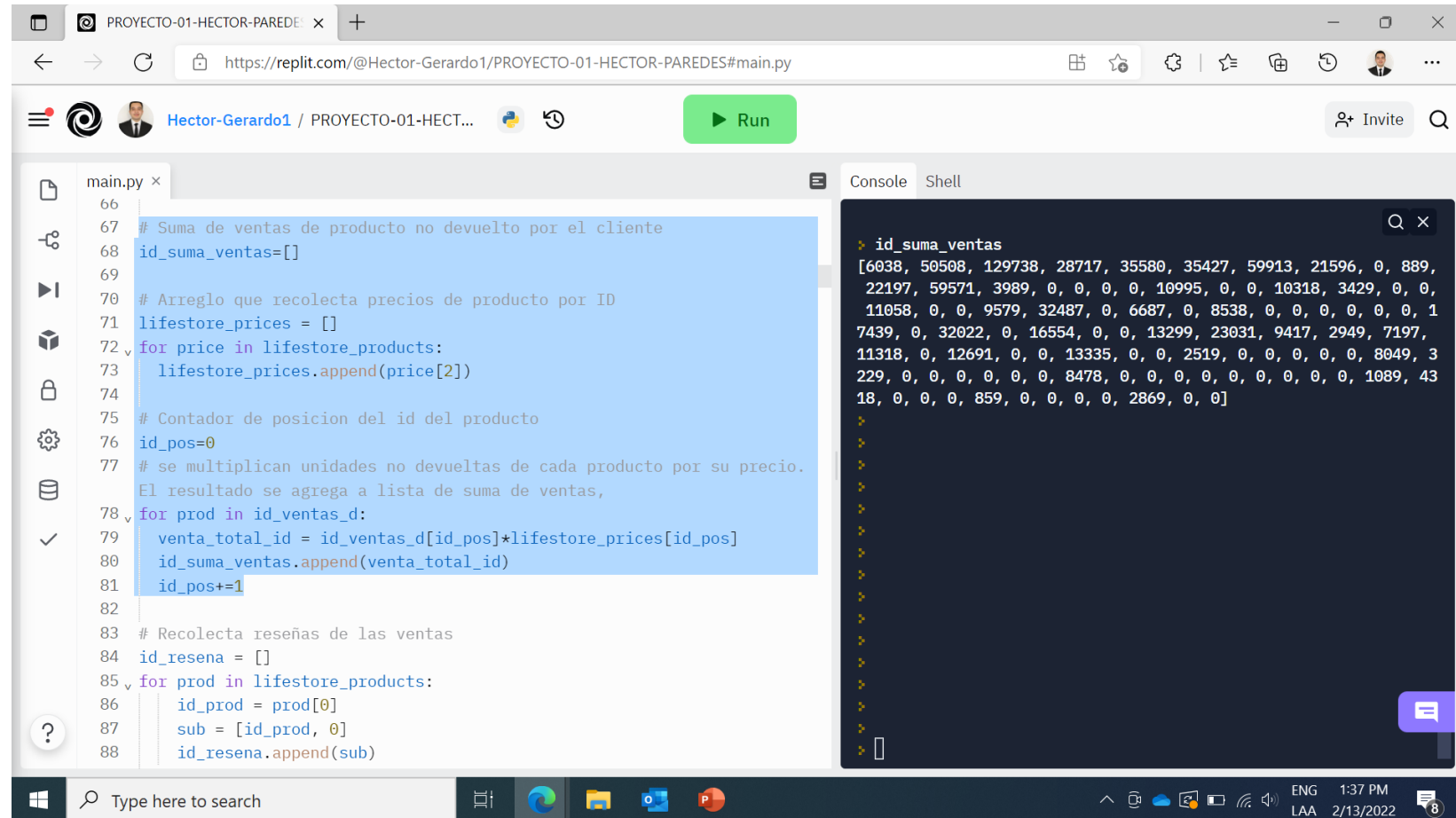
```
> id_ventas
[[1, 2], [2, 12], [3, 42], [4, 13], [5, 20], [6, 3], [7, 7], [8,
4], [9, 0], [10, 1], [11, 3], [12, 9], [13, 1], [14, 0], [15, 0],
[16, 0], [17, 0], [18, 5], [19, 0], [20, 0], [21, 2], [22, 1],
[23, 0], [24, 0], [25, 2], [26, 0], [27, 0], [28, 1], [29, 13],
[30, 0], [31, 3], [32, 0], [33, 2], [34, 0], [35, 0], [36, 0],
[37, 0], [38, 0], [39, 0], [40, 1], [41, 0], [42, 18], [43, 0],
[44, 6], [45, 0], [46, 0], [47, 11], [48, 9], [49, 3], [50, 1],
[51, 3], [52, 2], [53, 0], [54, 49], [55, 0], [56, 0], [57, 15],
[58, 0], [59, 0], [60, 1], [61, 0], [62, 0], [63, 0], [64, 0],
[65, 0], [66, 1], [67, 1], [68, 0], [69, 0], [70, 0], [71, 0],
[72, 0], [73, 0], [74, 2], [75, 0], [76, 0], [77, 0], [78, 0], [
79, 0], [80, 0], [81, 0], [82, 0], [83, 0], [84, 1], [85, 2], [8
6, 0], [87, 0], [88, 0], [89, 1], [90, 0], [91, 0], [92, 0], [93
, 0], [94, 1], [95, 0], [96, 0]]
>
```

Windows taskbar at the bottom shows the time as 1:22 PM on 2/13/2022.



# Ventas totales (id\_suma\_ventas)

Se multiplican las unidades vendidas (sin productos devueltos) por su precio, y se obtiene las ventas totales en \$ de cada producto.



The screenshot shows a web-based Python IDE (Replit) with a file named `main.py` and a console window. The code in `main.py` calculates the total sales for each product ID by multiplying the number of units sold by the price per unit. The console output displays the resulting list of total sales for each product ID.

```
66
67 # Suma de ventas de producto no devuelto por el cliente
68 id_suma_ventas=[]
69
70 # Arreglo que recolecta precios de producto por ID
71 lifestore_prices = []
72 for price in lifestore_products:
73     lifestore_prices.append(price[2])
74
75 # Contador de posicion del id del producto
76 id_pos=0
77 # se multiplican unidades no devueltas de cada producto por su precio.
  El resultado se agrega a lista de suma de ventas,
78 for prod in id_ventas_d:
79     venta_total_id = id_ventas_d[id_pos]*lifestore_prices[id_pos]
80     id_suma_ventas.append(venta_total_id)
81     id_pos+=1
82
83 # Recolecta reseñas de las ventas
84 id_resena = []
85 for prod in lifestore_products:
86     id_prod = prod[0]
87     sub = [id_prod, 0]
88     id_resena.append(sub)
```

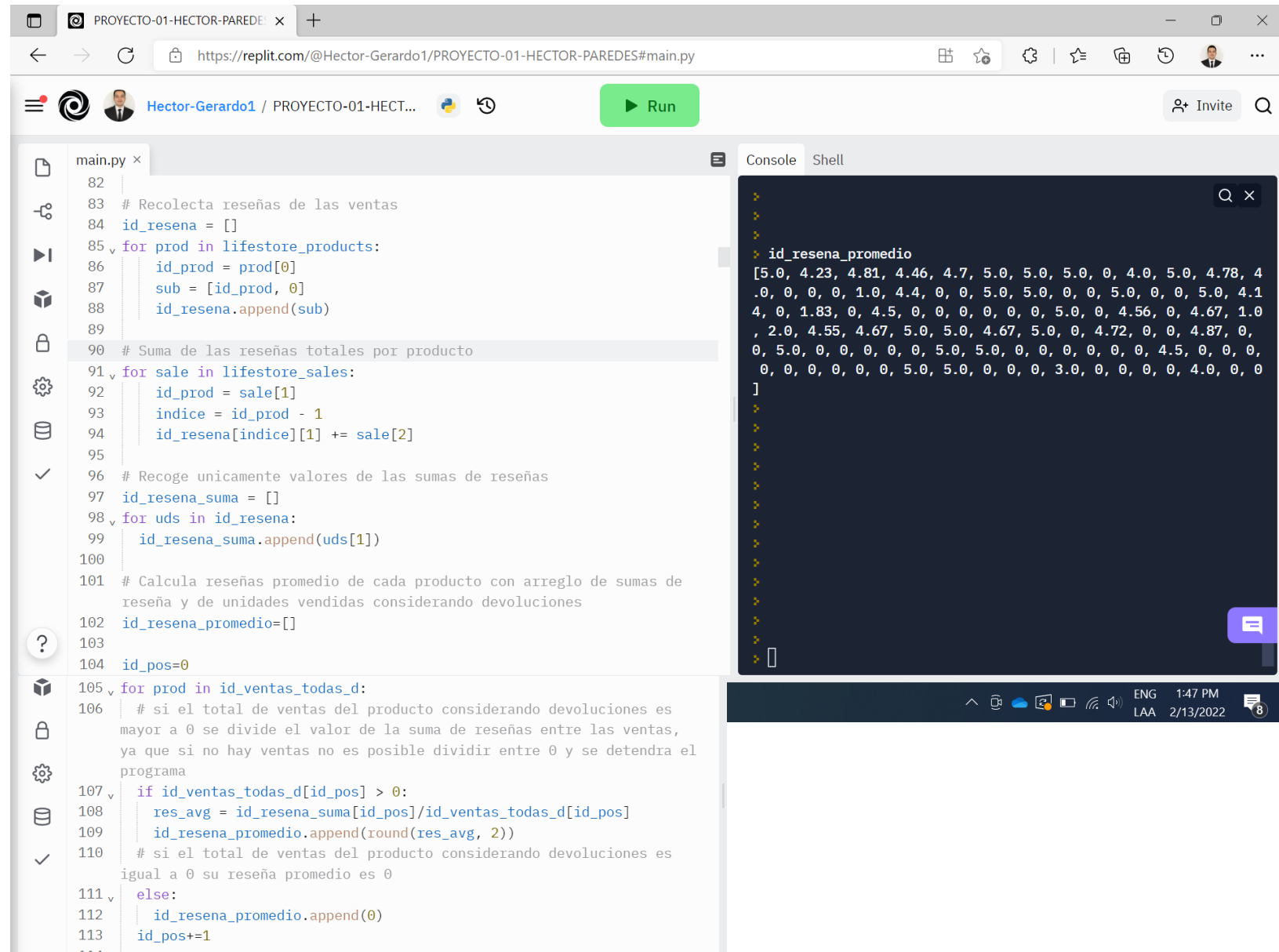
Console Output:

```
> id_suma_ventas
[6038, 50508, 129738, 28717, 35580, 35427, 59913, 21596, 0, 889,
 22197, 59571, 3989, 0, 0, 0, 10995, 0, 0, 10318, 3429, 0, 0,
 11058, 0, 0, 9579, 32487, 0, 6687, 0, 8538, 0, 0, 0, 0, 0, 1
 7439, 0, 32022, 0, 16554, 0, 0, 13299, 23031, 9417, 2949, 7197,
 11318, 0, 12691, 0, 0, 13335, 0, 0, 2519, 0, 0, 0, 0, 0, 8049, 3
 229, 0, 0, 0, 0, 0, 8478, 0, 0, 0, 0, 0, 0, 0, 1089, 43
 18, 0, 0, 0, 859, 0, 0, 0, 2869, 0, 0]
```

# Reseña (id\_resena\_promedio)

En el arreglo id\_resena se suman las reseñas de todas las ventas, incluyendo las que tuvieron devolución.

