EMTECH
Emerging Technologies Institute

Análisis
de la rotación de productos

PROYECTO 01

T: 5528423451 **E:** elias.bautistaflores@outlook.com

Resumen

El presente análisis de la rotación de productos identificando los siguientes elementos:

1. Productos más vendidos y productos rezagados a partir del análisis de las categorías con menores ventas y categorías con menores búsquedas.
2. Productos por reseña en el servicio a partir del análisis de categorías con mayores ventas y categorías con mayores búsquedas.

Dentro del documento se sugiere una estrategia de productos a retirar del mercado, así como un par de observaciones respecto a la base de datos.

Contenido

Venta de Productos

Puntos importantes relacionados a la venta de los productos.

- De los 96 productos disponibles, solo se vendieron 41 (sin contar devoluciones), lo que significa que 55 productos del catalogo no registraron ninguna venta.
- De las 283 ventas reportadas, 9 fueron devoluciones, significa que las ventas reales fueron de 274.
- El valor de las devoluciones fue por $22,261.
- Existen articulos que reportaron ventas y ya no tienen stock.

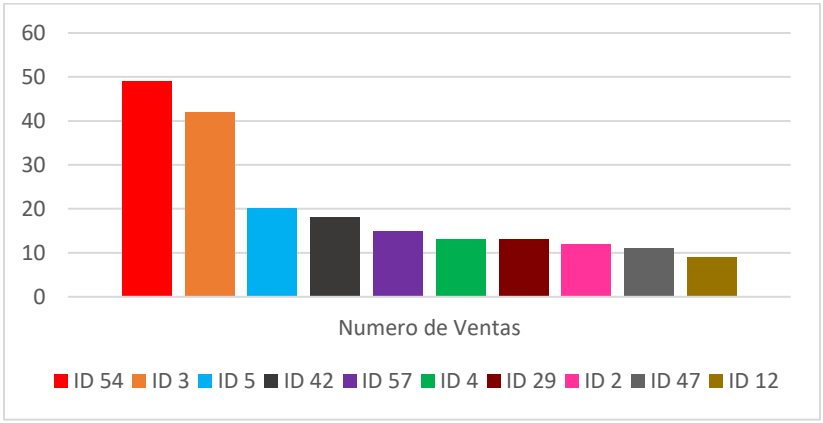
Productos con Mayor Venta

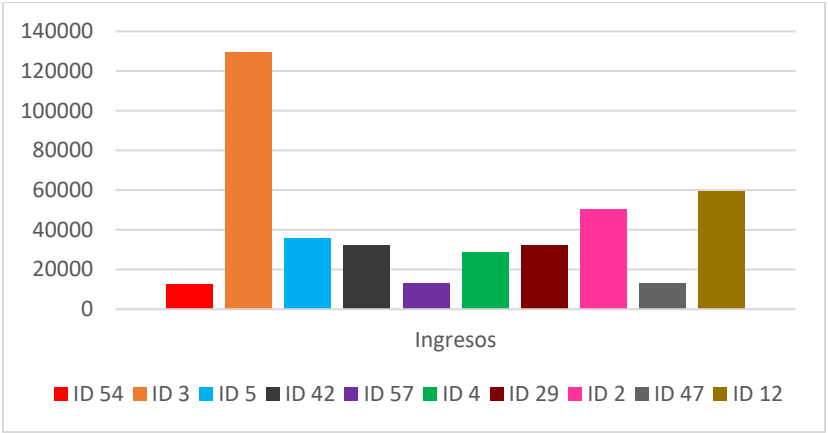
Productos que más se vendieron

1. SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm , con 49 ventas. (ID: 54)
2. Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth con 42 ventas. (ID: 3)
3. Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake), con 20 ventas. (ID: 5)
4. Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD con 18 ventas. (ID: 42)
5. SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm, con 15 ventas. (ID: 57)
6. Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire , con 13 ventas. (ID: 4)
7. Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD, con 13 ventas. (ID: 29)

8. Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth, con 12 ventas. (ID: 2)
9. SSD XPG SX8200 Pro, 256GB, PCI Express, M.2, con 11 ventas. (ID: 47)
10. Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0, con 9 ventas. (ID: 12

Grafica de los productos que más se vendieron y los ingresos que generaron.

