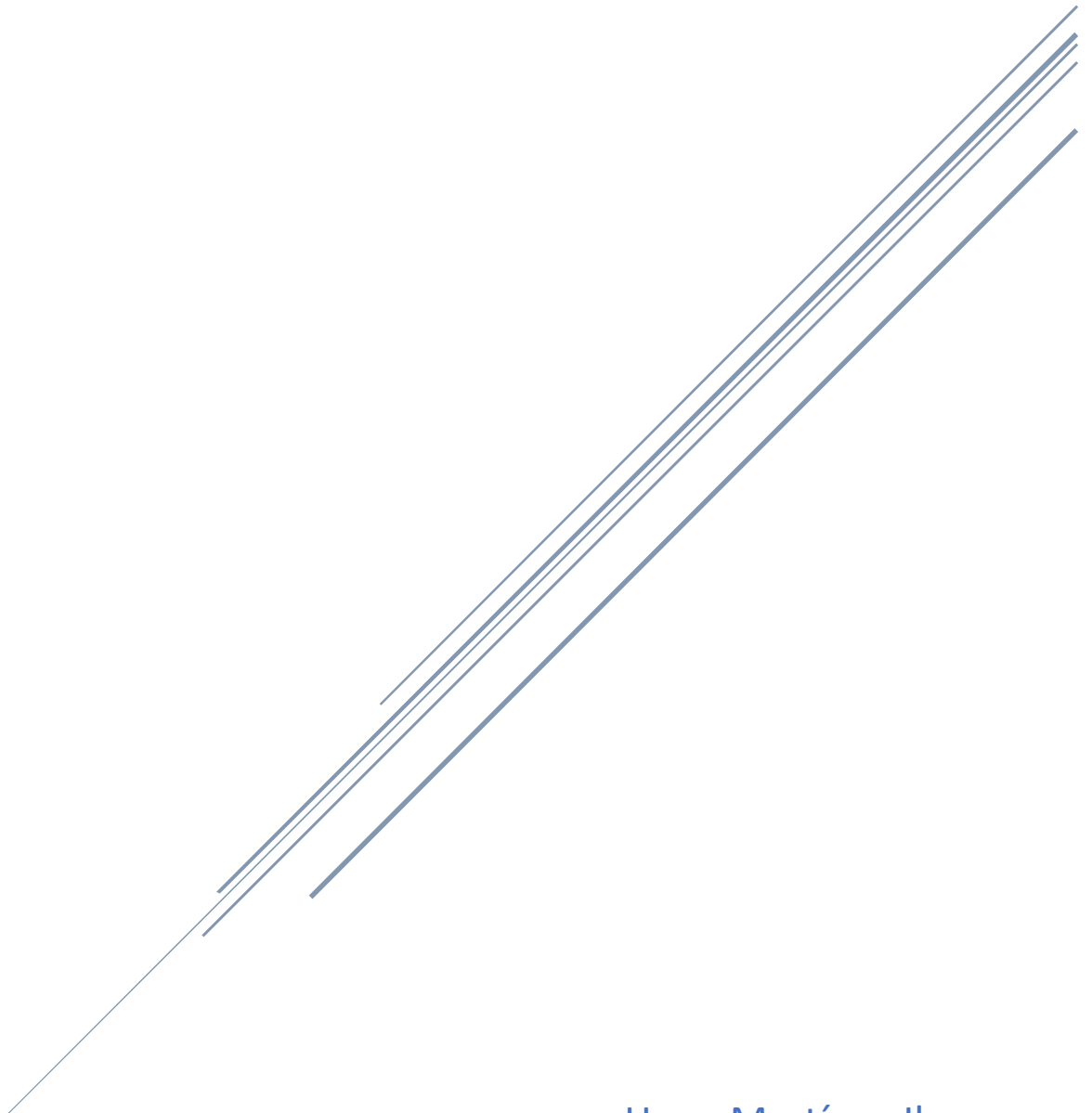


PROYECTO DE ANÁLISIS DE DATOS

Caso tienda LifeStore



Hugo Martínez Ibarra
Analista de datos

Índice

Introducción	2
Objetivo	2
Metodología.....	2
Consignas o tareas.....	3
Resultados.....	4
Solución.....	24
Conclusión.....	26
Código del programa.....	27

Introducción

La gestión de inventario es sin duda una de las áreas más importantes de cualquier negocio que esté comprometido con sus clientes y su visión de ser incrementar el valor de su empresa; una eficiente administración de sus productos permite satisfacer la demanda del mercado, pero además permite dirigir los recursos de la empresa de manera más inteligente en la operación e incluso en la planeación de proyectos.

Por tanto, el análisis de datos se ha vuelto una de las herramientas más valoradas y calificadas para participar en dicha tarea; su precisión cuantitativa y la disponibilidad de información que ofrece la convierten en la ruta adecuada para estudiar y dar solución a problemas como la acumulación de inventario. Es precisamente ese tipo de problema, el cual será abordado con la potencia de cómputo y habilidad de programación al reconocer las principales variables, su relación entre ellas y el tipo de información disponible para plantear una recomendación que pueda ser implementada en el modelo de negocio con el objetivo de lograr el mejor balance entre venta y disponibilidad de productos.

La tarea técnica consiste en visualizar y encontrar la mejor forma de extraer, organizar y filtrar la información relevante a ser analizada a partir de los registros de la base de datos disponible. En esa tarea estarán involucrados una serie de algoritmos que van desde la detección de meses con registro de ventas hasta el cálculo de ingreso neto por mes y por cada producto. Para llevar a cabo esta tarea, se ha empleado el lenguaje de programación Python que es bastante versátil y de propósito general en las tareas que puede desarrollar.

Después de haber extraído, clasificado y ordenado la información se presentarán los datos y relaciones más importante en forma de recursos de estadística descriptiva como graficas de barras de acuerdo a una serie de consignas relevantes para llevar a cabo un análisis y emitir juicios.

Casi al final, se hará una serie de recomendaciones y otros comentarios necesarios sobre la situación actual y futura de la empresa para resolver el problema de acumulación de inventario.

Finalmente se entregará un programa hecho en Python que contiene todos los algoritmos mencionados y es capaz de presentar la información de manera ordenada y clara, con opciones de visualización detalladas y opciones extra para el administrador del sistema.

Objetivo

Construir un programa en el lenguaje de programación de Python que sea capaz de extraer, clasificar, ordenar y presentar información de manera ordenada (a partir de una base de datos disponible) sobre las variables más importantes de los productos de un negocio y analizar con ayuda de estadística descriptiva la situación del negocio para resolver un problema de acumulación de inventario.

Metodología

1. Se identificarán las variables más importantes y disponibles a partir de la base de datos. Será necesario ver cómo se relacionan los datos para poder encontrar la mejor forma de extraerlos y clasificarlos.

2. Se extraerá la información más importante y se clasificará de la manera más eficiente posible, teniendo siempre en cuenta que dicho proceso sea lo más general y automatizado posible; es decir, que no dependa de la cantidad de datos, sólo de su forma.
3. Se filtrarán los datos más importantes y según la serie de consignas que se presentarán más adelante y las cuales ayudarán a resolver el problema en cuestión.
4. Se presentarán los datos de una forma clara y ordenada, así como ilustrativa para identificar los puntos clave que aportarán en la discusión sobre las decisiones a considerar.
5. Se presentará una serie de comentarios y recomendaciones para que el equipo de administración del negocio pueda implementarlas y resolver el problema.