Después, se recurre a la lista de ventas por cada producto incluyendo devolución para dividir la suma de reseñas entre ventas para obtener reseña promedio de cada producto,



```
main.py x
82
83 # Recolecta reseñas de las ventas
84 id_resena = []
85 for prod in lifestore_products:
86     id_prod = prod[0]
87     sub = [id_prod, 0]
88     id_resena.append(sub)
89
90 # Suma de las reseñas totales por producto
91 for sale in lifestore_sales:
92     id_prod = sale[1]
93     indice = id_prod - 1
94     id_resena[indice][1] += sale[2]
95
96 # Recoge unicamente valores de las sumas de reseñas
97 id_resena_suma = []
98 for uds in id_resena:
99     id_resena_suma.append(uds[1])
100
101 # Calcula reseñas promedio de cada producto con arreglo de sumas de
    reseña y de unidades vendidas considerando devoluciones
102 id_resena_promedio=[]
103
104 id_pos=0
105 for prod in id_ventas_todas_d:
106     # si el total de ventas del producto considerando devoluciones es
    mayor a 0 se divide el valor de la suma de reseñas entre las ventas,
    ya que si no hay ventas no es posible dividir entre 0 y se detendra el
    programa
107     if id_ventas_todas_d[id_pos] > 0:
108         res_avg = id_resena_suma[id_pos]/id_ventas_todas_d[id_pos]
109         id_resena_promedio.append(round(res_avg, 2))
110     # si el total de ventas del producto considerando devoluciones es
    igual a 0 su reseña promedio es 0
111     else:
112         id_resena_promedio.append(0)
113     id_pos+=1
114
```

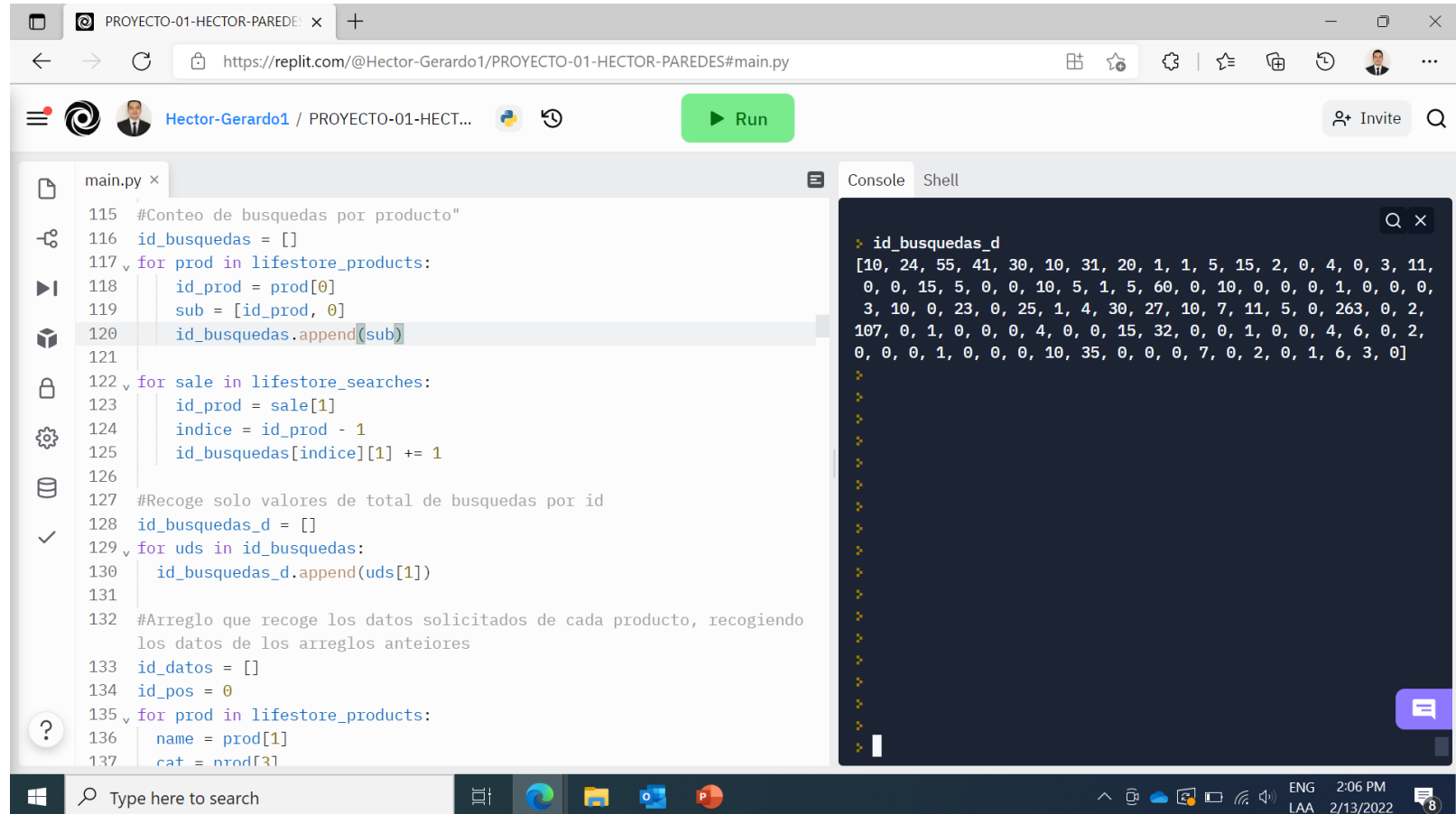
Console

```
> id_resena_promedio
[5.0, 4.23, 4.81, 4.46, 4.7, 5.0, 5.0, 5.0, 0, 4.0, 5.0, 4.78, 4
.0, 0, 0, 0, 1.0, 4.4, 0, 0, 5.0, 5.0, 0, 0, 5.0, 0, 0, 5.0, 4.1
4, 0, 1.83, 0, 4.5, 0, 0, 0, 0, 0, 5.0, 0, 4.56, 0, 4.67, 1.0
, 2.0, 4.55, 4.67, 5.0, 5.0, 4.67, 5.0, 0, 4.72, 0, 0, 4.87, 0,
0, 5.0, 0, 0, 0, 0, 5.0, 5.0, 0, 0, 0, 0, 4.5, 0, 0, 0,
0, 0, 0, 0, 0, 5.0, 5.0, 0, 0, 0, 3.0, 0, 0, 0, 4.0, 0, 0
]
```

ENG 1:47 PM  
LAA 2/13/2022

# Búsquedas (id\_resena\_promedio)

En el arreglo `id_busquedas` se contabilizan las búsquedas por producto registradas en la lista `lifestore_searches` de acuerdo con su `id`.



The screenshot shows a Replit Python environment with a file named `main.py` and a console window. The script in `main.py` performs the following steps:

- Initializes an empty list `id_busquedas`.
- Iterates over `lifestore_products` to create sub-arrays for each product ID.
- Iterates over `lifestore_searches` to count the number of searches for each product ID.
- Creates a list `id_busquedas_d` containing the total search counts for each product ID.
- Creates a list `id_datos` containing product details.

The console output shows the resulting `id_busquedas_d` list:

```
> id_busquedas_d
[10, 24, 55, 41, 30, 10, 31, 20, 1, 1, 5, 15, 2, 0, 4, 0, 3, 11,
0, 0, 15, 5, 0, 0, 10, 5, 1, 5, 60, 0, 10, 0, 0, 0, 1, 0, 0, 0,
3, 10, 0, 23, 0, 25, 1, 4, 30, 27, 10, 7, 11, 5, 0, 263, 0, 2,
107, 0, 1, 0, 0, 0, 4, 0, 0, 15, 32, 0, 0, 1, 0, 0, 4, 6, 0, 2,
0, 0, 0, 1, 0, 0, 0, 10, 35, 0, 0, 0, 7, 0, 2, 0, 1, 6, 3, 0]
```

# Análisis por categoría

Se obtienen las categorías disponibles a partir de un ciclo for.

A partir de los resultados recolectados en la lista `id_datos`, se filtran los resultados por categoría.

Para cada categoría con un ciclo for se obtienen los productos disponibles y se genera una nueva lista de productos que cumplan con el criterio de igualdad de la categoría.

Se calculan las ventas totales de la categoría y se agregan en una lista con ventas totales de todas las categorías.

Se ordena la nueva lista de productos por categoría del menos vendido al mas vendido.