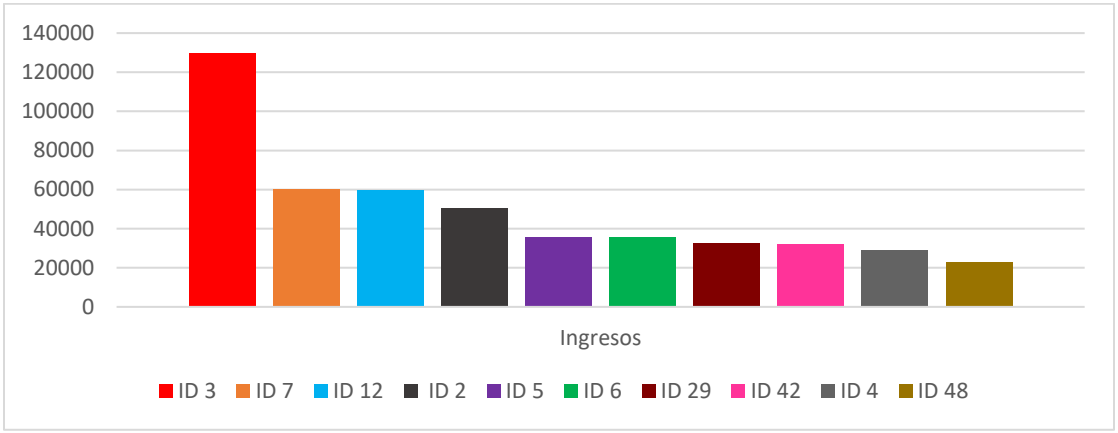


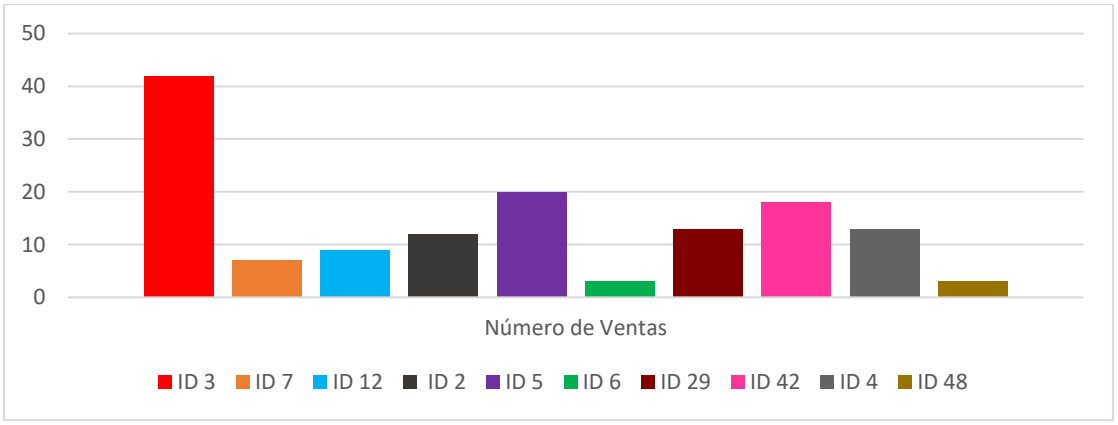


Productos con mayor ingreso

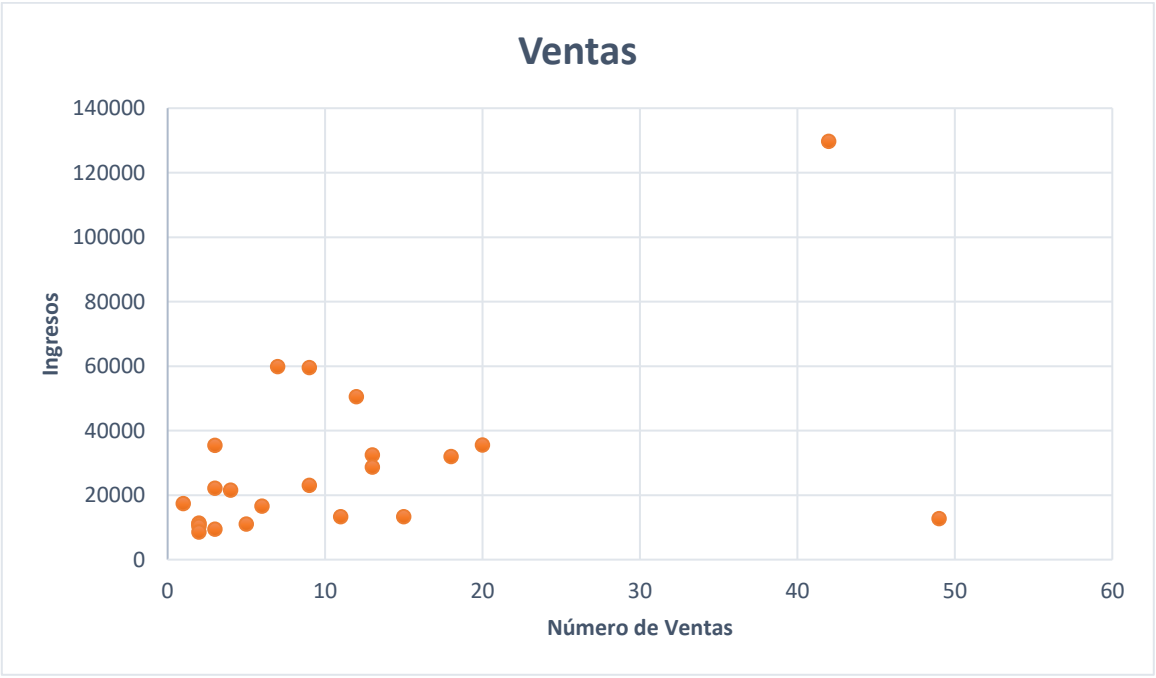
1. ID: 3, Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth, con 42 ventas, $129,738 en ingresos.
2. ID: 7, Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake) con 7 ventas, $59,913 en ingresos.
3. ID: 12, Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0 con 9 ventas, $59,571 en ingresos.
4. ID: 2, Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth, con 12 ventas, $50,508 en ingresos.
5. ID: 5, Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake), con 20 ventas, $35,580 en ingresos.
6. ID: 6, Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake), con 3 ventas, $35,427 en ingresos.
7. ID: 29, Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD con 13 ventas, $32,487 en ingresos.
8. ID: 42, Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD, con 18 ventas, $32,022 en ingresos.
9. ID: 4, Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire, con 13 ventas, $28,717 en ingresos.
10. ID: 48, SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2 con 9 ventas, $23,031 en ingresos.
11. ID: 11, Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Express 3.0 con 3 ventas, $22,197 en ingresos.
12. ID: 8, Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generiación - Coffee Lake), con 4 ventas, $21,596 en ingresos.
13. ID: 40, Tarjeta Madre Gigabyte XL-ATX TRX40 Designare, S-sTRX4, AMD TRX40, 256GB DDR4 para AMD, con 1 venta, $17,439 en ingresos.
14. ID: 44, Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AMD B450, 64GB DDR4 para AMD, con 6 ventas, $16,554 en ingresos.
15. ID: 57, SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm, con 15 ventas, $13,335 en ingresos.
16. ID: 47, SSD XPG SX8200 Pro, 256GB, PCI Express, M.2, con 11 ventas, $13,299.
17. ID: 54, SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm, con 49 ventas, $12,691.
18. ID: 52, SSD Western Digital WD Blue 3D NAND, 2TB, M.2, con 2 ventas, $11,318 en ingresos
19. ID: 25, Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming, 8GB 128-bit GDDR6, PCI Express 4.0, con 2 ventas, $11,058 en ingresos.
20. ID: 18, Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030, 2GB 64-bit GDDR5, PCI Express x16 3.0, con 5 ventas, $10,995 en ingresos.

Grafica de los productos que más ingresos generaron y cuántas ventas tuvieron.





Gráfica de ventas e ingresos de los productos.



Productos con Menor Venta

Productos que menos se vendieron

Dentro de esta lista existen 55 productos con 0 ventas registradas, dentro de esta lista los productos más relevantes son aquellos que tienen una gran cantidad de stock, tienen exceso de inventario y no se están vendiendo.