Recordar que los registros de datos están organizados en listas de la siguiente forma (escrito en lenguaje de programación Python):

```
lifestore_products = [[id_product, name, price, category, stock]]  
lifestore_sales = [[id_sale, id_product, score (from 1 to 5), date, refund (1 for true and 0 to false)]]  
lifestore_searches = [[id_search, id_product]]
```

La fecha de la venta 15 se cambió a 10/01/2020 y la de la venta 219 a 04/05/2020 por posibles errores en el registro de las fechas.

Consignas a tratar (aquí se da un resumen de los resultados, en el programa se pueden consultar todos):

1. Conocer los productos con mayor cantidad de ventas totales. Aquí se consideran productos con devolución y sin devolución (*refund*), pues en la práctica todas ellas fueron ventas.
2. Conocer los productos con mayores búsquedas.
3. Identificar los productos con menores ventas por categoría. Aquí se pone especial atención a los que tuvieron ventas igual a cero, pues son los de mayor preocupación para la empresa.
4. Identificar los productos con menores búsquedas por categoría. Mismo comentario que en la consigna anterior.
5. Reconocer los productos con mejores reseñas. Debido que existe un sistema de calificaciones de 1 a 5, existe la opción de ver qué productos obtuvieron las mejores calificaciones (y cuántas) o calcular una calificación promedio, pues hay productos que recibieron hasta 3 calificaciones diferentes en todas las ventas hechas. Se analizarán ambas opciones.
6. Reconocer los productos con peores reseñas. Por la misma situación explicada antes, se pondrá atención en los productos con mayor cantidad de calificaciones de valor 1, y en la otra opción a los productos con calificación promedio menor.

7. Calcular el total de ingresos mensuales. Aquí se hallará el monto de ingresos netos por mes, a partir de los ingresos por producto y por cada mes; es decir, el monto de ingresos generados por las ventas netas (sin devolución).

8. Calcular las ventas promedio mensuales. Será necesario contabilizar la cantidad de ventas total (de todos los productos y meses) de todo el registro disponible y enseguida dividirlo por la cantidad de meses que existe registro.

9. Calcular el total de ingresos anual. Esto será la cantidad de ingresos netos de todos los productos y meses que hay registro.

10. Averiguar los meses con más ventas al año. A partir de los datos de ventas por producto y mes se calcularán las ventas por mes y se ordenará respecto a dicho dato los meses.

Adicionalmente hay 3 consignas más a las que el Administrador tiene acceso, que son de ayuda para un análisis más amplio y son:

E1. Conocer los productos con menores ingresos por categoría.

E2. Productos ordenados de mayor a menos por cantidad de ingresos netos.

E3. Productos ordenados por mayor cantidad de stock. Aquí se considera el dato de stock como la cantidad disponible (restante) de cada producto al final de los registros disponibles.

E4. Razón de ventas/búsqueda y razón búsqueda/ventas.

Nota: No se ha elegido una cantidad definida (salvo en la consigna 5 y 6) porque en muchos casos los resultados que cumplen esas características no son muchos y es mejor visualizar todos en contexto con el resto de datos. Sólo en ciertos casos, se ignoraron los productos con valor observado igual a cero, porque para la consigna en cuestión no eran relevantes. En otros sí son importantes como los productos con menores ventas o búsquedas. Los resultados de todas las consignas se dan con detalle en el programa.

Para extraer, clasificar y ordenar eficientemente la información se construyeron 3 listas esenciales dentro del programa y de forma automática:

```
product_month_sales = [[id_product, reduced_name, category, [month_number, sales, refunds]]]
product_scores = [[id_product, reduced_name, category, [score, number_scores]]]
product_searches = [[id_product, reduced_name, category, searches]
```

La siguiente también fue bastante necesaria, pero surge de la base de datos inicial y la primera lista:

```
product_month_incomes = [[id_product, reduced_name, category, [month_number, incomes]]]
```

Resultados de tareas o consignas

Se muestran los productos ordenados de mayor a menor cantidad de ventas (se consideran igualmente los que tuvieron devolución, pues fueron ventas en la práctica). Sólo se consideran ventas mayores o igual a uno.

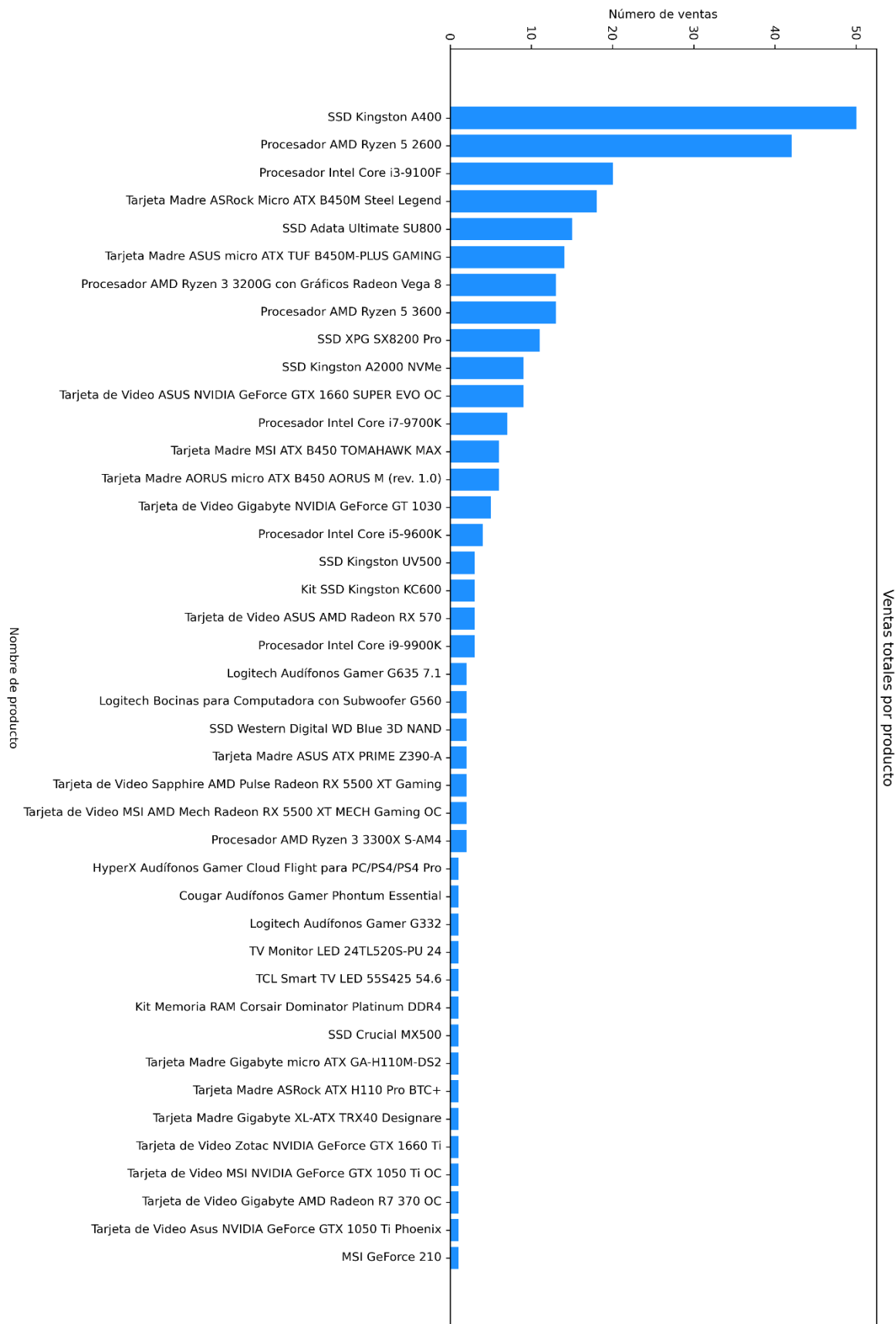


Tabla 1: Productos ordenados menor a mayor cantidad de ventas por categoria.