Se recogen los datos de ventas para obtener los cinco productos menos vendidos por medio de un ciclo while que recolecte los primeros 5 productos en la lista de menos vendidos

The screenshot shows a Replit terminal window with a Python script for category analysis. The script is divided into several sections, each highlighted with a black box and an arrow pointing to a corresponding text block on the left. The script uses a for loop to iterate over categories, a while loop to filter products, and a while loop to find the top 5 products by sales. The output is displayed in the terminal window on the right.

```
39 #Análisis de categorías
40 print("Categorías")
41 #Generamos un arreglo que recolecte las categorías disponibles
42 categories = []
43 for category in lifestore_products:
44     cat = category[3]
45     if cat not in categories:
46         categories.append(cat)
47     print(categories)
48 print("")

49 #Arreglo que almacene las ventas totales de cada categoría
50 ventas_por_categoria = []
51
52 #Cuenta categorías según el arreglo de categorías
53 conteo_cat = 0
54
55 #Se corre un ciclo para analizar cada categoría
56 while conteo_cat < len(categories):
57     #Resetea el orden de la lista de datos del id de producto menor al mayor
58     id_datos.sort(key= lambda x :x[0], reverse=False)
59
60     #Recoge los datos de los productos de la categoría
61     lifestore_products_cat=[prod for prod in id_datos if prod[2]==categories[conteo_cat]]
62     print(f'[{conteo_cat+1}] Categoría: {categories[conteo_cat]} ({len(lifestore_products_cat)} productos)')
63
64     #Recoge los ids de los productos de la categoría
65     ids_cat=[prod[0] for prod in id_datos if prod[2]==categories[conteo_cat]]
66     print('Productos disponibles: ', ids_cat)
67     print("")
68
69     #Recoge las ventas totales de los productos de la categoría
70     lifestore_sales_cat=[prod[4] for prod in id_datos if prod[2]==categories[conteo_cat]]
71     ventas_cat = sum(lifestore_sales_cat)
72     print(f'Ventas totales categoría {categories[conteo_cat]}: ${ventas_cat}')
73     ventas_por_categoria.append([categories[conteo_cat], ventas_cat])
74     print("")
75
76 #Productos menos vendidos por categoría
77 #Ordena los productos de la categoría del menos vendido al mas vendido
78 lifestore_products_cat.sort(key= lambda x :x[3], reverse=False)
79 #Encabezado de tabla
80 print(f'Productos menos vendidos en la categoría {categories[conteo_cat]}')
81 #Encabezado de columnas
82 print(' ID Nombre Cantidad')
83 #Contador de posición en la lista de productos de la categoría
84 pos=0
85
86 for prod in lifestore_products_cat:
87     #Si la categoría tiene mas de 5 artículos, obtener los 5 productos con menos ventas de acuerdo a la lista previamente ordenada
88     if len(ids_cat)>5:
89         while pos < 5:
90             name = lifestore_products_cat[pos][1]
91             name = name[:30]
92             print(f'{pos+1} {lifestore_products_cat[pos][0]} {name} ({lifestore_products_cat[pos][3]} uds)')
93             pos+=1
94     #Si la categoría tiene menos de 5 artículos, obtener todos los artículos del menos al mas vendido
95     else:
96         name = lifestore_products_cat[pos][1]
97         name = name[:30]
98         print(f'{pos+1} {lifestore_products_cat[pos][0]} {name} ({lifestore_products_cat[pos][3]} uds)')
99         pos+=1
100
101 print("")
102 #Productos menos buscados por categoría
103 #Ordena los productos de la categoría del menos buscado al mas buscado
104 lifestore_products_cat.sort(key= lambda x :x[6], reverse=False)
105 #Encabezado de tabla
106 print(f'Productos menos buscados en la categoría {categories[conteo_cat]}')
107 #Encabezado de columnas
108 print(' ID Nombre Busquedas')
109 #Contador de posición en la lista de productos de la categoría
110 pos=0
111
112 #Para cada producto de la categoría
113 for prod in lifestore_products_cat:
114     #Si la categoría tiene mas de 10 artículos, obtener los 5 productos con menos ventas de acuerdo a la lista previamente ordenada
115     if len(ids_cat)>10:
116         while pos < 5:
117             name = lifestore_products_cat[pos][1]
118             name = name[:30]
119             print(f'{pos+1} {lifestore_products_cat[pos][0]} {name} ({lifestore_products_cat[pos][3]} uds)')
120             pos+=1
121     #Si la categoría tiene menos de 10 artículos, obtener todos los artículos del menos al mas vendido
122     else:
123         name = lifestore_products_cat[pos][1]
124         name = name[:30]
125         print(f'{pos+1} {lifestore_products_cat[pos][0]} {name} ({lifestore_products_cat[pos][3]} uds)')
126         pos+=1
```

**Categorías**

['procesadores', 'tarjetas de video', 'tarjetas madre', 'discos duros', 'memorias usb', 'pantallas', 'bocinas', 'audifonos']

1 Categoría: procesadores (9 productos)  
Productos disponibles: [1, 2, 3, 4, 5, 6, 7, 8, 9]

Ventas totales categoría procesadores: \$367517

Productos menos vendidos en la categoría procesadores

ID	Nombre	Cantidad
1	9 Procesador Intel Core i3-9100, (8 uds)	8
2	1 Procesador AMD Ryzen 3 3200X 5 (2 uds)	2
3	6 Procesador Intel Core i9-9900K (3 uds)	3
4	8 Procesador Intel Core i5-9600K 4	4
5	2 Procesador AMD Ryzen 5 3600, 5	12
6	6 Procesador Intel Core i3-9100F 20	20
7	7 Procesador Intel Core i7-9700K 7	7
8	4 Procesador AMD Ryzen 3 3200G c 13	13
9	3 Procesador AMD Ryzen 5 2600, 5	42

2 Categoría: tarjetas de video (19 productos)  
Productos disponibles: [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]

Ventas totales categoría tarjetas de video: \$132825

Productos menos vendidos en la categoría tarjetas de video

ID	Nombre	Cantidad
1	14 Tarjeta de Video EVGA NVIDIA G (8 uds)	8
2	15 Tarjeta de Video EVGA NVIDIA G (8 uds)	8
3	16 Tarjeta de Video EVGA NVIDIA G (8 uds)	8
4	17 Tarjeta de Video Gigabyte AMD (8 uds)	8
5	19 Tarjeta de Video Gigabyte NVID (8 uds)	8

Productos menos buscados en la categoría tarjetas de video

ID	Nombre	Busquedas
1	14 Tarjeta de Video EVGA NVIDIA G (8 uds)	8
2	16 Tarjeta de Video EVGA NVIDIA G (8 uds)	8
3	19 Tarjeta de Video Gigabyte NVID (8 uds)	8
4	20 Tarjeta de Video Gigabyte NVID (8 uds)	8
5	23 Tarjeta de Video MSI Radeon X1 (8 uds)	8
6	24 Tarjeta de Video PNY NVIDIA Ge (8 uds)	8
7	27 Tarjeta de Video Visioner AMD (8 uds)	8
8	10 MSI Geforce 210, 1GB GDDR3, DV (1 uds)	1
9	13 Tarjeta de Video Asus NVIDIA G (1 uds)	1
10	17 Tarjeta de Video Gigabyte AMD (8 uds)	8

Ventas totales por categoría

Categoría	Ventas(\$)
1 procesadores	\$367517
2 tarjetas de video	\$132825
3 tarjetas madre	\$113727
4 discos duros	\$93237
5 pantallas	\$11278
6 audifonos	\$9135
7 bocinas	\$8478
8 memorias usb	\$2519

# Análisis por categoría

Enseguida se ordena la lista de productos de la categoría por el criterio del menos buscado al mas buscado.

Se recogen los datos de búsquedas para obtener los diez productos menos vendidos por medio de un ciclo while que recolecte los primeros 5 productos en la lista de menos buscados.

The screenshot displays a Replit environment with a Python script named `main.py` and its execution output in the console. The script performs an analysis of products categorized by 'procesadores' (processors) and 'tarjetas de video' (video cards). It sorts products by sales and searches to identify the top 5 least sold items in each category.

```
main.py
239 #Análisis de categorías
240 print("Categorías")
241 #Generamos un arreglo que recolece las categorías disponibles
242 categorias = []
243 #Por categoría en lifestore_products:
244 cat = categorias[3]
245 if cat not in categorias:
246     categorias.append(cat)
247 print(categorias)
248 print("")
249
250 #Arreglo que almacena las ventas totales de cada categoría
251 ventas_por_categoria=[]
252
253 #Cuenta categorías según el arreglo de categorías
254 conteo_cat=0
255
256 #Se corre un ciclo para analizar cada categoría
257 while conteo_cat < len(categorias):
258     #Resetea el orden de la lista de datos del id de producto menor al mayor
259     id_datos.sort(key= lambda x: x[0], reverse=False)
260
261     #Recolecta los datos de los productos de la categoría
262     lifestore_products_cat=[prod for prod in id_datos if prod[2]==categorias[conteo_cat]]
263     print(f'[{categorias[conteo_cat]}] ({len(lifestore_products_cat)} productos)')
264
265     #Recolecta los ids de los productos de la categoría
266     ids_cat=[prod[0] for prod in id_datos if prod[2]==categorias[conteo_cat]]
267     print('Productos disponibles: ', ids_cat)
268     print('')
269
270     #Recolecta las ventas totales de los productos de la categoría
271     lifestore_sales_cat=[prod[4] for prod in id_datos if prod[2]==categorias[conteo_cat]]
272     ventas_cat = sum(lifestore_sales_cat)
273     print(f'Ventas totales categoría {categorias[conteo_cat]}: ${ventas_cat}')
274     ventas_por_categoria.append([categorias[conteo_cat], ventas_cat])
275     print('')
276
277     #Productos menos vendidos por categoría
278     #Ordena los productos de la categoría del menos vendido al mas vendido
279     lifestore_products_cat.sort(key= lambda x: x[3], reverse=False)
280     #Encabezado de tabla
281     print(f'Productos menos vendidos en la categoría {categorias[conteo_cat]}')
282     #Encabezado de columnas
283     print(' ID Nombre Cantidad')
284     #Contador de posición en la lista de productos de la categoría
285     pos=0
286     #Para cada producto de la categoría
287     for prod in lifestore_products_cat:
288         #Si la categoría tiene mas de 5 artículos, obtener los 5 productos con menos ventas de acuerdo a la lista previamente ordenada
289         if len(ids_cat)>5:
290             while pos < 5:
291                 name = lifestore_products_cat[pos][1]
292                 name = name[:30]
293                 print(f'{pos+1} {lifestore_products_cat[pos][0]} {name} ({lifestore_products_cat[pos][3]} uds)')
294                 pos+=1
295         #Si la categoría tiene menos de 5 artículos, obtener todos los artículos del menos al mas vendido
296         else:
297             name = lifestore_products_cat[pos][1]
298             name = name[:30]
299             print(f'{pos+1} {lifestore_products_cat[pos][0]} {name} ({lifestore_products_cat[pos][3]} uds)')
300             pos+=1
301
302     print('')
303     #Productos menos buscados por categoría
304     #Ordena los productos de la categoría del menos buscado al mas buscado
305     lifestore_products_cat.sort(key= lambda x: x[6], reverse=False)
306     #Encabezado de tabla
307
308     #Encabezado de columnas
309     print(' ID Nombre Busquedas')
310     #Contador de posición en la lista de productos de la categoría
311     pos=0
312     #Para cada producto de la categoría
313     for prod in lifestore_products_cat:
314         #Si la categoría tiene mas de 10 artículos, obtener los 5 productos con menos ventas de acuerdo a la lista previamente ordenada
315         if len(ids_cat)>10:
316             while pos < 5:
317                 name = lifestore_products_cat[pos][1]
318                 name = name[:30]
319                 print(f'{pos+1} {lifestore_products_cat[pos][0]} {name} ({lifestore_products_cat[pos][3]} uds)')
320                 pos+=1
321         #Si la categoría tiene menos de 10 artículos, obtener todos los artículos del menos al mas vendido
322         else:
323             name = lifestore_products_cat[pos][1]
324             name = name[:30]
325             print(f'{pos+1} {lifestore_products_cat[pos][0]} {name} ({lifestore_products_cat[pos][3]} uds)')
326             pos+=1
```