Productos Con Exceso de Inventarios

	ID	STOCK	CATEGORIA	VALOR DEL STOCK
TARJETA MADRE ASUS MICRO ATX PRIME H370M-PLUS/CSM, S-1151, INTEL H370, HDMI, 64GB DDR4 PARA INTEL	41	286	TARJETAS MADRE	$952,094
MAKENA SMART TV LED 40S2 40'', FULL HD, WIDESCREEN, NEGRO	68	239	PANTALLAS	$1,010,731
GETTTECH AUDÍFONOS CON MICRÓFONO SONORITY, ALÁMBRICO, 1.2 METROS, 3.5MM, NEGRO/ROSA	92	232	AUDIFONOS	$34,568
TCL SMART TV LED 55S425 54.6, 4K ULTRA HD, WIDESCREEN, NEGRO	66	188	PANTALLAS	$ 1,513,212
TARJETA DE VIDEO VISIONTEK AMD RADEON HD 5450, 1GB DDR3, PCI EXPRESS X16 2.1	26	180	TARJETAS DE VIDEO	$ 224,820
SEIKI TV LED SC-39HS950N 38.5, HD, WIDESCREEN, NEGRO	63	146	PANTALLAS	$ 491,874
GINGA AUDÍFONOS CON MICRÓFONO GI18ADJ01BT-RO, BLUETOOTH, ALÁMBRICO/INALÁMBRICO, 3.5MM, ROJO	93	102	AUDIFONOS	$ 22,240
ASUS T. MADRE UATX M4A88T-M, S-AM3, DDR3 PARA PHENOM II/ATHLON II/SEMPRON 100	39	98	TARJETAS MADRE	$ 212,562
HISENSE SMART TV LED 40H5500F 39.5, FULL HD, WIDESCREEN, NEGRO	69	94	PANTALLAS	$ 503,746
SAMSUNG TV LED LH43QMREBGCXGO 43, 4K ULTRA HD, WIDESCREEN, NEGRO	64	71	PANTALLAS	$ 854,059
TARJETA MADRE ASROCK ATX Z490 STEEL LEGEND, S-1200, INTEL Z490, HDMI, 128GB DDR4 PARA INTEL	37	60	TARJETAS MADRE	$ 257,340

TARJETA MADRE AORUS ATX Z390 ELITE, S-1151, INTEL Z390, HDMI, 64GB DDR4 PARA INTEL	30	50	TARJETAS MADRE	$201,450

Ventas por Categoría

CATEGORIA	Número de Ventas
Procesadores	104
Tarjetas de video	26
Tarjetas madre	49
Discos duros	94
Memorias usb	1
Pantallas	2
Bocinas	2
Audífonos	5

Como podemos observar hay cuatro categorías que tienen muy bajas ventas, de la cual destaca la categoría pantallas por su alto costo y su exceso en el inventario. Muchos de los productos de la categoría *pantallas* representan un gran costo en el inventario y no reportan ninguna venta.

Productos Devueltos

- Se registraron 9 devoluciones en total, por un valor de $22,261.
- El producto con ID 31 tiene registradas 3 devoluciones, es el único producto que tiene registrada más de una devolución.

ID	Producto	Precio	Stock
31	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	2229	120

Comprobar si el lote de este producto funciona correctamente y que no existan productos dañados, de ser asi consultar con el proveedor.

Calificaciones

Productos con Mejor Calificación

Los diez productos que más veces han sido evaluados con 5 (el score más alto) son:

	ID	VECES QUE HA SIDO EVALUADO CON 5
SSD KINGSTON A400, 120GB, SATA III, 2.5'', 7MM	54	38
PROCESADOR AMD RYZEN 5 2600, S-AM4, 3.40GHZ, SIX-CORE, 16MB L3 CACHE, CON DISIPADOR WRAITH STEALTH	3	35
PROCESADOR INTEL CORE I3-9100F, S-1151, 3.60GHZ, QUAD-CORE, 6MB CACHE (9NA. GENERACIÓN - COFFEE LAKE)	5	14
SSD ADATA ULTIMATE SU800, 256GB, SATA III, 2.5'', 7MM	57	13
TARJETA MADRE ASROCK MICRO ATX B450M STEEL LEGEND, S-AM4, AMD B450, HDMI, 64GB DDR4 PARA AMD	42	10
TARJETA MADRE ASUS MICRO ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 PARA AMD	29	8
PROCESADOR AMD RYZEN 3 3200G CON GRÁFICOS RADEON VEGA 8, S-AM4, 3.60GHZ, QUAD-CORE, 4MB L3, CON DISIPADOR WRAITH SPIRE	4	7
PROCESADOR INTEL CORE I7-9700K, S-1151, 3.60GHZ, 8-CORE, 12MB SMART CACHE (9NA. GENERACIÓN COFFEE LAKE)	7	7
TARJETA DE VIDEO ASUS NVIDIA GEFORCE GTX 1660 SUPER EVO OC, 6GB 192-BIT GDDR6, PCI EXPRESS X16 3.0	12	7
SSD XPG SX8200 PRO, 256GB, PCI EXPRESS, M.2	47	7

Productos con Peor Calificación

Solo existen cuatro productos que han sido evaluados con 1(la calificación más baja) son:

	ID	VECES QUE HA SIDO EVALUADO CON 1
SSD KINGSTON A400, 120GB, SATA III, 2.5", 7MM	31	4
PROCESADOR AMD RYZEN 5 2600, S-AM4, 3.40GHZ, SIX-CORE, 16MB L3 CACHE, CON DISIPADOR WRAITH STEALTH	29	2
PROCESADOR INTEL CORE I3-9100F, S-1151, 3.60GHZ, QUAD-CORE, 6MB CACHE (9NA. GENERACIÓN - COFFEE LAKE)	17	1
SSD ADATA ULTIMATE SU800, 256GB, SATA III, 2.5", 7MM	45	1

Nota: El producto con ID 31 es el que más veces ha sido evaluado con 1 y también el producto que más ha sido devuelto.