ID	Nombre de producto	No. Ventas
Categoria	1	procesadores
9	Procesador Intel Core i3-8100	0
1	Procesador AMD Ryzen 3 3300X S-AM4	2
6	Procesador Intel Core i9-9900K	3
8	Procesador Intel Core i5-9600K	4
7	Procesador Intel Core i7-9700K	7
4	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8	13
2	Procesador AMD Ryzen 5 3600	13
5	Procesador Intel Core i3-9100F	20
3	Procesador AMD Ryzen 5 2600	42
Categoria	2	tarjetas de video
27	Tarjeta de Video VisionTek AMD Radeon HD5450	0
26	Tarjeta de Video VisionTek AMD Radeon HD 5450	0
24	Tarjeta de Video PNY NVIDIA GeForce RTX 2080	0
23	Tarjeta de Video MSI Radeon X1550	0
20	Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC	0
19	Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile	0
16	Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming	0
15	Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming	0
14	Tarjeta de Video EVGA NVIDIA GeForce GT 710	0
28	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti	1
22	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC	1
17	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC	1
13	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix	1
10	MSI GeForce 210	1
25	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming	2
21	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC	2
11	Tarjeta de Video ASUS AMD Radeon RX 570	3
18	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030	5
12	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC	9
Categoria	3	tarjetas madre
43	Tarjeta Madre ASUS ATX ROG STRIX Z390-E GAMING	0
41	Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM	0
39	ASUS T. Madre uATX M4A88T-M	0
38	Tarjeta Madre Gigabyte Micro ATX H310M DS2 2.0	0
37	Tarjeta Madre ASRock ATX Z490 STEEL LEGEND	0
36	Tarjeta Madre Gigabyte micro ATX Z490M GAMING X (rev. 1.0)	0

35	Tarjeta Madre Gigabyte micro ATX Z390 M GAMING	0
34	Tarjeta Madre ASUS ATX ROG STRIX B550-F GAMING WI-FI	0
32	Tarjeta Madre ASRock Z390 Phantom Gaming 4	0
30	Tarjeta Madre AORUS ATX Z390 ELITE	0
46	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2	1
45	Tarjeta Madre ASRock ATX H110 Pro BTC+	1
40	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare	1
33	Tarjeta Madre ASUS ATX PRIME Z390-A	2
44	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX	6
31	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0)	6
29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING	14
42	Tarjeta Madre ASRock Micro ATX B450M Steel Legend	18
Categoria 4 discos duros		
59	SSD Samsung 860 EVO	0
58	SSD para Servidor Lenovo Thinksystem S4510	0
56	SSD para Servidor Lenovo Thinksystem S4500	0
55	SSD para Servidor Supermicro SSD-DM128-SMCMVN1	0
53	SSD Addlink Technology S70	0
50	SSD Crucial MX500	1
52	SSD Western Digital WD Blue 3D NAND	2
51	SSD Kingston UV500	3
49	Kit SSD Kingston KC600	3
48	SSD Kingston A2000 NVMe	9
47	SSD XPG SX8200 Pro	11
57	SSD Adata Ultimate SU800	15
54	SSD Kingston A400	50
Categoria 5 memorias usb		
61	Kit Memoria RAM Corsair Vengeance LPX DDR4	0
60	Kit Memoria RAM Corsair Dominator Platinum DDR4	1
Categoria 6 pantallas		
73	Samsung Smart TV LED UN55TU7000FXZX 55	0
72	Hisense Smart TV LED 50H8F 49.5	0
71	Samsung Smart TV LED UN32J4290AF 32	0
70	Samsung Smart TV LED 43	0
69	Hisense Smart TV LED 40H5500F 39.5	0
68	Makena Smart TV LED 40S2 40"	0
65	Samsung Smart TV LED UN70RU7100FXZX 70	0
64	Samsung TV LED LH43QMREBGCXGO 43	0
63	Seiki TV LED SC-39HS950N 38.5	0
62	Makena Smart TV LED 32S2 32"	0
67	TV Monitor LED 24TL520S-PU 24	1

66	TCL Smart TV LED 55S425 54.6	1
Categoría	7	bocinas
83	Ghia Bocina Portátil BX500	0
82	Ghia Bocina Portátil BX400	0
81	Ghia Bocina Portátil BX900	0
80	Ghia Bocina Portátil BX800	0
79	Naceb Bocina Portátil NA-0301	0
78	Ghia Bocina Portátil BX300	0
77	Verbatim Bocina Portátil Mini	0
76	Acteck Bocina con Subwoofer AXF-290	0
75	Lenovo Barra de Sonido	0
74	Logitech Bocinas para Computadora con Subwoofer G560	2
Categoría	8	audifonos
96	Klip Xtreme Audífonos Blast	0
95	logear Audífonos Gamer GHG601	0
93	Ginga Audífonos con Micrófono GI18ADJ01BT-RO	0
92	Getttech Audífonos con Micrófono Sonority	0
91	Genius GHP-400S Audífonos	0
90	Energy Sistem Audífonos con Micrófono Headphones 1	0
88	Audífonos Gamer Balam Rush Orphix RGB 7.1	0
87	Acer Audífonos Gamer Galea 300	0
86	ASUS Audífonos Gamer ROG Theta 7.1	0
94	HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro	1
89	Cougar Audífonos Gamer Phontum Essential	1
84	Logitech Audífonos Gamer G332	1
85	Logitech Audífonos Gamer G635 7.1	2