The console output shows the following results:

```
Categorías
['procesadores', 'tarjetas de video', 'tarjetas madre', 'discos duros', 'memorias usb', 'pantallas', 'bocinas', 'audifonos']

1 Categoría: procesadores (9 productos)
Productos disponibles: [1, 2, 3, 4, 5, 6, 7, 8, 9]

Ventas totales categoría procesadores: $367517

Productos menos vendidos en la categoría procesadores
ID Nombre Cantidad
1 9 Procesador Intel Core i3-3100, (8 uds)
2 1 Procesador AMD Ryzen 3 3200X, (2 uds)
3 6 Procesador Intel Core i9-9900K, (3 uds)
4 8 Procesador Intel Core i5-9600K, (4 uds)
5 2 Procesador AMD Ryzen 5 3600, (5 uds)
6 6 Procesador Intel Core i3-9100F, (6 uds)
7 7 Procesador Intel Core i7-9700K, (7 uds)
8 4 Procesador AMD Ryzen 3 3200G, (4 uds)
9 3 Procesador AMD Ryzen 5 2600, (5 uds)

2 Categoría: tarjetas de video (19 productos)
Productos disponibles: [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]

Ventas totales categoría tarjetas de video: $112825

Productos menos vendidos en la categoría tarjetas de video
ID Nombre Cantidad
1 14 Tarjeta de Video EVGA NVIDIA G, (8 uds)
2 15 Tarjeta de Video EVGA NVIDIA G, (8 uds)
3 16 Tarjeta de Video EVGA NVIDIA G, (8 uds)
4 17 Tarjeta de Video Gigabyte AMD, (8 uds)
5 19 Tarjeta de Video Gigabyte NVID, (8 uds)

Productos menos buscados en la categoría tarjetas de video
ID Nombre Busquedas
1 14 Tarjeta de Video EVGA NVIDIA G, 0
2 16 Tarjeta de Video EVGA NVIDIA G, 0
3 19 Tarjeta de Video Gigabyte NVID, 0
4 20 Tarjeta de Video Gigabyte NVID, 0
5 23 Tarjeta de Video MSI Radeon X1, 0
6 24 Tarjeta de Video PNY NVIDIA Ge, 0
7 27 Tarjeta de Video Visioner AMD, 0
8 10 MSI Geforce 210, 1GB GDDR3, DV, 1
9 13 Tarjeta de Video Asus NVIDIA G, 1
10 15 Tarjeta de Video EVGA NVIDIA G, 0

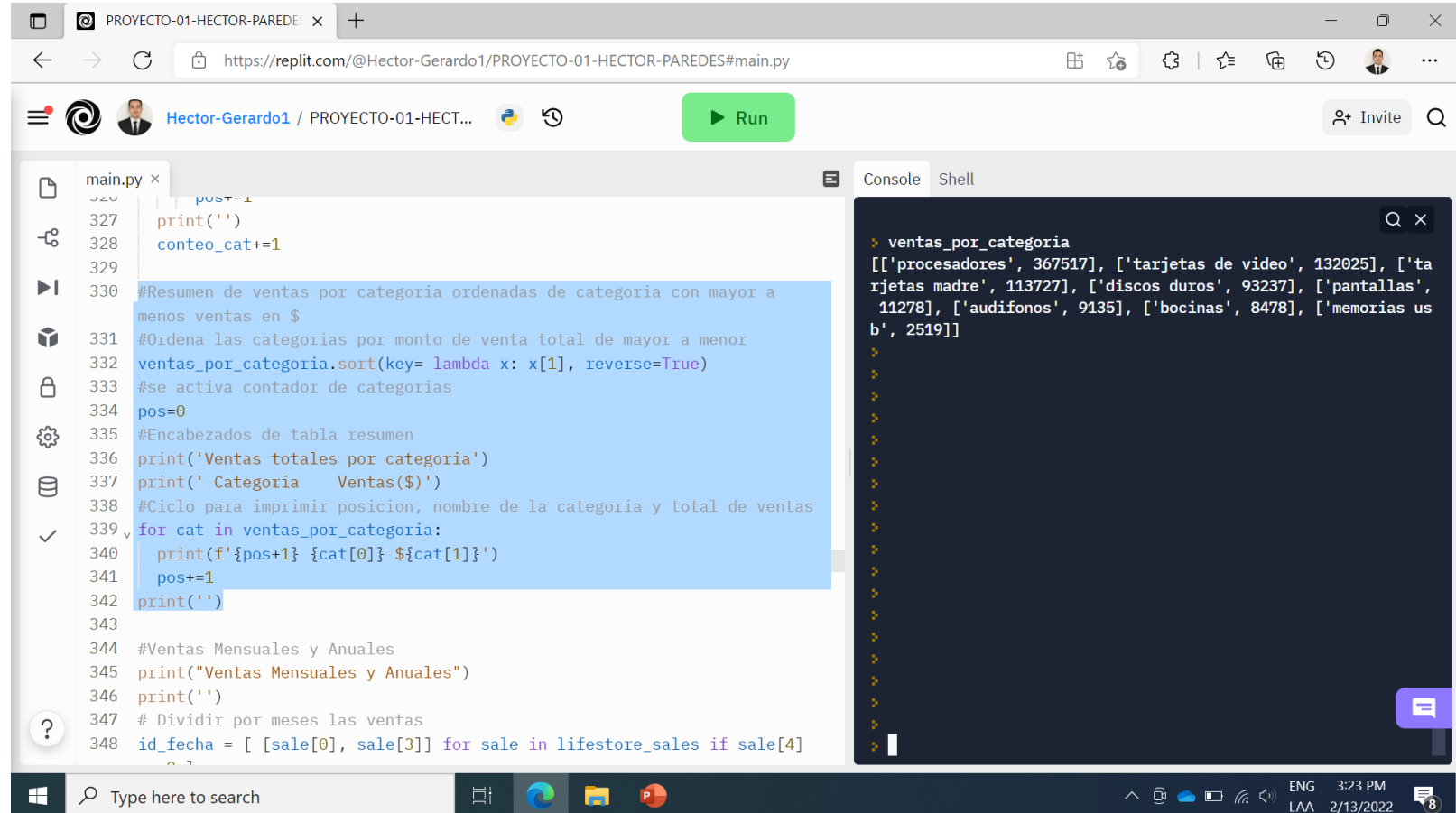
Ventas totales por categoría
Categoría Ventas($)
1 procesadores $367517
2 tarjetas de video $112825
3 tarjetas madre $113727
4 discos duros $93237
5 pantallas $11278
6 audifonos $9135
7 bocinas $8478
8 memorias usb $2519
```

# Análisis por categoría (ventas\_por\_categoria)

Se recolectan las ventas totales por categoría sumadas en el ciclo anterior de análisis de categorías.

Se ordena la lista de la categoría con más ventas a la de menos ventas.

Se imprime cada valor de la lista (lugar, nombre y ventas (\$)).



The screenshot shows a Replit environment with a Python script named `main.py` and its output in the console.

**Script Code (main.py):**

```
326     post+=1
327     print('')
328     conteo_cat+=1
329
330 #Resumen de ventas por categoria ordenadas de categoria con mayor a
    menos ventas en $
331 #Ordena las categorias por monto de venta total de mayor a menor
332 ventas_por_categoria.sort(key= lambda x: x[1], reverse=True)
333 #se activa contador de categorias
334 pos=0
335 #Encabezados de tabla resumen
336 print('Ventas totales por categoria')
337 print(' Categoria    Ventas($)')
338 #Ciclo para imprimir posicion, nombre de la categoria y total de ventas
339 for cat in ventas_por_categoria:
340     print(f'{pos+1} {cat[0]} ${cat[1]}')
341     pos+=1
342 print('')
343
344 #Ventas Mensuales y Anuales
345 print("Ventas Mensuales y Anuales")
346 print('')
347 # Dividir por meses las ventas
348 id_fecha = [ [sale[0], sale[3]] for sale in lifestore_sales if sale[4]
```

**Console Output:**

```
> ventas_por_categoria
[['procesadores', 367517], ['tarjetas de video', 132025], ['ta
rjetas madre', 113727], ['discos duros', 93237], ['pantallas',
11278], ['audifonos', 9135], ['bocinas', 8478], ['memorias us
b', 2519]]
```