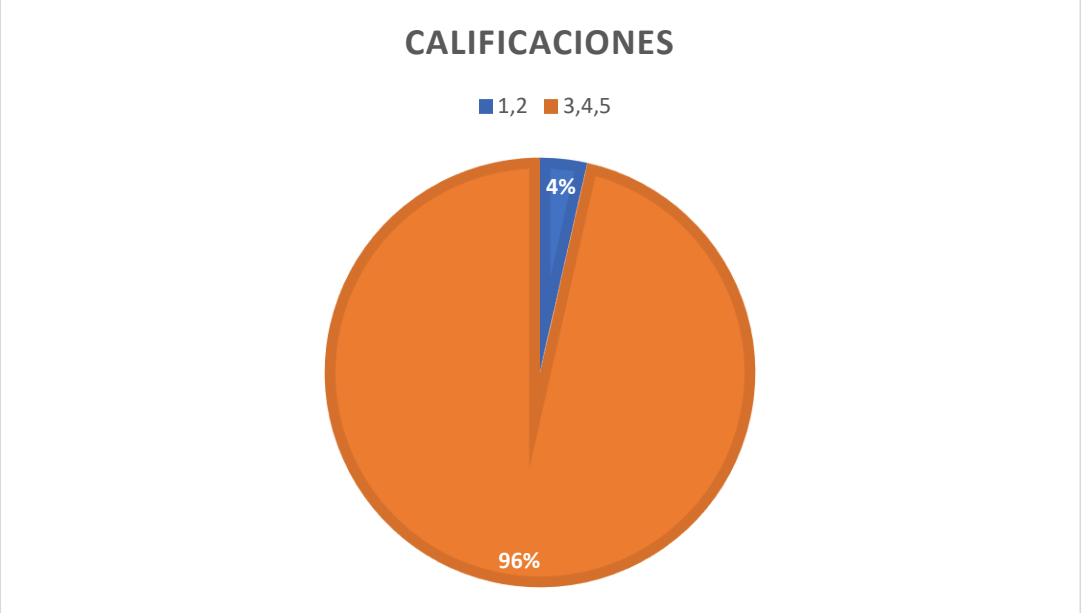
Si consideramos a los productos evaluados con 2 también, la lista crece a seis productos:

	ID	VECES QUE HA SIDO EVALUADO CON 1 ó 2
SSD KINGSTON A400, 120GB, SATA III, 2.5", 7MM	31	4
PROCESADOR AMD RYZEN 5 2600, S-AM4, 3.40GHZ, SIX-CORE, 16MB L3 CACHE, CON DISIPADOR WRAITH STEALTH	29	2
PROCESADOR INTEL CORE I3-9100F, S-1151, 3.60GHZ, QUAD-CORE, 6MB CACHE (9NA. GENERACIÓN - COFFEE LAKE)	17	1
SSD ADATA ULTIMATE SU800, 256GB, SATA III, 2.5", 7MM	45	1
TARJETA MADRE GIGABYTE MICRO ATX GA-H110M-DS2, S-1151, INTEL H110, 32GB DDR4 PARA INTEL	46	1
SSD KINGSTON A400, 120GB, SATA III, 2.5", 7MM	54	1

De todas las ventas únicamente 10 tienen calificación de 1 o 2, y 8 ventas tienen calificación de 1.

Si en total se registraron 283 ventas, quiere decir que solo el 3% de las ventas tuvieron la calificación más baja.

Si consideramos 1 y 2 como calificaciones quiere decir que el 96% de las ventas tiene una calificación en el rango de 3 a 5 (regular a excelente).



Ingresos

Ingresos Mensuales

MES	INGRESO	Número de Ventas
Enero	$117,738	52
Febrero	$107,270	40
Marzo	$162,931	49
Abril	$191,066	74
Mayo	$91,936	34
Junio	$36,949	11
Julio	$26,949	11
Agosto	$3,077	3
Septiembre	$0	0
Octubre	$0	0
Noviembre	$0	0
Diciembre	$0	0

Los últimos 4 meses del año no se registraron ventas.