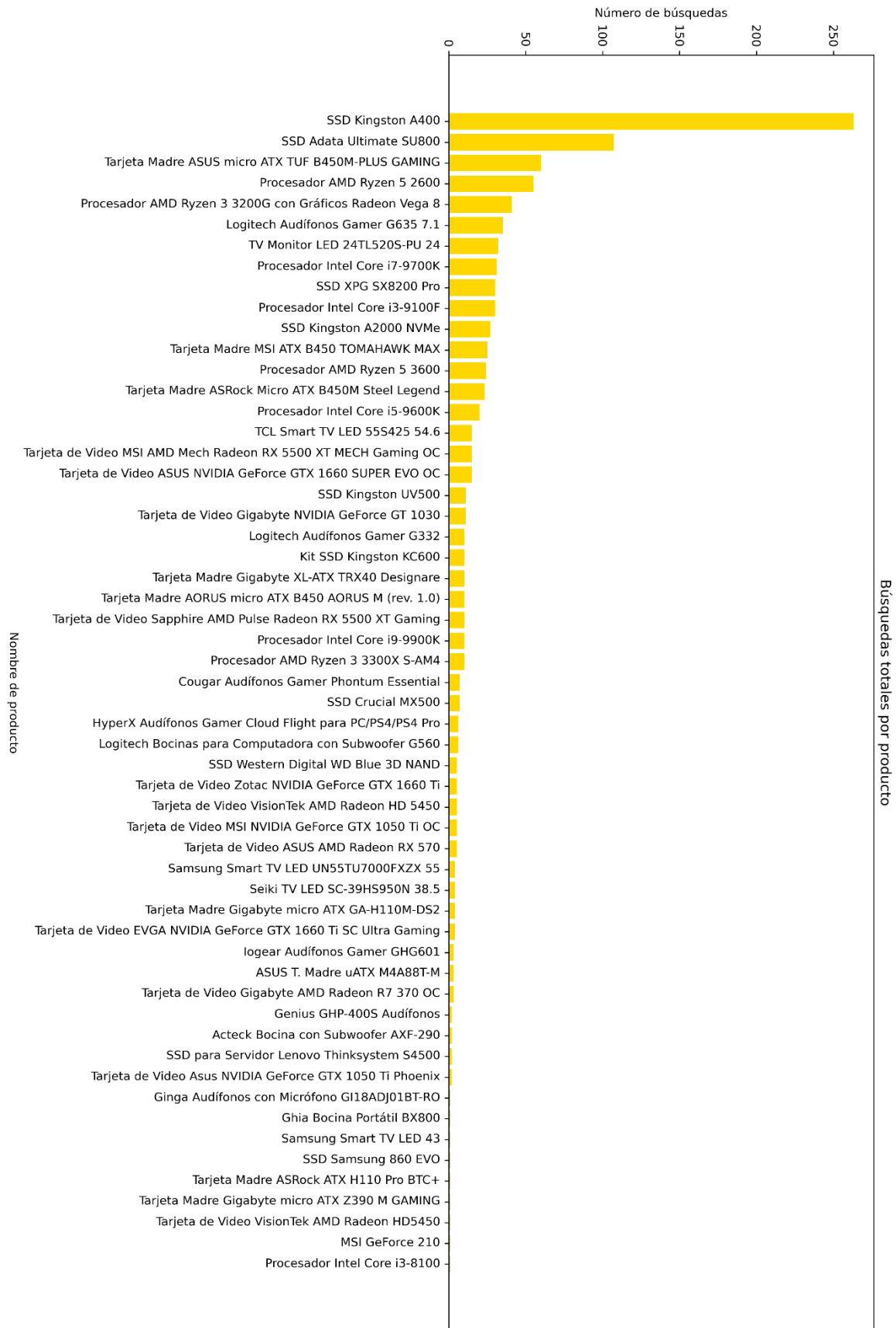
Se puede apreciar que entre los productos más vendidos y que aportan más ventas en general se encuentran procesadores, discos duros SSD y tarjetas madre.

Mientras que los productos sin ventas son en su mayoría bocinas, audífonos, pantallas, cerca de la mitad de tarjetas madre y de tarjetas de video.

Será importante ver qué relación existe entre la cantidad de ventas y las búsquedas registradas

En la siguiente página se muestran los productos ordenados de mayor a menor cantidad de búsquedas.

Cabe mencionar que estos resultados son búsquedas hechas en todo el tiempo de registro de ventas y no es posible conocer con precisión a que mes pertenecen. Sin embargo, existe cierto comportamiento esperado, pues casi todos los primeros 10 productos más vendidos son también los más buscados. Existe una excepción que es la pantalla TV Monitor LED 24TL520S-PU 24.



Cabe mencionar que el número de búsquedas en la mayoría de los productos llega a ser poco más de 4 veces la cantidad de ventas hechas de ese producto. Lo que implica que aproximadamente un 25% de esas búsquedas llegan a ser ventas consolidadas.

Tabla 2: Productos ordenados menor a mayor cantidad de búsquedas por categoría.

ID	Nombre de producto	No. Búsquedas
Categoría	1	procesadores
9	Procesador Intel Core i3-8100	1
6	Procesador Intel Core i9-9900K	10
1	Procesador AMD Ryzen 3 3300X S-AM4	10
8	Procesador Intel Core i5-9600K	20
2	Procesador AMD Ryzen 5 3600	24
5	Procesador Intel Core i3-9100F	30
7	Procesador Intel Core i7-9700K	31
4	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8	41
3	Procesador AMD Ryzen 5 2600	55
Categoría	2	tarjetas de video
24	Tarjeta de Video PNY NVIDIA GeForce RTX 2080	0
23	Tarjeta de Video MSI Radeon X1550	0
20	Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC	0
19	Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile	0
16	Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming	0
14	Tarjeta de Video EVGA NVIDIA GeForce GT 710	0
27	Tarjeta de Video VisionTek AMD Radeon HD5450	1
10	MSI GeForce 210	1
13	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix	2
17	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC	3
15	Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming	4
28	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti	5
26	Tarjeta de Video VisionTek AMD Radeon HD 5450	5
22	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC	5
11	Tarjeta de Video ASUS AMD Radeon RX 570	5
25	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming	10
18	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030	11
21	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC	15
12	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC	15
Categoría	3	tarjetas madre
43	Tarjeta Madre ASUS ATX ROG STRIX Z390-E GAMING	0
41	Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM	0
38	Tarjeta Madre Gigabyte Micro ATX H310M DS2 2.0	0

37	Tarjeta Madre ASRock ATX Z490 STEEL LEGEND	0
36	Tarjeta Madre Gigabyte micro ATX Z490M GAMING X (rev. 1.0)	0
34	Tarjeta Madre ASUS ATX ROG STRIX B550-F GAMING WI-FI	0
33	Tarjeta Madre ASUS ATX PRIME Z390-A	0
32	Tarjeta Madre ASRock Z390 Phantom Gaming 4	0
30	Tarjeta Madre AORUS ATX Z390 ELITE	0
45	Tarjeta Madre ASRock ATX H110 Pro BTC+	1
35	Tarjeta Madre Gigabyte micro ATX Z390 M GAMING	1
39	ASUS T. Madre uATX M4A88T-M	3
46	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2	4
40	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare	10
31	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0)	10
42	Tarjeta Madre ASRock Micro ATX B450M Steel Legend	23
44	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX	25
29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING	60
Categoria 4 discos duros		
58	SSD para Servidor Lenovo Thinksystem S4510	0
55	SSD para Servidor Supermicro SSD-DM128-SMCMVN1	0
53	SSD Addlink Technology S70	0
59	SSD Samsung 860 EVO	1
56	SSD para Servidor Lenovo Thinksystem S4500	2
52	SSD Western Digital WD Blue 3D NAND	5
50	SSD Crucial MX500	7
49	Kit SSD Kingston KC600	10
51	SSD Kingston UV500	11
48	SSD Kingston A2000 NVMe	27
47	SSD XPG SX8200 Pro	30
57	SSD Adata Ultimate SU800	107
54	SSD Kingston A400	263
Categoria 5 memorias usb		
61	Kit Memoria RAM Corsair Vengeance LPX DDR4	0
60	Kit Memoria RAM Corsair Dominator Platinum DDR4	0
Categoria 6 pantallas		
72	Hisense Smart TV LED 50H8F 49.5	0
71	Samsung Smart TV LED UN32J4290AF 32	0
69	Hisense Smart TV LED 40H5500F 39.5	0
68	Makena Smart TV LED 40S2 40"	0
65	Samsung Smart TV LED UN70RU7100FXZX 70	0
64	Samsung TV LED LH43QMREBGXGO 43	0
62	Makena Smart TV LED 32S2 32"	0
70	Samsung Smart TV LED 43	1