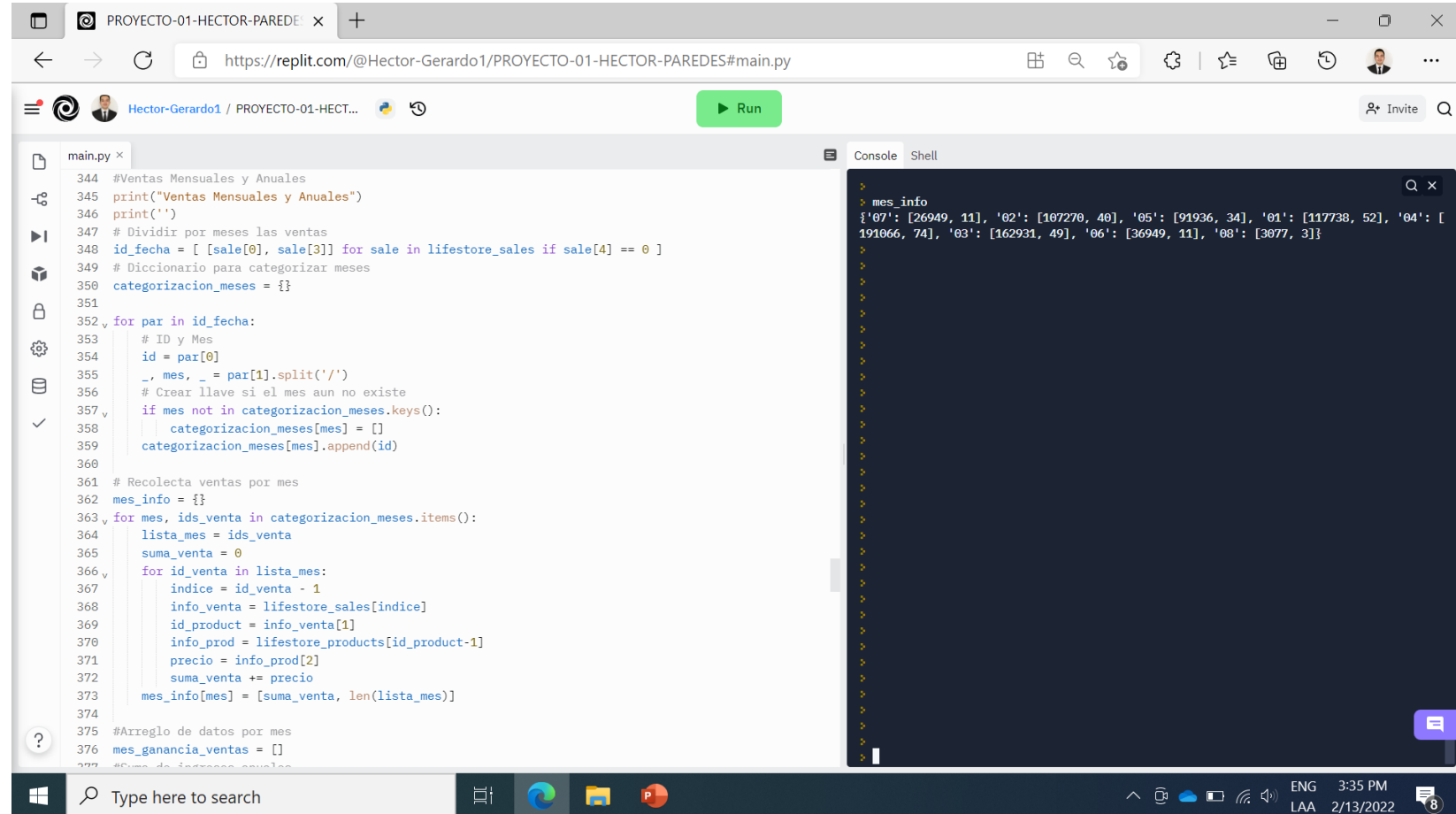
# Análisis por mes

Por medio de las fechas en `lifestore_sales`, se toma el número del mes.

Se genera un diccionario que recolecte los meses donde hay ventas.

Para cada mes, se contabiliza cada venta sin devolución y se suma el precio del producto según su ID y se contabiliza el total de unidades vendidas.

Estos datos se reúnen en un diccionario.



The screenshot shows a Replit IDE window titled "PROYECTO-01-HECTOR-PAREDES" with a file named "main.py". The code processes sales data from a dictionary `lifestore_sales` and a product dictionary `lifestore_products`. It iterates through sales, categorizes them by month, and calculates the total sales and number of units for each month. The console output shows the resulting `mes_info` dictionary.

```
344 #Ventas Mensuales y Anuales
345 print("Ventas Mensuales y Anuales")
346 print('')
347 # Dividir por meses las ventas
348 id_fecha = [ [sale[0], sale[3]] for sale in lifestore_sales if sale[4] == 0 ]
349 # Diccionario para categorizar meses
350 categorizacion_meses = {}
351
352 for par in id_fecha:
353     # ID y Mes
354     id = par[0]
355     _, mes, _ = par[1].split('/')
356     # Crear llave si el mes aun no existe
357     if mes not in categorizacion_meses.keys():
358         categorizacion_meses[mes] = []
359     categorizacion_meses[mes].append(id)
360
361 # Recolecta ventas por mes
362 mes_info = {}
363 for mes, ids_venta in categorizacion_meses.items():
364     lista_mes = ids_venta
365     suma_venta = 0
366     for id_venta in lista_mes:
367         indice = id_venta - 1
368         info_venta = lifestore_sales[indice]
369         id_producto = info_venta[1]
370         info_prod = lifestore_products[id_producto-1]
371         precio = info_prod[2]
372         suma_venta += precio
373     mes_info[mes] = [suma_venta, len(lista_mes)]
374
375 #Arreglo de datos por mes
376 mes_ganancia_ventas = []
377 #Suma de ingresos anuales
```

Console Output:

```
> mes_info
{'07': [26949, 11], '02': [107270, 40], '05': [91936, 34], '01': [117738, 52], '04': [191066, 74], '03': [162931, 49], '06': [36949, 11], '08': [3077, 3]}
```

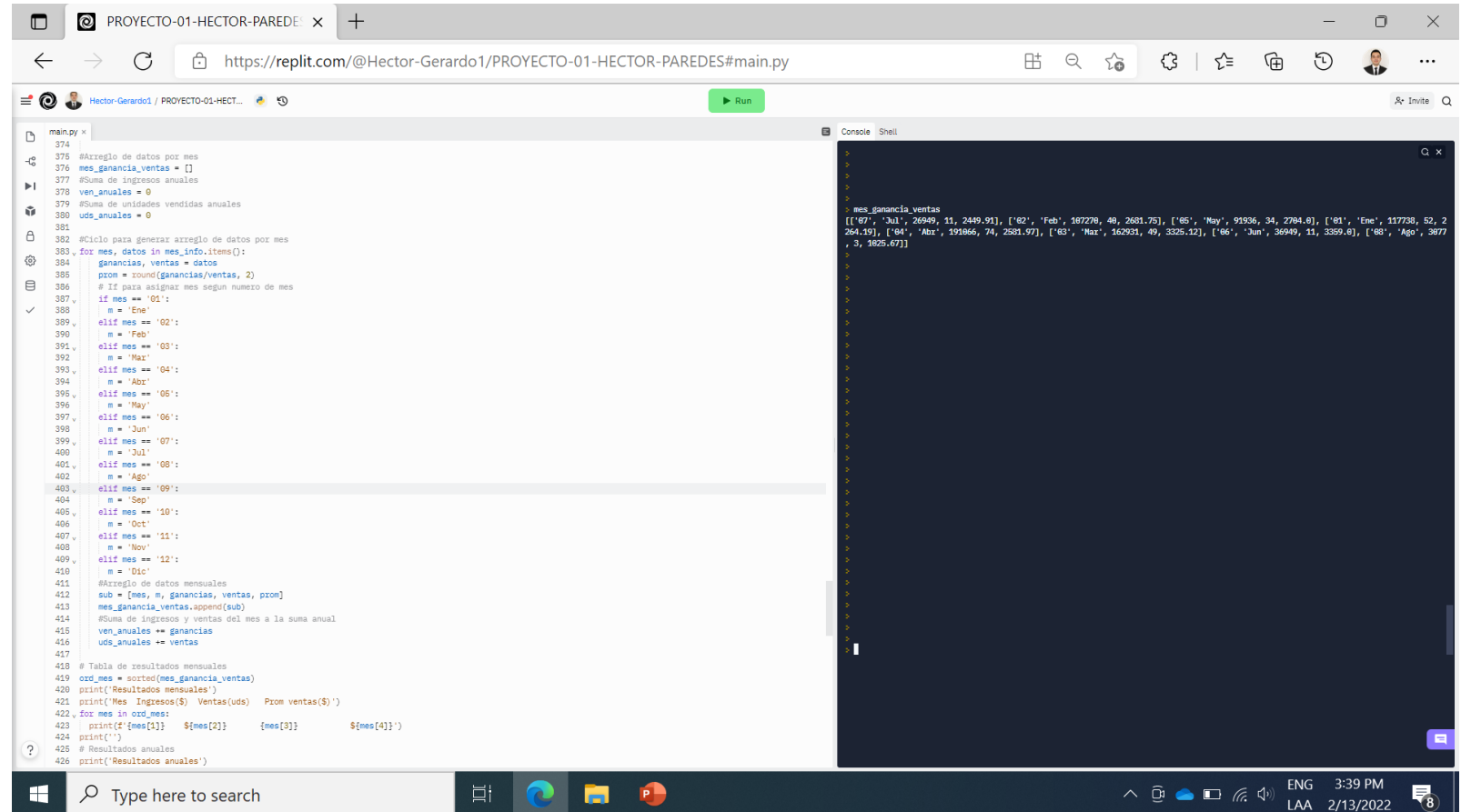


# Análisis por mes

Se toma el diccionario de ventas mensuales para convertirlo en lista.

A la lista, se agrega la abreviatura del mes según su número por medio de un if.

Se calcula además las ventas promedio por mes dividiendo ventas mensuales entre unidades vendidas.