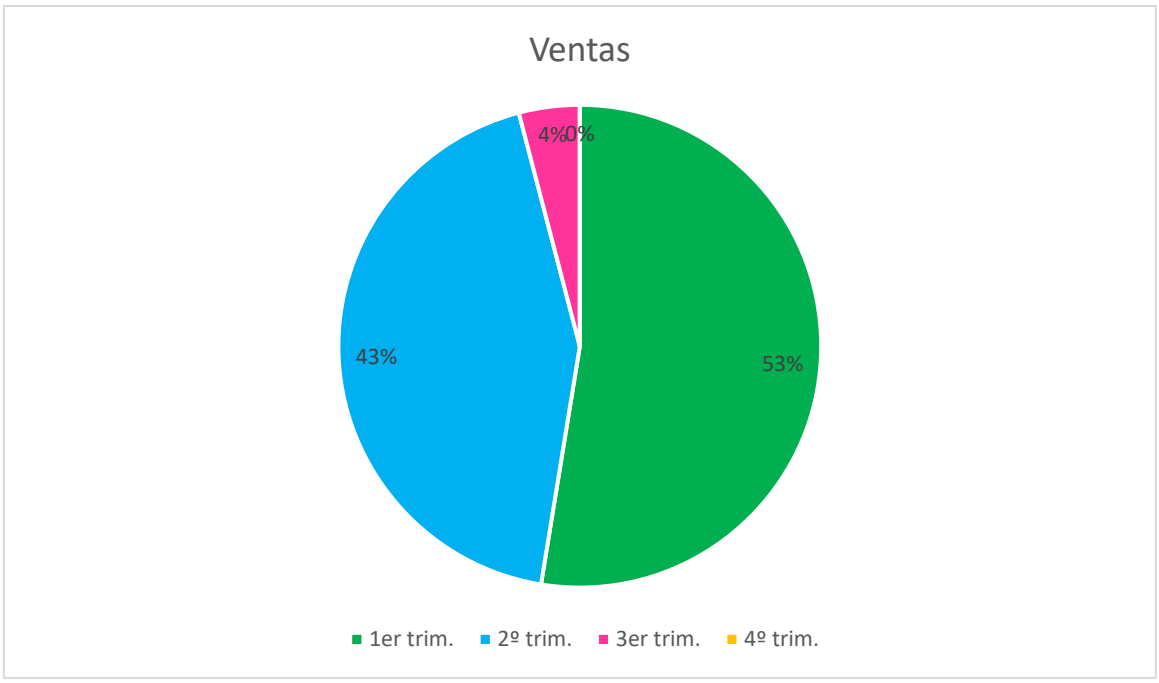
Promedio Mensual

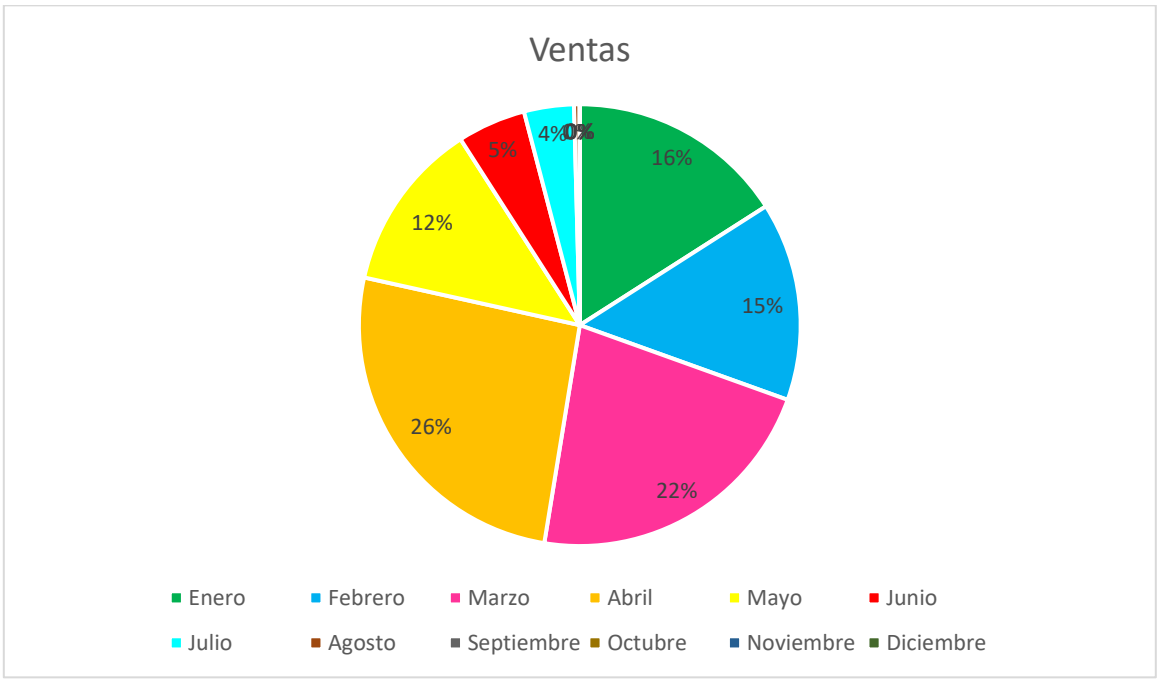
El promedio mensual es de: $61,493.00

El mes con mayor número de ventas y mayor ingreso fue Abril.

Total Anual

Se registraron 274 (restando las 9 ventas que fueron devoluciones) ventas en el 2020 por un valor total de $737,916.


Ventas
4%
0%
43%
53%
1er trim.
2º trim.
3er trim.
4º trim.


Ventas
5%
4%
16%
12%
15%
26%
22%
Enero
Febrero
Marzo
Abril
Mayo
Junio
Julio
Agosto
Septiembre
Octubre
Noviembre
Diciembre

Búsquedas

Productos con mayores búsquedas

Productos que más se buscaron

1. ID: 54, SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm con 263 búsquedas.
2. ID: 57, SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm con 107 búsquedas.
3. ID: 29, Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD con 60 búsquedas.
4. ID: 3, Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth con 55 búsquedas.
5. ID: 4, Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire con 41 búsquedas
6. ID 85, Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul con 35 búsquedas.
7. ID: 67, TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro con 32 busquedas.
8. ID: 7, Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake) con 31 búsquedas
9. ID: 5, Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake) con 30 búsquedas.
10. ID: 47, SSD XPG SX8200 Pro, 256GB, PCI Express, M.2 con 30 busquedas.
11. ID: 48, SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2 con 27 busquedas.
12. ID: 44, Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AMD B450, 64GB DDR4 para AMD con 25 búsquedas.
13. ID: 2, Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth con 24 bússquedas.
14. ID: 42, Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD con 23 búsquedas.
15. ID: 8, Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generiación - Coffee Lake) con 20 búsquedas

Productos con cero búsquedas y mucho producto en stock.

1. ID: 41, Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM, S-1151, Intel H370, HDMI, 64GB DDR4 para Intel con 286 en stock.
2. ID: 68, Makena Smart TV LED 40S2 40'', Full HD, Widescreen, Negro con 239 en stock.
3. ID: 92, Getttech Audífonos con Micrófono Sonority, Alámbrico, 1.2 Metros, 3.5mm, Negro/Rosa con 232 en stock.
4. ID: 69, Hisense Smart TV LED 40H5500F 39.5, Full HD, Widescreen, Negro con 94 en stock.
5. ID: 64, Samsung TV LED LH43QMREBGCXGO 43, 4K Ultra HD, Widescreen, Negrocon 71 en stock.
6. ID 37, Tarjeta Madre ASRock ATX Z490 STEEL LEGEND, S-1200, Intel Z490, HDMI, 128GB DDR4 para Intel con 60 en stock
7. ID: 30, Tarjeta Madre AORUS ATX Z390 ELITE, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel con 50 en stock.
8. ID: 33, Tarjeta Madre ASUS ATX PRIME Z390-A, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel con 43 en stock.
9. ID: 14, SSD XPG SX8200 Pro, 256GB, PCI Express, M.2 con 36 en stock.
10. ID: 82, SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2 con 31 en stock.

Productos sin stock

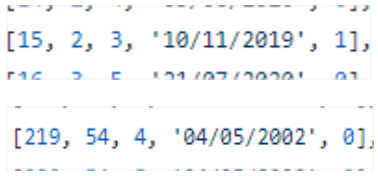
Debe considerarse renovar el stock de los siguientes artículos con stock en ceros debido a su número de ventas y a su número de búsquedas.

ID	Producto	Número de Ventas	Número de Búsquedas
42	Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	18	23
12	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0	9	15
21	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC, 8GB 128-bit GDDR6, PCI Express 4.0	2	15
51	SSD Kingston UV500, 480GB, SATA III, mSATA	3	11
44	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AMD B450, 64GB DDR4 para AMD	6	25
22	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC, 4GB 128-bit GDDR5, PCI Express x16 3.0	1	5

Fechas Atípicas

Como observación me gustaría señalar que dentro de la lista ventas existen 2 fechas atípicas que no son del año 2020, una es del 2019 y otra es del 2002. Verificar si están fechas son correctas.

ID	Id del Producto	Score	FECHA	Refund
15	2	3	10/11/2019	1
219	54	4	04/05/2002	0



Capturas tomadas del Github del Proyecto

Conclusiones

A modo de conclusión dejo las siguientes sugerencias:

1. Considerar aplicar ofertas para las pantallas, ya que es un producto de mucho valor en el inventario y no se esta vendiendo. Después de liberar el inventario debería considerarse retirar estos productos del inventario.
2. Definir una fecha límite para vender gran parte de las pantallas en caso de no lograrlo, considerar ofrecer una liquidación.
3. A partir de septiembre ya no se registraron ventas, puede ser porque algunos artículos de la categoría tarjetas de video y tarjetas madre ya no tienen stock.
4. Comprobar el lote del producto con ID 31 ya que fue el que más devoluciones registro y peor fue calificado por los clientes.
5. Enfocarse en vender partes para computadora, ya que artículos como bocinas, pantallas, audífonos y memorias usb no se están vendiendo.
6. El producto que mejor se vende y más ingresos deja es el articulo con Id 3, el articulo que más se vende es el que tiene Id 54.
7. El primer trimestre del 2020 fue el que más ventas reportó.