73	Samsung Smart TV LED UN55TU7000FXZX 55	4
63	Seiki TV LED SC-39HS950N 38.5	4
66	TCL Smart TV LED 55S425 54.6	15
67	TV Monitor LED 24TL520S-PU 24	32
Categoria 7		bocinas
83	Ghia Bocina Portátil BX500	0
82	Ghia Bocina Portátil BX400	0
81	Ghia Bocina Portátil BX900	0
79	Naceb Bocina Portátil NA-0301	0
78	Ghia Bocina Portátil BX300	0
77	Verbatim Bocina Portátil Mini	0
75	Lenovo Barra de Sonido	0
80	Ghia Bocina Portátil BX800	1
76	Acteck Bocina con Subwoofer AXF-290	2
74	Logitech Bocinas para Computadora con Subwoofer G560	6
Categoria 8		audifonos
96	Klip Xtreme Audífonos Blast	0
92	Getttech Audífonos con Micrófono Sonority	0
90	Energy Sistem Audífonos con Micrófono Headphones 1	0
88	Audífonos Gamer Balam Rush Orphix RGB 7.1	0
87	Acer Audífonos Gamer Galea 300	0
86	ASUS Audífonos Gamer ROG Theta 7.1	0
93	Ginga Audífonos con Micrófono GI18ADJ01BT-RO	1
91	Genius GHP-400S Audífonos	2
95	logear Audífonos Gamer GHG601	3
94	HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro	6
89	Cougar Audífonos Gamer Phontum Essential	7
84	Logitech Audífonos Gamer G332	10
85	Logitech Audífonos Gamer G635 7.1	35

Los productos sin búsquedas son en su mayoría bocinas, memorias usb, pantallas, tarjetas de video y la mitad de tarjetas madre y audífonos que no tiene al menos una búsqueda. También ciertas tarjetas madre y discos duros no fueron buscados en todo ese tiempo.

A continuación, se mostrarán resultados de la cantidad de calificaciones por producto, sólo de aquellos que tienen al menos una calificación dentro del rango 1 a 5, evidentemente los productos que no aparecen ahí no concluyeron en una venta y por tanto no fueron calificados.

Después se presentará una tabla de los mismos productos, pero con una calificación promedio calculada (se redondeó a un dígito después del punto) como:

$$\text{Calificación promedio de producto} = \frac{5(\text{notas de 5}) + 4(\text{notas de 4}) + 3(\text{notas de 3}) + 2(\text{notas de 2}) + 1(\text{notas de 1})}{\text{número de calificaciones (ventas)}}$$



Tabla 3: Calificación promedio de productos vendidos (redondeado a un dígito después del punto).

ID	Nombre de producto	Calificación promedio
22	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC	5
28	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti	5
40	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare	5
50	SSD Crucial MX500	5
60	Kit Memoria RAM Corsair Dominator Platinum DDR4	5
66	TCL Smart TV LED 55S425 54.6	5
67	TV Monitor LED 24TL520S-PU 24	5
84	Logitech Audífonos Gamer G332	5
1	Procesador AMD Ryzen 3 3300X S-AM4	5
21	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC	5
25	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming	5
52	SSD Western Digital WD Blue 3D NAND	5
85	Logitech Audífonos Gamer G635 7.1	5
6	Procesador Intel Core i9-9900K	5
11	Tarjeta de Video ASUS AMD Radeon RX 570	5
49	Kit SSD Kingston KC600	5
8	Procesador Intel Core i5-9600K	5
7	Procesador Intel Core i7-9700K	5
57	SSD Adata Ultimate SU800	4.9
12	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC	4.8
3	Procesador AMD Ryzen 5 2600	4.8
51	SSD Kingston UV500	4.7
44	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX	4.7
48	SSD Kingston A2000 NVMe	4.7
5	Procesador Intel Core i3-9100F	4.7
54	SSD Kingston A400	4.7
42	Tarjeta Madre ASRock Micro ATX B450M Steel Legend	4.6
33	Tarjeta Madre ASUS ATX PRIME Z390-A	4.5
74	Logitech Bocinas para Computadora con Subwoofer G560	4.5
4	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8	4.5
47	SSD XPG SX8200 Pro	4.5
18	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030	4.4
2	Procesador AMD Ryzen 5 3600	4.2
29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING	4.1
10	MSI GeForce 210	4
13	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix	4
94	HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro	4

89	Cougar Audífonos Gamer Phontum Essential	3
46	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2	2
31	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0)	1.8
17	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC	1
45	Tarjeta Madre ASRock ATX H110 Pro BTC+	1

Prácticamente todos los productos vendidos tienen calificaciones superiores a 4. De los 42 productos diferentes vendidos, 18 son de calificación prácticamente de 5, 19 de calificación superior a 4 y menor a 5, y 5 de calificación menor a 4. Lo anterior se puede ver de la siguiente forma:

Grado de satisfacción	Calificación	Cantidad de productos
Prácticamente del 100%	5	18
Superior o igual al 75% pero menor del 100%	$4 \leq \text{calificación} < 5$	19
Inferior al 75%	$1 \leq \text{calificación} < 4$	5

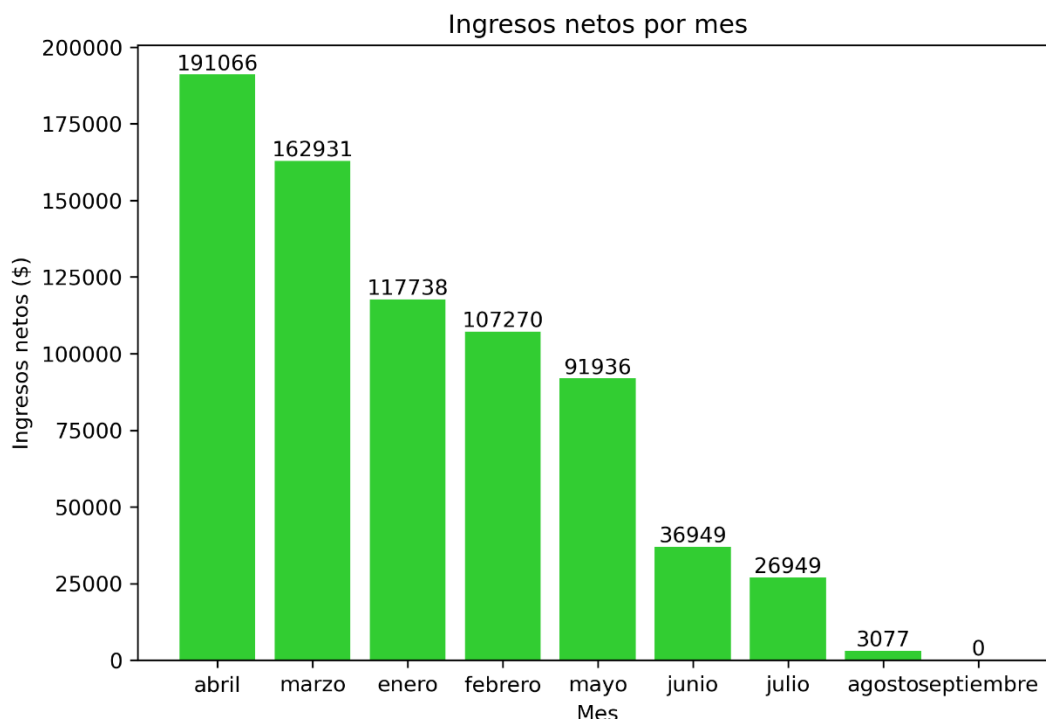
Mientras que de la gráfica de barras se puede observar que los productos más comprados tienen calificaciones (notas) en su mayoría de 5 y 4, a excepción del producto Tarjeta Madre ASUS ATX ROG STRIX Z390-E GAMING que tiene algunas calificaciones de 1.