The screenshot shows a web browser window with a Replit account. The URL is <https://replit.com/@Hector-Gerardo1/PROYECTO-01-HECTOR-PAREDES#main.py>. The editor displays a Python script named `main.py` with the following content:

```
374
375 #Arreglo de datos por mes
376 mes_ganancia_ventas = []
377 #Suma de ingresos anuales
378 ven_anuales = 0
379 #Suma de unidades vendidas anuales
380 uds_anuales = 0
381
382 #Ciclo para generar arreglo de datos por mes
383 for mes, datos in mes_info.items():
384     ganancias, ventas = datos
385     prom = round(ganancias/ventas, 2)
386     # If para asignar mes segun numero de mes
387     if mes == '01':
388         m = 'Ene'
389     elif mes == '02':
390         m = 'Feb'
391     elif mes == '03':
392         m = 'Mar'
393     elif mes == '04':
394         m = 'Abr'
395     elif mes == '05':
396         m = 'May'
397     elif mes == '06':
398         m = 'Jun'
399     elif mes == '07':
400         m = 'Jul'
401     elif mes == '08':
402         m = 'Ago'
403     elif mes == '09':
404         m = 'Sep'
405     elif mes == '10':
406         m = 'Oct'
407     elif mes == '11':
408         m = 'Nov'
409     elif mes == '12':
410         m = 'Dic'
411     #Arreglo de datos mensuales
412     sub = [mes, m, ganancias, ventas, prom]
413     mes_ganancia_ventas.append(sub)
414     #Suma de ingresos y ventas del mes a la suma anual
415     ven_anuales += ganancias
416     uds_anuales += ventas
417
418 # Tabla de resultados mensuales
419 ord_mes = sorted(mes_ganancia_ventas)
420 print('Resultados mensuales:')
421 print('Mes Ingresos($) Ventas(uds) Prom Ventas($)'')
422 for mes in ord_mes:
423     print(f'{mes[1]:10} {mes[2]:10} {mes[3]:10} {mes[4]:10}')
424 print('')
425 # Resultados anuales
426 print('Resultados anuales')
```

The console output shows the following data for `mes_ganancia_ventas`:

```
mes_ganancia_ventas
[[('01', 'Jul', 26949, 41, 2449.91), ('02', 'Feb', 187270, 40, 2681.75), ('05', 'May', 91936, 34, 2704.6), ('01', 'Ene', 117738, 52, 2264.19), ('04', 'Abr', 191866, 74, 2591.97), ('03', 'Mar', 162931, 49, 3325.12), ('06', 'Jun', 36949, 11, 3359.6), ('08', 'Ago', 3877, 3, 1825.67)]]
```



# Análisis por mes

Se ordena la lista obtenida y se imprimen los resultados por:

- Mes
- Ingresos (\$)
- Ventas (unidades)

```
417
418 # Tabla de resultados mensuales
419 ord_mes = sorted(mes_ganancia_ventas)
420 print('Resultados mensuales')
421 print('Mes Ingresos($) Ventas(uds) Prom ventas($)\n')
422 for mes in ord_mes:
423     print(f'{mes[1]} {mes[2]} {mes[3]} {mes[4]}\n')
424 print('')
425 # Resultados anuales
426 print('Resultados anuales')
427 print(f'Ingresos anuales: {ven_anuales}')
428 print(f'Ventas realizadas anuales: {uds_anuales}')
429 print('')
430
431 # Resultados mensuales ordenados por ingresos mensuales
432 ord_gancia = sorted(mes_ganancia_ventas, key=lambda x: x[2], reverse=True)
433 print('Meses con mas ingresos')
434 print('Mes Ingresos($)\n')
435 for mes in ord_gancia:
436     print(f'{mes[1]} {mes[2]}\n')
437 print('')
438
439 # Resultados mensuales ordenados por ventas realizadas
440 ord_ventas = sorted(mes_ganancia_ventas, key=lambda x: x[3], reverse=True)
441 print('Meses con mas ventas')
442 print('Mes Ventas\n')
443 for mes in ord_ventas:
444     print(f'{mes[1]} {mes[3]}\n')
445 print('')
446
447 print('Informacion adicional')
448 #Calcula el dinero devuelto a clientes
449 print('Devoluciones')
450 id_pos = 0
451 id_dev = []
452 for prod in id_ventas:
453     #Calcula devoluciones restando de todas las ventas las que no tuvieron devoluciones
454     dev = id_ventas_todas[id_pos]-id_ventas_d[id_pos]
455     #acortar nombre del producto
456     name = lifetore_products[id_pos][1]
457     name = name[:30]
458     #en la lista se recoge 0: id del producto, 1 nombre, 2 unidades devueltas, 3 precio del producto
459     id_dev.append([id_ventas[id_pos][0], name, dev, lifetore_prices[id_pos]])
460     id_pos+=1
461 id_dev.sort(key= lambda x : x[2], reverse=True)
462
```

Ventas Mensuales y Anuales

Resultados mensuales

Mes	Ingresos(\$)	Ventas(uds)	Prom ventas(\$)
Ene	\$117738	52	\$2264.19
Feb	\$167278	48	\$2661.75
Mar	\$162931	49	\$3325.12
Abr	\$191866	74	\$2581.97
May	\$91936	34	\$2704.8
Jun	\$36949	11	\$3359.8
Jul	\$26949	11	\$2449.91
Ago	\$3877	3	\$1625.67

Resultados anuales

Ingresos anuales: \$737916

Ventas realizadas anuales: 274

Meses con mas ingresos

Mes	Ingresos(\$)
Abr	\$191866
Mar	\$162931
Ene	\$117738
Feb	\$167278
May	\$91936
Jun	\$36949
Jul	\$26949
Ago	\$3877

Meses con mas ventas

Mes	Ventas
Abr	74
Mar	49
Ene	52
Feb	48
May	34
Jun	11
Jul	11

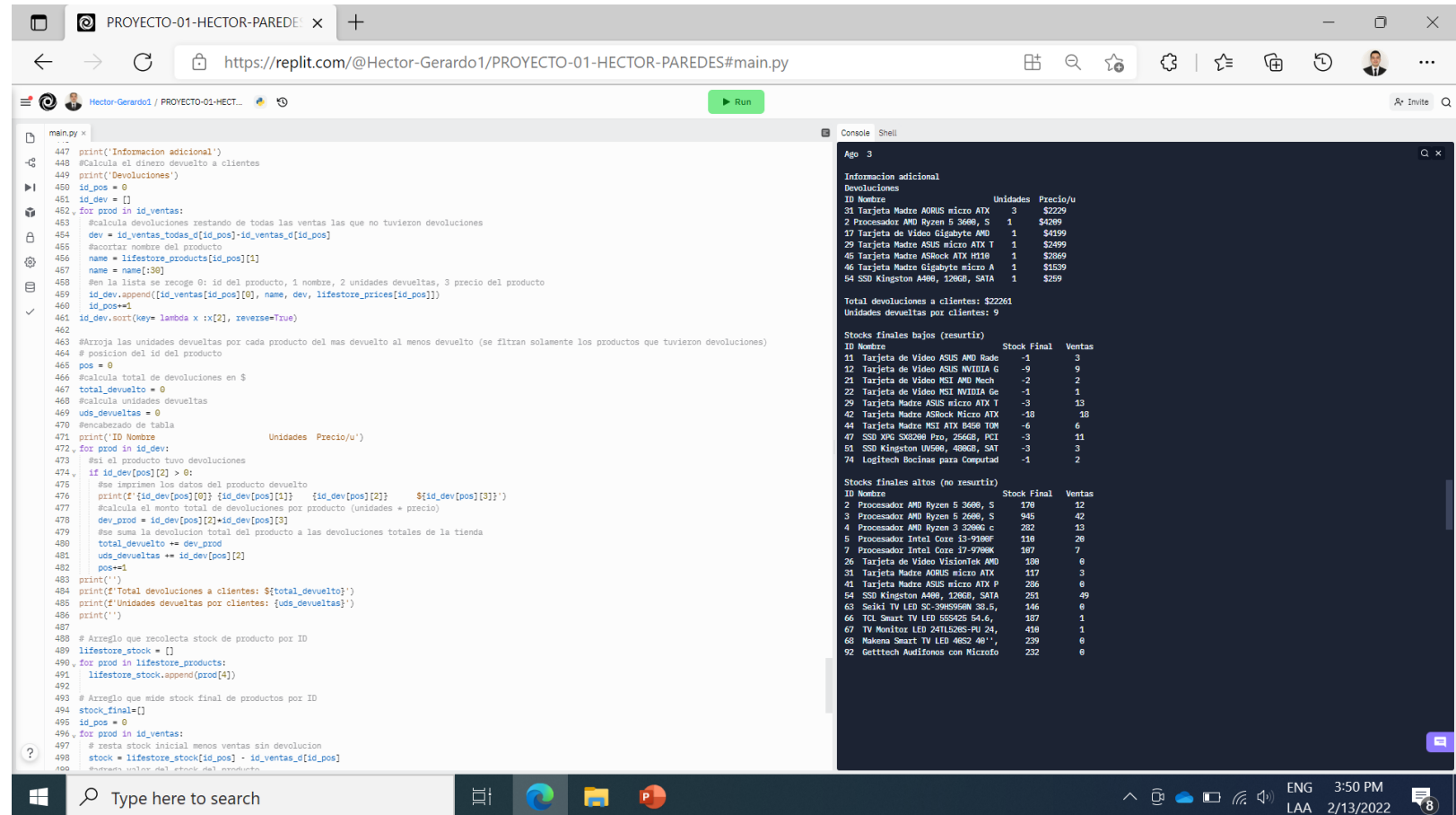
# ANÁLISIS ADICIONALES

---

# Información adicional

Para obtener resultados de interés para la empresa, se calcula lo siguiente:

- Devoluciones totales:
  - Para cada producto, se restan de las ventas (incluyendo devoluciones) las ventas sin devolución y se multiplican por el precio del producto.
- Stock final:
  - Del stock disponible en `lifestore_products` para cada producto, se restan las ventas totales (sin devolución).



```
main.py x
447 print('Información adicional')
448 #Calcula el dinero devuelto a clientes
449 print('Devoluciones')
450 id_pos = 0
451 id_dev = []
452 for prod in id_ventas:
453     #Calcula devoluciones restando de todas las ventas las que no tuvieron devoluciones
454     dev = id_ventas[id_pos] - id_ventas[id_dev]
455     #Cortar nombre del producto
456     name = lifestore_products[id_pos][1]
457     name = name[:30]
458     #En la lista se recoge 0: id del producto, 1 nombre, 2 unidades devueltas, 3 precio del producto
459     id_dev.append([id_ventas[id_pos][0], name, dev, lifestore_prices[id_pos]])
460     id_pos += 1
461     id_dev.sort(key=lambda x: x[2], reverse=True)
462
463 #Arroja las unidades devueltas por cada producto del mas devuelto al menos devuelto (se filtran solamente los productos que tuvieron devoluciones)
464 # posición del id del producto
465 pos = 0
466 #calcula total de devoluciones en $
467 total_devuelto = 0
468 #calcula unidades devueltas
469 uds_devueltas = 0
470 #encabezado de tabla
471 print('ID Nombre          Unidades Precio/u')
472 for prod in id_dev:
473     #si el producto tuvo devoluciones
474     if id_dev[pos][2] > 0:
475         #se imprimen los datos del producto devuelto
476         print(f'{id_dev[pos][0]} {id_dev[pos][1]} {id_dev[pos][2]} ${id_dev[pos][3]}')
477         #calcula el monto total de devoluciones por producto (unidades * precio)
478         dev_prod = id_dev[pos][2] * id_dev[pos][3]
479         #se suma la devolucion total del producto a las devoluciones totales de la tienda
480         total_devuelto += dev_prod
481         uds_devueltas += id_dev[pos][2]
482         pos += 1
483
484 print('')
485 print(f'Total devoluciones a clientes: ${total_devuelto}')
486 print(f'Unidades devueltas por clientes: {uds_devueltas}')
487 print('')
488
489 # Arreglo que recolecta stock de producto por ID
490 lifestore_stock = []
491 for prod in lifestore_products:
492     lifestore_stock.append(prod[4])
493
494 # Arreglo que mide stock final de productos por ID
495 stock_final = []
496 id_pos = 0
497 for prod in id_ventas:
498     # resta stock inicial menos ventas sin devolucion
499     stock = lifestore_stock[id_pos] - id_ventas[id_pos]
500     # devuelve un valor del stock del producto
```