Definición del código

Login

Debido a que no se especifico las funciones de los roles administrador y usuario, ambos pueden hacer lo mismo.

Para entrar en el sistema solo este dado de alta un administrador con las siguientes credenciales:

Usuario: **Admin** (Incluyendo la mayúscula)

Contraseña: **root** (en minúsculas)

En caso de querer entrar con un usuario este debe de registrarse y posteriormente logearse.

```
 --------------------
|   LIFESTORE®       |
 --------------------
¿Estas Registrado? s/n (Ingresa q para salir):
```

El sistema pregunta si el usuario o administrador esta registrado, ingresar la letra s de sí, en caso de tener una cuenta o la n de no, en caso de no tener cuenta y querer registrarse.

```
¿Estas Registrado? s/n (Ingresa q para salir): n
Introduce un nombre de usuario: Elias
Introduce una contraseña: Elias123

Usuario creado satisfactoriamente!
```

De esa forma se registra un usuario en el sistema.

```
 --------------------
|   LIFESTORE®       |
 --------------------
¿Estas Registrado? s/n (Ingresa q para salir): n
Introduce un nombre de usuario: Elias

Ese usuario ya existe!
```

Si queremos crear un usuario que ya existe el sistema no lo permitirá.

```
 --------------------
|   LIFESTORE®       |
 --------------------
¿Estas Registrado? s/n (Ingresa q para salir): s
Ingresa tu usuario: Admin
Ingresa tu contraseña: root

====> Has iniciado sesión como  Admin ☺
-------------------------------------------
```

De esa forma podemos entrar al sistema como Admin.

```
 ---------------------------MENU-------------------------------------------------
 |  1.- Lista de los 50 Productos con Mayores Ventas                             |
 |  2.- Lista de los 100 Productos con Mayores Busquedas                         |
 |  3.- Lista de los 50 Productos con Menores Ventas (Agrupados por categoría)   |
 |  4.- Lista de los 100 Productos con Menores Busquedas (Agrupados por categoría) |
 |  5.- Lista de los Productos con Mejor Reseña (Unicamente con 5 de score)      |
 |  6.- Lista de los Productos con Mejor Reseña (5 o 4 de Score)                 |
 |  7.- Lista de los Productos con Peor Reseña (1 o 2 de Score)                  |
 |  8.- Lista de los Productos con Peor Reseña (Unicamente con 1 de score)       |
 |  9.- TOTAL DE INGRESOS, DINERO EN DEVOLUCIONES                                |
 |  10.- VENTAS POR MES Y PROMEDIO MENSUAL                                       |
 |  11.- VENTAS QUE NO ESTAN DENTRO DEL AÑO 2020                                 |
 |  12.- PRODUCTOS DEVUELTOS                                                     |
 |  13.- LISTA DE PRODUCTOS CON MENORES VENTAS Y MAYOR STOCK                     |
 |  14.- VALOR DEL INVENTARIO POR PRODUCTO                                       |
 |  15.- VENTAS POR CATEGORIA                                                    |
 |  16.- PRODUCTOS CON MENOR BUSQUEDA Y MAYOR STOCK                              |
 |  0- Salir                                                                     |
 -------------------------------------------------------------------------------
 Selecciona una opcion...
```

El sistema se compone de este menú, una vez que se haya validado el usuario, accedera al bloque while donde se encuentra este menu.

```
while opcion != "0":
```

Funciona mediante ese while y a menos que la variable opción sea diferente de 0, se seguirá ejecutando.

```
opcion=input("Selecciona una opcion...")
if opcion == '1':
```

Dependiendo del valor de la variable opción se ejecutará una instrucción de código.

```
else:
    print("INGRESA UNA OPCION VALIDA DEL MENU!!!")
```

Si el usuario ingresa un valor que no este dentro del rango de las opciones del menú, imprimirá este mensaje, pero seguirá dentro del bucle while, ya que la variable opcion no vale 0.

```
Selecciona una opcion...1
=====>NO SE INCLUYEN PRODUCTOS CON DEVOLUCION
--ID--||-------------PRODUCTOS-------------------|| -----------------VENTAS---------------||------STOCK---
1 .- 54  ||  SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm  ||  49  ||  300  ||
2 .- 3  ||  Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth  ||  42  ||  987  ||
3 .- 5  ||  Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)  ||  20  ||  130  ||
4 .- 42  ||  Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD  ||  18  ||  0  ||
5 .- 57  ||  SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm  ||  15  ||  15  ||
6 .- 4  ||  Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire  ||  13  ||  295  |
7 .- 29  ||  Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD  ||  13  ||  10  ||
8 .- 2  ||  Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth  ||  12  ||  182  ||
9 .- 47  ||  SSD XPG SX8200 Pro, 256GB, PCI Express, M.2  ||  11  ||  8  ||
10 .- 12  ||  Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0  ||  9  ||  0  ||
```

De esa forma se imprimen la mayoría de las listas, algunas con mas campos otras con menos.

```python
productos_venta = []
#Itero sobre los productos, por cada producto itero sobre las ventas y mediante un if identifico cuales son las coincidencias
for n in range(len(lifestore_products)):
    y = [] #En esta lista iran los productos y sus respectivas ventas, reseteo su valor para que cada producto tenga su sublista con su nombre y sus respectivas
    ventas
    x=0
    id_p = lifestore_products[n][0]
    name = lifestore_products[n][1]
    stock = lifestore_products[n][4]
    for j in range(len(lifestore_sales)):
        if (lifestore_products[n][0] == lifestore_sales[j][1]) and (lifestore_sales[j][4] == 0): #Al añadir esta condicion no tomara en cuenta a los productos
        con devolucion
            x+=1 #Al encontrar una coincidencia en el Id_Producto de ambas listas le suma uno a la variable x, al final tendra el valor de las veces que se
            vendio dicho producto
    y.append(id_p)
    y.append(name) #Añado el nombre del producto y = ["Nombre_Prod"]
    y.append(x) #Añado el total de ventas del producto y = ["Nombre_Prod", no_Ventas]
    y.append(stock) #Añado el stock y = ["Nombre_Prod", no_Ventas, Stock]
    productos_venta.append(y) #Añado esa lista a otra lista, donde iran todos los productos y sus ventas
```

El código en su mayoría viene comentando explicando el proceso que sigue, cual es el valor que van adquiriendo las variables en cada iteración y cuál es la función de cada variable.