Los productos que son comprados poco, no obstante, son bien calificados.

Cabe mencionar que sólo la Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0) tiene más de una compra y la mayoría son calificaciones de 1, se podría considerar este como el producto con las peores reseñas.

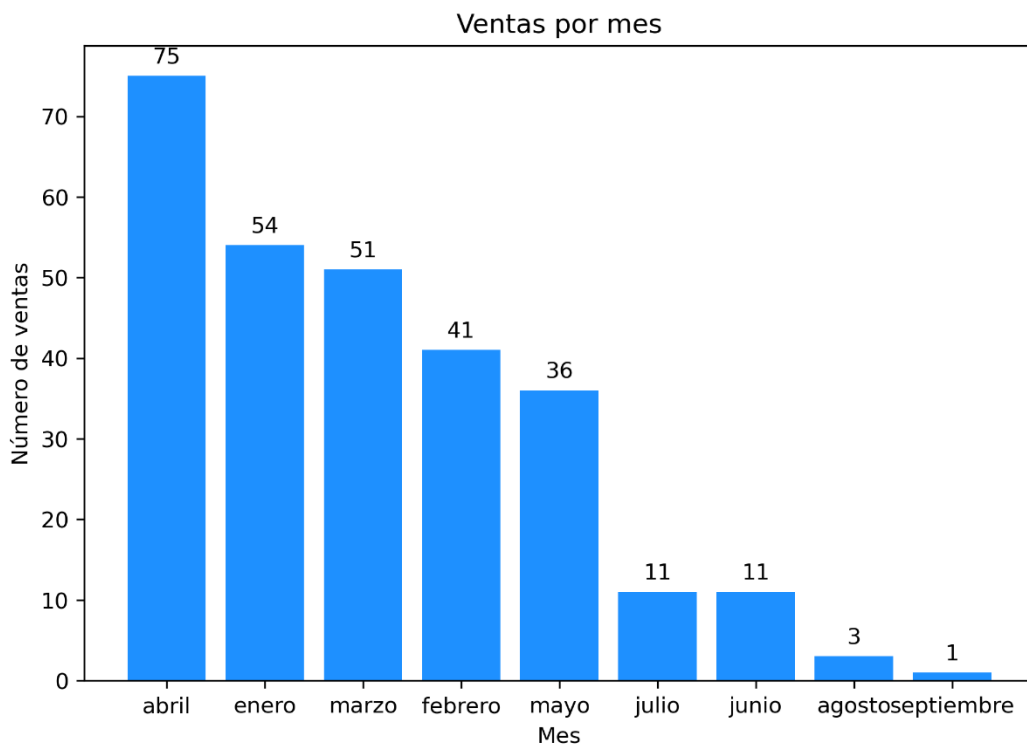
Ahora se presentarán resultados que conciernen a los ingresos netos (sin considerar ventas de las que hubo posteriormente una devolución).

En primer lugar, se muestra una gráfica que exhibe los montos de ingresos netos por mes:



Los tres meses con más ingresos son claramente abril, marzo y enero; con una pequeña diferencia entre enero, febrero y mayo. Junio, julio y agosto con los ingresos más bajos, pero septiembre sin ingresos.

Ahora se va a comparar con la cantidad de ventas por mes, mediante la siguiente gráfica.



Es importante mencionar que enero presenta mayor cantidad de ventas, pero no así de ingresos netos, comparado con marzo. Por otra parte, septiembre sólo presenta una venta, pero esta tuvo devolución, por eso no cuenta en los ingresos netos.

Los siguientes datos son de bastante utilidad para tener en contexto las cantidades anteriores.