Console Shell

Ago 3

Información adicional

Devoluciones

ID	Nombre	Unidades	Precio/u
31	Tarjeta Madre AORUS micro ATX	3	\$2229
2	Procesador AMD Ryzen 5 3600, S	1	\$4299
17	Tarjeta de Video Gigabyte AMD	1	\$4199
29	Tarjeta Madre ASUS micro ATX T	1	\$2499
45	Tarjeta Madre ASRock ATX H118	1	\$2869
46	Tarjeta Madre Gigabyte micro A	1	\$1539
54	SSD Kingston A400, 20GB, SATA	1	\$259

Total devoluciones a clientes: \$22261  
Unidades devueltas por clientes: 9

Stocks finales bajos (resurtir)

ID	Nombre	Stock Final	Ventas
11	Tarjeta de Video ASUS AMD Rade	-1	3
12	Tarjeta de Video MSI NVIDIA G	-9	9
21	Tarjeta de Video MSI AMD Mech	-2	2
22	Tarjeta de Video MSI NVIDIA Ge	-1	1
29	Tarjeta Madre ASUS micro ATX T	-3	13
42	Tarjeta Madre ASRock Micro ATX	-18	18
44	Tarjeta Madre MSI ATX B450 TOM	-6	6
47	SSD XPG SX8200 Pro, 256GB, PCI	-3	11
81	SSD Kingston UV500, 48GB, SAT	-3	3
74	Logitech Bocinas para Computad	-1	2

Stocks finales altos (no resurtir)

ID	Nombre	Stock Final	Ventas
2	Procesador AMD Ryzen 5 3600, S	178	12
3	Procesador AMD Ryzen 5 2600, S	945	42
4	Procesador AMD Ryzen 3 3200G c	282	13
5	Procesador Intel Core i3-9100F	116	26
7	Procesador Intel Core i7-9700K	187	7
26	Tarjeta de Video VisionTek AMD	188	0
31	Tarjeta Madre AORUS micro ATX	117	3
41	Tarjeta Madre ASUS micro ATX P	206	0
54	SSD Kingston A400, 128GB, SATA	251	49
63	Seiki TV LED 32-39HS9960 32.5,	146	0
66	TCL Smart TV LED 55S425 54.6,	187	1
67	TV Monitor LED 24TL528S-RU 24,	418	1
68	Makina Smart TV LED 48S2 48",	239	0
92	Gettitech Audifonos con Microfo	232	0

# RESULTADOS

---

# Top 5 Productos con más ventas

---

## Productos con mas ventas (unidades)

ID	Nombre	Cantidad
1 54	SSD Kingston A400, 120GB, SATA	(49 uds)
2 3	Procesador AMD Ryzen 5 2600, S	(42 uds)
3 5	Procesador Intel Core i3-9100F	(20 uds)
4 42	Tarjeta Madre ASRock Micro ATX	(18 uds)
5 57	SSD Adata Ultimate SU800, 256G	(15 uds)

## Productos con mas ventas (\$)

ID	Nombre	Cantidad
1 3	Procesador AMD Ryzen 5 2600, S	\$129738
2 7	Procesador Intel Core i7-9700K	\$59913
3 12	Tarjeta de Video ASUS NVIDIA G	\$59571
4 2	Procesador AMD Ryzen 5 3600, S	\$50508
5 5	Procesador Intel Core i3-9100F	\$35580

# Top 10 Productos más buscados

---

## Productos con mas busquedas

ID	Nombre	Busquedas
1	54 SSD Kingston A400, 120GB, SATA	263
2	57 SSD Adata Ultimate SU800, 256G	107
3	29 Tarjeta Madre ASUS micro ATX T	60
4	3 Procesador AMD Ryzen 5 2600, S	55
5	4 Procesador AMD Ryzen 3 3200G c	41
6	85 Logitech Audifonos Gamer G635	35
7	67 TV Monitor LED 24TL520S-PU 24,	32
8	7 Procesador Intel Core i7-9700K	31
9	5 Procesador Intel Core i3-9100F	30
10	47 SSD XPG SX8200 Pro, 256GB, PCI	30

# Top 5 Mejores y Peores Reseñas

---

## Productos con mejor reseñas promedio

ID	Nombre	Reseña
1	85 Logitech Audifonos Gamer G635	5.0
2	67 TV Monitor LED 24TL520S-PU 24,	5.0
3	7 Procesador Intel Core i7-9700K	5.0
4	8 Procesador Intel Core i5-9600K	5.0
5	21 Tarjeta de Video MSI AMD Mech	5.0

## Productos con peor reseña promedio

ID	Nombre	Reseña
1	26 Tarjeta de Video Gigabyte AMD	1.0
2	15 Tarjeta Madre ASRock ATX H110	1.0
3	63 Tarjeta Madre AORUS micro ATX	1.83
4	73 Tarjeta Madre Gigabyte micro A	2.0
5	39 Cougar Audifonos Gamer Phontum	3.0

# Análisis Categoría: Procesadores

---

```
1 Categoría: procesadores (9 productos)
Productos disponibles: [1, 2, 3, 4, 5, 6, 7, 8, 9]

Ventas totales categoría procesadores: $367517

Productos menos vendidos en la categoría procesadores
ID Nombre                               Cantidad
1 9 Procesador Intel Core i3-8100,    (8 uds)
2 1 Procesador AMD Ryzen 3 3300X S    (2 uds)
3 6 Procesador Intel Core i9-9900K    (3 uds)
4 8 Procesador Intel Core i5-9600K    (4 uds)
5 7 Procesador Intel Core i7-9700K    (7 uds)

Productos menos buscados en la categoría procesadores
ID Nombre                               Busquedas
1 9 Procesador Intel Core i3-8100,      0
2 1 Procesador AMD Ryzen 3 3300X S      2
3 6 Procesador Intel Core i9-9900K      3
4 8 Procesador Intel Core i5-9600K      4
5 2 Procesador AMD Ryzen 5 3600, S     12
6 5 Procesador Intel Core i3-9100F     20
7 7 Procesador Intel Core i7-9700K      7
8 4 Procesador AMD Ryzen 3 3200G c     13
9 3 Procesador AMD Ryzen 5 2600, S     42
```



# Análisis Categoría: Tarjetas de video

---

2 Categoría: tarjetas de video (19 productos)

Productos disponibles: [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]

Ventas totales categoria tarjetas de video: \$132025

Productos menos vendidos en la categoria tarjetas de video

ID	Nombre	Cantidad
1 14	Tarjeta de Video EVGA NVIDIA G	(0 uds)
2 15	Tarjeta de Video EVGA NVIDIA G	(0 uds)
3 16	Tarjeta de Video EVGA NVIDIA G	(0 uds)
4 17	Tarjeta de Video Gigabyte AMD	(0 uds)
5 19	Tarjeta de Video Gigabyte NVID	(0 uds)

Productos menos buscados en la categoria tarjetas de video

ID	Nombre	Busquedas
1 14	Tarjeta de Video EVGA NVIDIA G	0
2 16	Tarjeta de Video EVGA NVIDIA G	0
3 19	Tarjeta de Video Gigabyte NVID	0
4 20	Tarjeta de Video Gigabyte NVID	0
5 23	Tarjeta de Video MSI Radeon X1	0
6 24	Tarjeta de Video PNY NVIDIA Ge	0
7 27	Tarjeta de Video VisionTek AMD	0
8 10	MSI GeForce 210, 1GB GDDR3, DV	1
9 13	Tarjeta de Video Asus NVIDIA G	1
10 17	Tarjeta de Video Gigabyte AMD	0

# Análisis Categoría: Tarjetas madre

---

3 Categoría: tarjetas madre (18 productos)

Productos disponibles: [29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46]

Ventas totales categoría tarjetas madre: \$113727

Productos menos vendidos en la categoría tarjetas madre

ID	Nombre	Cantidad
1 30	Tarjeta Madre AORUS ATX Z390 E	(0 uds)
2 32	Tarjeta Madre ASRock Z390 Phan	(0 uds)
3 34	Tarjeta Madre ASUS ATX ROG STR	(0 uds)
4 35	Tarjeta Madre Gigabyte micro A	(0 uds)
5 36	Tarjeta Madre Gigabyte micro A	(0 uds)

Productos menos buscados en la categoría tarjetas madre

ID	Nombre	Busquedas
1 30	Tarjeta Madre AORUS ATX Z390 E	0
2 32	Tarjeta Madre ASRock Z390 Phan	0
3 34	Tarjeta Madre ASUS ATX ROG STR	0
4 36	Tarjeta Madre Gigabyte micro A	0
5 37	Tarjeta Madre ASRock ATX Z490	0
6 38	Tarjeta Madre Gigabyte Micro A	0
7 41	Tarjeta Madre ASUS micro ATX P	0
8 43	Tarjeta Madre ASUS ATX ROG STR	0
9 33	Tarjeta Madre ASUS ATX PRIME Z	2
10 35	Tarjeta Madre Gigabyte micro A	0