La lógica que use para iterar sobre las listas resumida en pasos fue:

1. Iterar sobre la lista productos (for n in range(len(lifestore_products)
2. Para acceder a una sublista utilizamos subíndices [n][j] (n: posición de la lista, j: elemento de la lista)
3. Para acceder al segundo elemento de la tercer sublista los índices serian -> [1][2] (Las posiciones inician en 0)
4. Para cada producto, iterar sobre la lista de ventas buscando una coincidencia mediante el campo Id Product (El campo que relaciona las 3 listas)
5. Si la sublista sobre la que se itera de lifestore_sales coincide en el Id_Prod sumamos +1 a una variable temporal
6. Añadimos todos los campos en orden que nos interesan a una lista(Y), después la lista resultante la añadimos a otra lista(Z), cada que se ejecute el for declaramos la lista(Y) vacía para que se llene con la información de un producto diferente

```
if (lifestore_products[n][0] == lifestore_sales[j][1]) and (lifestore_sales[
j][4] == 0):
```

Esa es la condición que compara el id_proct, para cada producto de la lista de productos iterara sobre toda la lista de ventas buscando una coincidencia.

También se evalúa la posición 4 de las sublistas de venta para comprobar que el producto no fue devuelto, si quisiéramos contar también a los productos devueltos quitamos eso de la condición o si queremos únicamente productos con devolución cambiamos ese cero por uno.

```
y.append(id_p)
y.append(name) #Añado el nombre del producto y = ["Nombre_Prod"]
y.append(x) #Añado el total de ventas del producto y = ["Nombre_Prod", no_Ventas]
y.append(stock) #Añado el stock y = ["Nombre_Prod", no_Ventas, Stock]
productos_venta.append(y) #Añado esa lista a otra lista, donde iran todos los productos y sus ventas
```

```
#Utilizo el metodo bubble sort y le agrego el parametro reverse pa
l = len(productos_venta)
for i in range(0, l):
    for j in range(0, l-i-1):
        if (productos_venta[j][2] < productos_venta[j + 1][2]):
            tempo = productos_venta[j]
            productos_venta[j]= productos_venta[j + 1]
            productos_venta[j + 1]= tempo
```

De esta forma ordenamos nuestra lista de listas, a través de un bubble sort, en el cual indicamos la posición de la sublista que queremos ordenar, si de forma ascendente o descendente.

Aquí hemos utilizado la técnica de Bubble Sort para realizar la clasificación. Hemos intentado acceder al tercer elemento de las sublistas utilizando los bucles anidados. Esto realiza el método de clasificación in-place. La complejidad del tiempo es similar al bubble sorting, es decir, O (n ^ 2)

```
for n in range(50):
    print(n+1,".-", end=" ")
    for j in range(len(productos_venta[n])):
        print(productos_venta[n][j], end=" ")
        print(" || ", end= ' ')
    print()
```

Utilizamos un for para imprimir en consola los valores de nuestra lista, el parámetro end="" omite el salto de línea, se une con el siguiente print, la unión será el valor que le indiquemos a ese parámetro. Recorremos la lista de la misma forma, ya que también es una lista de listas, por estética se imprimirán dos barras || para separar cada elemento.

```
for n in range(len(lifestore_products)):
    y = []
    x=0
    name = lifestore_products[n][1]
    for j in range(len(lifestore_searches)):
        if lifestore_products[n][0] == lifestore_searches[j][1]:
            x+=1
    y.append(lifestore_products[n][0])
    y.append(name)
    y.append(x)
    productos_busqueda.append(y)
#productos_busqueda es la lista de listas que contiene el producto y
```

Usamos exactamente la misma lógica para la búsqueda de los productos.

Lo único que varía con las demás listas es la forma en que la ordenamos antes de imprimirla o la cantidad de campos que agregamos a la lista, en algunos añadimos categoría, en otros id.

```
productos_con_mejorevaluacion = []
for n in range(len(lifestore_products)):
    y = []
    x=0
    name = lifestore_products[n][1]
    for j in range(len(lifestore_sales)):
        if (lifestore_products[n][0] == lifestore_sales[j][1]) and (lifestore_sales[j][2] == 5 or lifestore_sales[j][2] == 4):
            x+=1
    if x > 0:
        y.append(lifestore_products[n][0])
        y.append(name)
        y.append(x)
        productos_con_mejorevaluacion.append(y)
```

Para la lista de calificaciones lo único que cambia es la condición del if, además de buscar una coincidencia en el Id Product también busca que la venta en el campo de la Score tenga un valor de 4 o 5.

Ya que hay productos que retornaran un valor de 0 para la variable x (variable temporal) en esa condición por lo que solo tomo en cuenta aquellas que su valor sea mayor a 0.

Dentro del código se repite el nombre de las variables, esto porque son variables temporales que solo existen dentro del bucle donde son creadas.

Para las demás calificaciones podemos reutilizar el código y modificar los valores de 4 y 5, para poner las calificaciones que nos interesen analizar.

Por ejemplo 1 y 2, o solamente 1.

```
productos_con_menorevaluacion = []
for n in range(len(lifestore_products)):
    y = []
    x=0
    name = lifestore_products[n][1]
    for j in range(len(lifestore_sales)):
        if (lifestore_products[n][0] == lifestore_sales[j][1]) and (lifestore_sales[j][2] == 1 or lifestore_sales[j][2] == 2):
            x+=1
    if x > 0:
        y.append(lifestore_products[n][0])
        y.append(name)
        y.append(x)
        productos_con_menorevaluacion.append(y)
```

```
productos_con_menorevaluacion = []
for n in range(len(lifestore_products)):
    y = []
    x=0
    name = lifestore_products[n][1]
    for j in range(len(lifestore_sales)):
        if (lifestore_products[n][0] == lifestore_sales[j][1]) and (lifestore_sales[j][2] == 1):
            x+=1
    if x > 0:
        y.append(lifestore_products[n][0])
        y.append(name)
        y.append(x)
        productos_con_menorevaluacion.append(y)
```