Monto anual de ingresos netos (sin contar productos con devolución): \$ 737916

Monto promedio de ingresos netos mensuales: \$ 81990.67

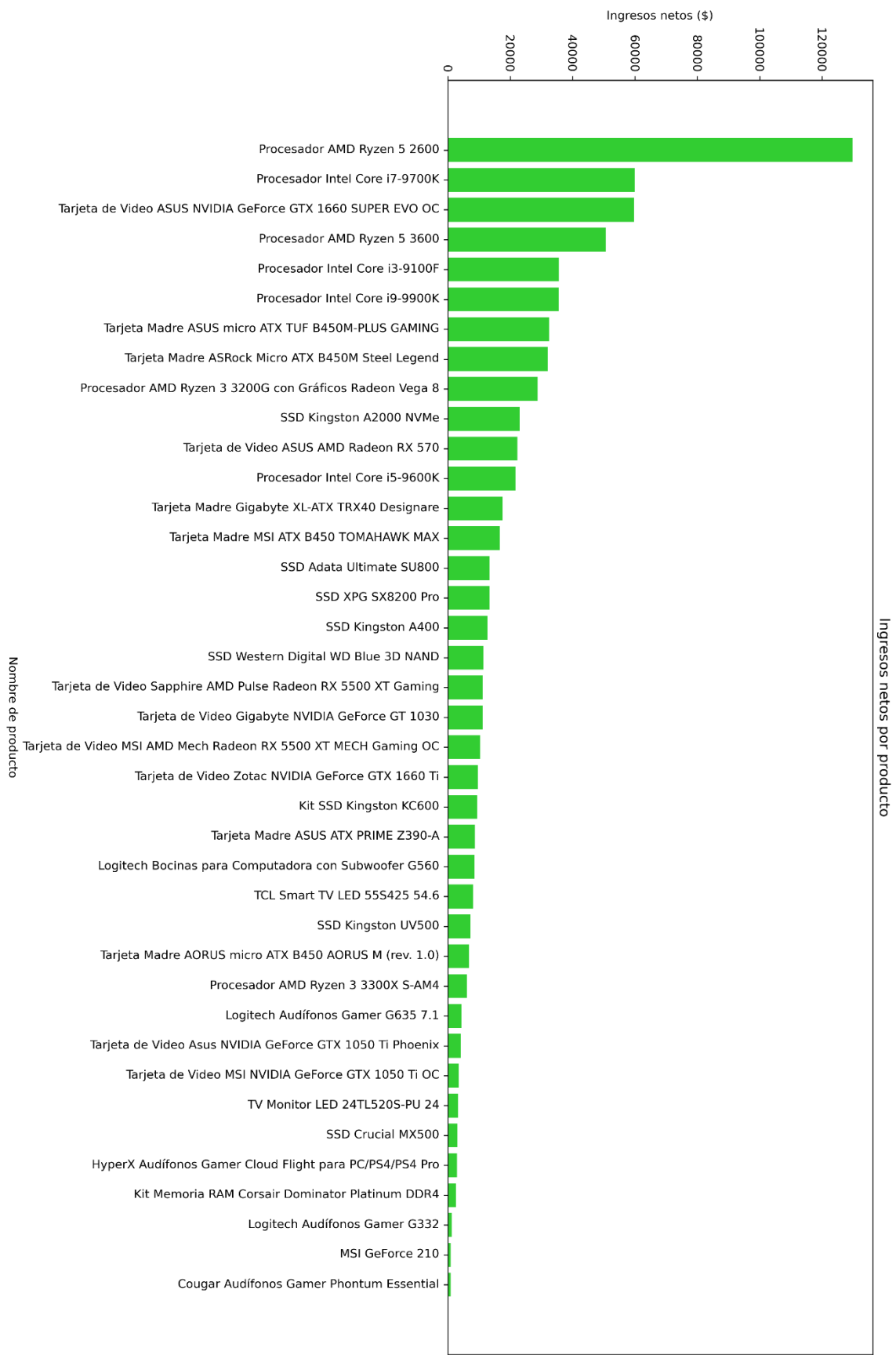
Cantidad de ventas totales: 283

Monto promedio de ventas mensuales: 31.44

Con lo anterior, se tiene que el mes de abril representa el 25.89% del total de ingresos de todo este tiempo y el 26.50% aproximadamente del total de ventas.

Por otra parte, la diferencia de ventas entre el mes de marzo y enero representa unos ingresos de \$45193, poco más que los ingresos totales de junio.

Ahora se verá una gráfica con los productos ordenados de mayor a menor cantidad de ingresos netos generados.



La tendencia que se observa es que los productos más vendidos son también los productos que generan más ingresos netos, donde la principal excepción es el producto TV Monitor LED 24TL520S-PU 24 que sólo tiene una venta, pero su ingreso neto generado es bastante alto.

Enseguida una tabla del stock de cada producto, que servirá para reconocer si se cuenta con la suficiente capacidad se seguir ofreciendo los productos más vendidos.

Tabla 5: Stock de todos los productos.

ID	Nombre de producto	Stock
3	Procesador AMD Ryzen 5 2600	987
67	TV Monitor LED 24TL520S-PU 24	411
54	SSD Kingston A400	300
4	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8	295
41	Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM	286
68	Makena Smart TV LED 40S2 40"	239
92	Getttech Audífonos con Micrófono Sonority	232
66	TCL Smart TV LED 55S425 54.6	188
2	Procesador AMD Ryzen 5 3600	182
26	Tarjeta de Video VisionTek AMD Radeon HD 5450	180
63	Seiki TV LED SC-39HS950N 38.5	146
93	Ginga Audífonos con Micrófono GI18ADJ01BT-RO	139
5	Procesador Intel Core i3-9100F	130
31	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0)	120
7	Procesador Intel Core i7-9700K	114
39	ASUS T. Madre uATX M4A88T-M	98
69	Hisense Smart TV LED 40H5500F 39.5	94
84	Logitech Audífonos Gamer G332	83
64	Samsung TV LED LH43QMREBGCXGO 43	71
37	Tarjeta Madre ASRock ATX Z490 STEEL LEGEND	60
6	Procesador Intel Core i9-9900K	54
48	SSD Kingston A2000 NVMe	50
30	Tarjeta Madre AORUS ATX Z390 ELITE	50
46	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2	49
33	Tarjeta Madre ASUS ATX PRIME Z390-A	43
27	Tarjeta de Video VisionTek AMD Radeon HD5450	43
85	Logitech Audífonos Gamer G635 7.1	39
14	Tarjeta de Video EVGA NVIDIA GeForce GT 710	36
9	Procesador Intel Core i3-8100	35
82	Ghia Bocina Portátil BX400	31

79	Naceb Bocina Portátil NA-0301	31
35	Tarjeta Madre Gigabyte micro ATX Z390 M GAMING	30
45	Tarjeta Madre ASRock ATX H110 Pro BTC+	25
86	ASUS Audífonos Gamer ROG Theta 7.1	20
81	Ghia Bocina Portátil BX900	20
76	Acteck Bocina con Subwoofer AXF-290	18
91	Genius GHP-400S Audífonos	16
83	Ghia Bocina Portátil BX500	16
58	SSD para Servidor Lenovo Thinksystem S4510	16
1	Procesador AMD Ryzen 3 3300X S-AM4	16
88	Audífonos Gamer Balam Rush Orphix RGB 7.1	15
80	Ghia Bocina Portátil BX800	15
57	SSD Adata Ultimate SU800	15
38	Tarjeta Madre Gigabyte Micro ATX H310M DS2 2.0	15
15	Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming	15
52	SSD Western Digital WD Blue 3D NAND	13
10	MSI GeForce 210	13
94	HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro	12
75	Lenovo Barra de Sonido	11
72	Hisense Smart TV LED 50H8F 49.5	11
70	Samsung Smart TV LED 43	10
60	Kit Memoria RAM Corsair Dominator Platinum DDR4	10
59	SSD Samsung 860 EVO	10
55	SSD para Servidor Supermicro SSD-DM128-SMCMVN1	10
36	Tarjeta Madre Gigabyte micro ATX Z490M GAMING X (rev. 1.0)	10
32	Tarjeta Madre ASRock Z390 Phantom Gaming 4	10
29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING	10
25	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming	10
23	Tarjeta de Video MSI Radeon X1550	10
20	Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC	10
16	Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming	10
87	Acer Audífonos Gamer Galea 300	8
47	SSD XPG SX8200 Pro	8
19	Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile	8
8	Procesador Intel Core i5-9600K	8
65	Samsung Smart TV LED UN70RU7100FXZX 70	7
62	Makena Smart TV LED 32S2 32"	6
61	Kit Memoria RAM Corsair Vengeance LPX DDR4	5
43	Tarjeta Madre ASUS ATX ROG STRIX Z390-E GAMING	5
18	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030	5
89	Cougar Audífonos Gamer Phontum Essential	4

73	Samsung Smart TV LED UN55TU7000FXZX 55	4
50	SSD Crucial MX500	4
71	Samsung Smart TV LED UN32J4290AF 32	3
56	SSD para Servidor Lenovo Thinksystem S4500	3
49	Kit SSD Kingston KC600	3
28	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti	3
96	Klip Xtreme Audífonos Blast	2
95	logear Audífonos Gamer GHG601	2
78	Ghia Bocina Portátil BX300	2
34	Tarjeta Madre ASUS ATX ROG STRIX B550-F GAMING WI-FI	2
24	Tarjeta de Video PNY NVIDIA GeForce RTX 2080	2
11	Tarjeta de Video ASUS AMD Radeon RX 570	2
90	Energy Sistem Audífonos con Micrófono Headphones 1	1
77	Verbatim Bocina Portátil Mini	1
74	Logitech Bocinas para Computadora con Subwoofer G560	1
53	SSD Addlink Technology S70	1
40	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare	1
17	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC	1
13	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix	1
51	SSD Kingston UV500	0
44	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX	0
42	Tarjeta Madre ASRock Micro ATX B450M Steel Legend	0
22	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC	0
21	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC	0
12	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC	0

Las últimas tablas son razones calculadas que permitirán decidir qué productos conservar en stock, retirar del negocio o mejorar su difusión para promover su venta.