# Análisis Categoría: Discos Duros

---

```
4 Categoría: discos duros (13 productos)
Productos disponibles: [47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59]

Ventas totales categoria discos duros: $93237

Productos menos vendidos en la categoria discos duros
ID Nombre                               Cantidad
1 53 SSD Addlink Technology S70, 51    (0 uds)
2 55 SSD para Servidor Supermicro S    (0 uds)
3 56 SSD para Servidor Lenovo Think    (0 uds)
4 58 SSD para Servidor Lenovo Think    (0 uds)
5 59 SSD Samsung 860 EVO, 1TB, SATA    (0 uds)

Productos menos buscados en la categoria discos duros
ID Nombre                               Busquedas
1 53 SSD Addlink Technology S70, 51     0
2 55 SSD para Servidor Supermicro S     0
3 58 SSD para Servidor Lenovo Think     0
4 59 SSD Samsung 860 EVO, 1TB, SATA     0
5 56 SSD para Servidor Lenovo Think     0
6 52 SSD Western Digital WD Blue 3D     2
7 50 SSD Crucial MX500, 1TB, SATA I     1
8 49 Kit SSD Kingston KC600, 1TB, S     3
9 51 SSD Kingston UV500, 480GB, SAT     3
10 48 SSD Kingston A2000 NVMe, 1TB,     9
```

# Análisis Categoría: Memorias USB

---

5 Categoría: memorias usb (2 productos)

Productos disponibles: [60, 61]

Ventas totales categoria memorias usb: \$2519

Productos menos vendidos en la categoria memorias usb

ID	Nombre	Cantidad
1	61 Kit Memoria RAM Corsair Vengea	(0 uds)
2	60 Kit Memoria RAM Corsair Domina	(1 uds)

Productos menos buscados en la categoria memorias usb

ID	Nombre	Busquedas
1	61 Kit Memoria RAM Corsair Vengea	0
2	60 Kit Memoria RAM Corsair Domina	1

# Análisis Categoría: Pantallas

---

6 Categoría: pantallas (12 productos)  
Productos disponibles: [62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73]

Ventas totales categoría pantallas: \$11278

Productos menos vendidos en la categoría pantallas

ID	Nombre	Cantidad
1 62	Makena Smart TV LED 32S2 32'',	(0 uds)
2 63	Seiki TV LED SC-39HS950N 38.5,	(0 uds)
3 64	Samsung TV LED LH43QMREBGCXG0	(0 uds)
4 65	Samsung Smart TV LED UN70RU710	(0 uds)
5 68	Makena Smart TV LED 40S2 40'',	(0 uds)

Productos menos buscados en la categoría pantallas

ID	Nombre	Busquedas
1 62	Makena Smart TV LED 32S2 32'',	0
2 64	Samsung TV LED LH43QMREBGCXG0	0
3 65	Samsung Smart TV LED UN70RU710	0
4 68	Makena Smart TV LED 40S2 40'',	0
5 69	Hisense Smart TV LED 40H5500F	0
6 71	Samsung Smart TV LED UN32J4290	0
7 72	Hisense Smart TV LED 50H8F 49.	0
8 70	Samsung Smart TV LED 43, Full	0
9 63	Seiki TV LED SC-39HS950N 38.5,	0
10 73	Samsung Smart TV LED UN55TU700	0

# Análisis Categoría: Bocinas

---

7 Categoría: bocinas (10 productos)  
Productos disponibles: [74, 75, 76, 77, 78, 79, 80, 81, 82, 83]

Ventas totales categoria bocinas: \$8478

Productos menos vendidos en la categoria bocinas

ID	Nombre	Cantidad
1	75 Lenovo Barra de Sonido, Alambr	(0 uds)
2	76 Acteck Bocina con Subwoofer AX	(0 uds)
3	77 Verbatim Bocina Portatil Mini,	(0 uds)
4	78 Ghia Bocina Portatil BX300, B1	(0 uds)
5	79 Naceb Bocina Portatil NA-0301,	(0 uds)

Productos menos buscados en la categoria bocinas

ID	Nombre	Busquedas
1	75 Lenovo Barra de Sonido, Alambr	0
2	77 Verbatim Bocina Portatil Mini,	0
3	78 Ghia Bocina Portatil BX300, B1	0
4	79 Naceb Bocina Portatil NA-0301,	0
5	81 Ghia Bocina Portatil BX900, B1	0
6	82 Ghia Bocina Portatil BX400, B1	0
7	83 Ghia Bocina Portatil BX500, B1	0
8	80 Ghia Bocina Portatil BX800, B1	0
9	76 Acteck Bocina con Subwoofer AX	0
10	74 Logitech Bocinas para Computad	2

# Análisis Categoría: Bocinas

---

```
8 Categoría: audifonos (13 productos)
Productos disponibles: [84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96]

Ventas totales categoria audifonos: $9135

Productos menos vendidos en la categoria audifonos
ID Nombre                               Cantidad
1 86 ASUS Audifonos Gamer ROG Theta    (0 uds)
2 87 Acer Audifonos Gamer Galea 300    (0 uds)
3 88 Audifonos Gamer Balam Rush Orp    (0 uds)
4 90 Energy Sistem Audifonos con Mi    (0 uds)
5 91 Genius GHP-400S Audifonos, Ala    (0 uds)

Productos menos buscados en la categoria audifonos
ID Nombre                               Busquedas
1 86 ASUS Audifonos Gamer ROG Theta      0
2 87 Acer Audifonos Gamer Galea 300      0
3 88 Audifonos Gamer Balam Rush Orp      0
4 90 Energy Sistem Audifonos con Mi      0
5 92 Getttech Audifonos con Microfo      0
6 96 Klip Xtreme Audifonos Blast, B      0
7 93 Ginga Audifonos con Microfono      0
8 91 Genius GHP-400S Audifonos, Ala      0
9 95 Iogear Audifonos Gamer GHG601,      0
10 94 HyperX Audifonos Gamer Cloud F      1
```

## Ventas por categoría

Las categorías con más ventas son:

- Procesadores
- Tarjetas de video
- Tarjetas madres

### Ventas totales por categoría

Categoría	Ventas(\$)
1 procesadores	\$367517
2 tarjetas de video	\$132025
3 tarjetas madre	\$113727
4 discos duros	\$93237
5 pantallas	\$11278
6 audifonos	\$9135
7 bocinas	\$8478
8 memorias usb	\$2519



# Ventas mensuales y anuales

---

## Ventas Mensuales y Anuales

### Resultados mensuales

Mes	Ingresos(\$)	Ventas(uds)	Prom ventas(\$)
Ene	\$117738	52	\$2264.19
Feb	\$107270	40	\$2681.75
Mar	\$162931	49	\$3325.12
Abr	\$191066	74	\$2581.97
May	\$91936	34	\$2704.0
Jun	\$36949	11	\$3359.0
Jul	\$26949	11	\$2449.91
Ago	\$3077	3	\$1025.67

### Resultados anuales

Ingresos anuales: \$737916

Ventas realizadas anuales: 274

# Meses con más ventas

Los meses con más ventas tanto en ingresos como unidades son:

- Abril
- Marzo
- Enero
- Febrero

## Meses con mas ingresos

Mes	Ingresos(\$)
-----	--------------

Abr	\$191066
-----	----------

Mar	\$162931
-----	----------

Ene	\$117738
-----	----------

Feb	\$107270
-----	----------

May	\$91936
-----	---------

Jun	\$36949
-----	---------

Jul	\$26949
-----	---------

Ago	\$3077
-----	--------

## Meses con mas ventas

Mes	Ventas
-----	--------

Abr	74
-----	----

Mar	49
-----	----

Ene	52
-----	----

Feb	40
-----	----

May	34
-----	----

Jun	11
-----	----

Jul	11
-----	----

Ago	3
-----	---

# Análisis Devoluciones

---

## Devoluciones

ID	Nombre	Unidades	Precio/u
31	Tarjeta Madre AORUS micro ATX	3	\$2229
2	Procesador AMD Ryzen 5 3600, S	1	\$4209
17	Tarjeta de Video Gigabyte AMD	1	\$4199
29	Tarjeta Madre ASUS micro ATX T	1	\$2499
45	Tarjeta Madre ASRock ATX H110	1	\$2869
46	Tarjeta Madre Gigabyte micro A	1	\$1539
54	SSD Kingston A400, 120GB, SATA	1	\$259

Total devoluciones a clientes: \$22261

Unidades devueltas por clientes: 9

# Análisis de Stock

## Stocks finales altos (no resurtir)

ID	Nombre	Stock Final	Ventas
2	Procesador AMD Ryzen 5 3600, S	170	12
3	Procesador AMD Ryzen 5 2600, S	945	42
4	Procesador AMD Ryzen 3 3200G c	282	13
5	Procesador Intel Core i3-9100F	110	20
7	Procesador Intel Core i7-9700K	107	7
26	Tarjeta de Video VisionTek AMD	180	0
31	Tarjeta Madre AORUS micro ATX	117	3
41	Tarjeta Madre ASUS micro ATX P	286	0
54	SSD Kingston A400, 120GB, SATA	251	49
63	Seiki TV LED SC-39HS950N 38.5,	146	0
66	TCL Smart TV LED 55S425 54.6,	187	1
67	TV Monitor LED 24TL520S-PU 24,	410	1
68	Makena Smart TV LED 40S2 40'',	239	0
92	Getttech Audifonos con Microfo	232	0
93	Ginga Audifonos con Microfono	139	0

## Stocks finales bajos (resurtir)

ID	Nombre	Stock Final	Ventas
11	Tarjeta de Video ASUS AMD Rade	-1	3
12	Tarjeta de Video ASUS NVIDIA G	-9	9
21	Tarjeta de Video MSI AMD Mech	-2	2
22	Tarjeta de Video MSI NVIDIA Ge	-1	1
29	Tarjeta Madre ASUS micro ATX T	-3	13
42	Tarjeta Madre ASRock Micro ATX	-18	18
44	Tarjeta Madre MSI ATX B450 TOM	-6	6
47	SSD XPG SX8200 Pro, 256GB, PCI	-3	11
51	SSD Kingston UV500, 480GB, SAT	-3	3
74	Logitech Bocinas para Computad	-1	2

# CONCLUSIONES

---

# Conclusiones

---

- El periodo con más ventas es el primer trimestre del año, mientras el tercer trimestre (septiembre-diciembre) no registra ventas.
- La categoría de USB aporta muy pocas ventas a la empresa.
- La demanda productos 11, 12, 21, 22, 29, 42, 44, 47, 51, y 74 sobrepasó el stock inicial disponible durante el año. Es importante resurtir estos productos y contar con un mayor stock.
- Incentivar las ventas de productos con un stock final mayor a 100, no resurtirlos. (Ejemplos: productos 26, 31, 41, 63 y 66).
- Revisar el estado de las “Tarjeta Madre AORUS” (ID 31) al ser un producto que reincidió en devoluciones (3 devoluciones).