Lo único que hacemos es modificar el if, para imprimir la lista hacemos lo mismo, ya que la lista final sigue siendo una lista de listas, no importa el tamaño de las sublistas.

```
ingresos = []
#Itero sobre los productos, por cada producto itero sobre las ventas y mediante un if identi
for n in range(len(lifestore_products)):
    y = [] #En esta lista iran los productos y sus respectivas ventas, reseteo su valor para
    ventas
    x=0
    id_p = lifestore_products[n][0]
    name = lifestore_products[n][1]
    price = lifestore_products[n][2]
    for j in range(len(lifestore_sales)):
        if (lifestore_products[n][0] == lifestore_sales[j][1]) and (lifestore_sales[j][4] ==
        con devolucion
            x+=1 #Al encontrar una coincidencia en el Id_Producto de ambas listas le suma un
            vendio dicho producto
    y.append(id_p)
    y.append(name) #Añado el nombre del producto y = ["Nombre_Prod"]
    y.append(x) #Añado el total de ventas del producto y = ["Nombre_Prod", no_Ventas]
    y.append(price * x) #Añado el ingreso del producto y = ["Nombre_Prod", no_Ventas, ingres
    ingresos.append(y) #Añado esa lista a otra lista, donde iran todos los productos, ventas
```

Para calcular los ingresos usamos exactamente el mismo procedimiento que utlizamos en la lista de la venta de productos solo que al final agregamos un campo a la sublista que será el ingreso, compuesto del precio del producto multiplicado por la variable x (la variable que representa el número de ventas del producto).

```
y.append(price * x)
```

```
total_ingresos = 0
for n in range(len(ingresos)):
    total_ingresos+=ingresos[n][3]
```

Una vez hecha la lista de ingresos iteramos sobre la lista y en cada iteración sumamos el valor del elemento donde se encuentra el ingreso del producto, al final la variable tendrá el total de ingresos de todas las ventas.

```
meses = [0,0,0,0,0,0,0,0,0,0,0,0]
meses_ventas = [0,0,0,0,0,0,0,0,0,0,0,0]
#Itero sobre las ventas, por cada venta itero sobre los productos y mediante un if identifico cual
for n in range(len(lifestore_sales)):
    #y = [] #En esta lista iran los productos y sus respectivas ventas, reseteo su valor para que
    ventas
    x=0
    date = lifestore_sales[n][3]
    mes = date.split(sep='/')
    for j in range(len(lifestore_products)):
        if (lifestore_sales[n][1] == lifestore_products[j][0]) and (lifestore_sales[n][4] == 0):
            if mes[1] == '01':
                meses_ventas[0]+=lifestore_products[j][2]
                meses[0]+=1
            if mes[1] == '02':
                meses_ventas[1]+=lifestore_products[j][2]
                meses[1]+=1
```

Para obtener las ventas por mes, primero definimos dos listas, ambas con doce valores una tendrá el numero de ventas por mes y la otra tendrá los ingresos por mes.

Iteramos sobre la lista ventas y para cada venta iteramos sobre cada producto de la lista productos para obtener el precio del producto que se vendio

Tomamos una venta, seleccionamos la posición que contiene la fecha, como es un string utlizamos el método Split para dividirla, indicamos en el parámetro sep, que vamos a dividir ese string por barras /. El resultado sera una lista de 3 posiciones donde la posición del mes será la segunda, la que tiene el índice 1.

Con el if indicamos que si hay una coincidencia en el id del producto y la venta no esta marcada como devolución proceda.

Si el string de la fecha, que dividimos en la posición mes (en el código este valor es mes[1]) tiene el valor '01' quiere decir que la venta fue en enero.

La lista de meses sumara una venta en la posición del mes que le corresponde, si es en enero sumara 1 a la posición 0, si es en febrero sumara 1 a la posición 1, y asi consecutivamente con todos los meses.

La lista meses_ventas también insertara valores en en orden, posición 0 para enero, 1 para febrero, 2 para marzo y asi consecutivamente. Solo que esta lista sumara el precio del producto que cumplio con el if, ósea el producto correspondiente a la venta sobre la cual se itero. Al final cada posición sumara los ingresos de su respectivo mes.

```
print("--MES--||----NUMERO DE VENTAS---||--TOTAL INGRESOS POR MES--")
print()
print("Enero: ||", meses[0]," || $",meses_ventas[0])
print("Febrero: ||", meses[1]," || $",meses_ventas[1])
print("Marzo: ||", meses[2]," || $",meses_ventas[2])
print("Abril: ||", meses[3]," || $",meses_ventas[3])
print("Mayo: ||", meses[4]," || $",meses_ventas[4])
print("Junio: ||", meses[5]," || $",meses_ventas[5])
print("Julio: ||", meses[6]," || $",meses_ventas[6])
print("Agosto: ||",meses[7]," || $",meses_ventas[7])
print("Septiembre: ||", meses[8]," || $",meses_ventas[8])
print("Octubre: ||", meses[9]," || $",meses_ventas[9])
print("Noviembre: ||", meses[10]," || $",meses_ventas[10])
print("Diciembre: ||", meses[11]," || $",meses_ventas[11])
print()
print("-----------------------------------------------------------------------")
print("TOTALES: ",sum(meses)," || $",sum(meses_ventas)," ||")
print("-----------------------------------------------------------------------")
print("PROMEDIO MENSUAL: $", float((sum(meses_ventas)/12)))
print("-----------------------------------------------------------------------")
#FIN VENTAS POR MES#
```

Mientras iteraba me percate que solo existen dos ventas en la lista con un año diferente al 2020.

```
x=0
print("Ventas Registradas en un año diferente al 2020")
for n in range(len(lifestore_sales)):
    date = lifestore_sales[n][3]
    año = date.split(sep='/')
    for j in range(len(lifestore_products)):
        if (lifestore_sales[n][1] == lifestore_products[j][0]):
            if año[2] != '2020':
                print("Id_Venta: ",lifestore_sales[n][0], " Month: ",año[1], " Year: ",año[2])
                x+=lifestore_products[j][2]
print("Cantidad: ",x)
#FIN FECHAS#
```

Para esto solo puse que tomara en cuenta las ventas que, en el string de la fecha, ya dividido, en la posición del año, sea diferente a “2020”.

```
Ventas Registradas en un año diferente al 2020
Id_Venta:  15  Month:  11  Year:  2019
Id_Venta:  219  Month:  05  Year:  2002
```

Para salir del sistema y terminar con el programa ingresamos 0 en la consola

Y luego ingresamos la letra q, en el login, de esa forma terminamos el programa.

LINK DEL REPOSITORIO EN GITHUB:

https://github.com/EliasBautista/Curso_Emtech.git