Tabla 6: Productos ordenados de mayor a menor en razón de ventas/búsquedas.

ID	Nombre de producto	Ventas/Búsquedas
45	Tarjeta Madre ASRock ATX H110 Pro BTC+	1
10	MSI GeForce 210	1
42	Tarjeta Madre ASRock Micro ATX B450M Steel Legend	0.78
3	Procesador AMD Ryzen 5 2600	0.76
5	Procesador Intel Core i3-9100F	0.67
31	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0)	0.6
12	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC	0.6
11	Tarjeta de Video ASUS AMD Radeon RX 570	0.6
2	Procesador AMD Ryzen 5 3600	0.54

13	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix	0.5
18	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030	0.45
52	SSD Western Digital WD Blue 3D NAND	0.4
47	SSD XPG SX8200 Pro	0.37
74	Logitech Bocinas para Computadora con Subwoofer G560	0.33
48	SSD Kingston A2000 NVMe	0.33
17	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC	0.33
4	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8	0.32
49	Kit SSD Kingston KC600	0.3
6	Procesador Intel Core i9-9900K	0.3
51	SSD Kingston UV500	0.27
46	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2	0.25
44	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX	0.24
29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING	0.23
7	Procesador Intel Core i7-9700K	0.23
28	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti	0.2
25	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming	0.2
22	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC	0.2
8	Procesador Intel Core i5-9600K	0.2
1	Procesador AMD Ryzen 3 3300X S-AM4	0.2
54	SSD Kingston A400	0.19
94	HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro	0.17
89	Cougar Audífonos Gamer Phontum Essential	0.14
57	SSD Adata Ultimate SU800	0.14
50	SSD Crucial MX500	0.14
21	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC	0.13
84	Logitech Audífonos Gamer G332	0.1
40	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare	0.1
66	TCL Smart TV LED 55S425 54.6	0.07
85	Logitech Audífonos Gamer G635 7.1	0.06
67	TV Monitor LED 24TL520S-PU 24	0.03
95	logear Audífonos Gamer GHG601	0
93	Ginga Audífonos con Micrófono GI18ADJ01BT-RO	0
91	Genius GHP-400S Audífonos	0
80	Ghia Bocina Portátil BX800	0
76	Acteck Bocina con Subwoofer AXF-290	0
73	Samsung Smart TV LED UN55TU7000FXZX 55	0
70	Samsung Smart TV LED 43	0
63	Seiki TV LED SC-39HS950N 38.5	0
59	SSD Samsung 860 EVO	0
56	SSD para Servidor Lenovo Thinksystem S4500	0

39	ASUS T. Madre uATX M4A88T-M	0
35	Tarjeta Madre Gigabyte micro ATX Z390 M GAMING	0
27	Tarjeta de Video VisionTek AMD Radeon HD5450	0
26	Tarjeta de Video VisionTek AMD Radeon HD 5450	0
15	Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming	0
9	Procesador Intel Core i3-8100	0

Tabla 7: Productos ordenados de mayor a menor en razón de búsquedas/ventas.

ID	Nombre de producto	Búsquedas/Ventas
67	TV Monitor LED 24TL520S-PU 24	32
85	Logitech Audífonos Gamer G635 7.1	17.5
66	TCL Smart TV LED 55S425 54.6	15
84	Logitech Audífonos Gamer G332	10
40	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare	10
21	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC	7.5
57	SSD Adata Ultimate SU800	7.13
89	Cougar Audífonos Gamer Phontum Essential	7
50	SSD Crucial MX500	7
94	HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro	6
54	SSD Kingston A400	5.26
28	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti	5
25	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming	5
22	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC	5
8	Procesador Intel Core i5-9600K	5
1	Procesador AMD Ryzen 3 3300X S-AM4	5
7	Procesador Intel Core i7-9700K	4.43
29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING	4.29
44	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX	4.17
46	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2	4
51	SSD Kingston UV500	3.67
49	Kit SSD Kingston KC600	3.33
6	Procesador Intel Core i9-9900K	3.33
4	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8	3.15
74	Logitech Bocinas para Computadora con Subwoofer G560	3
48	SSD Kingston A2000 NVMe	3
17	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC	3
47	SSD XPG SX8200 Pro	2.73
52	SSD Western Digital WD Blue 3D NAND	2.5
18	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030	2.2
13	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix	2
2	Procesador AMD Ryzen 5 3600	1.85

31	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0)	1.67
12	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC	1.67
11	Tarjeta de Video ASUS AMD Radeon RX 570	1.67
5	Procesador Intel Core i3-9100F	1.5
3	Procesador AMD Ryzen 5 2600	1.31
42	Tarjeta Madre ASRock Micro ATX B450M Steel Legend	1.28
45	Tarjeta Madre ASRock ATX H110 Pro BTC+	1
10	MSI GeForce 210	1
60	Kit Memoria RAM Corsair Dominator Platinum DDR4	0
33	Tarjeta Madre ASUS ATX PRIME Z390-A	0

Se puede observar que para razones ventas/búsquedas mayores a 0.5 significa que es muy probable (más del 50%) que la búsqueda de ese producto concluya en una venta. Estos productos son:

1. Tarjeta Madre ASRock ATX H110 Pro BTC+
2. MSI GeForce 210
3. Tarjeta Madre ASRock Micro ATX B450M Steel Legend
4. Procesador AMD Ryzen 5 2600
5. Procesador Intel Core i3-9100F
6. Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0)
7. Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC
8. Tarjeta de Video ASUS AMD Radeon RX 570
9. Procesador AMD Ryzen 5 3600
10. Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix

Mientras que los siguientes no llegan ni al 20% de probabilidad de concluir en una venta:

- Procesador AMD Ryzen 3 3300X S-AM4
- SSD Kingston A400
- HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro
- Cougar Audífonos Gamer Phontum Essential
- SSD Adata Ultimate SU800
- SSD Adata Ultimate SU800
- SSD Crucial MX500
- Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC
- Logitech Audífonos Gamer G332
- Tarjeta Madre Gigabyte XL-ATX TRX40 Designare
- TCL Smart TV LED 55S425 54.6
- Logitech Audífonos Gamer G635 7.1
- TV Monitor LED 24TL520S-PU 24
- logear Audífonos Gamer GHG601
- Ginga Audífonos con Micrófono G118ADJ01BT-RO
- Genius GHP-400S Audífonos
- Ghia Bocina Portátil BX800
- Acteck Bocina con Subwoofer AXF-290

- Samsung Smart TV LED UN55TU7000FXZX 55
- Samsung Smart TV LED 43
- Seiki TV LED SC-39HS950N 38.5
- SSD Samsung 860 EVO
- SSD para Servidor Lenovo Thinksystem S4500
- ASUS T. Madre uATX M4A88T-M
- Tarjeta Madre Gigabyte micro ATX Z390 M GAMING
- Tarjeta de Video VisionTek AMD Radeon HD5450
- Tarjeta de Video VisionTek AMD Radeon HD 5450
- Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming
- Procesador Intel Core i3-8100

Un caso especial es el producto TV Monitor LED 24TL520S-PU 24, que a pesar de haber sólo una compra esta tiene un gran impacto en los ingresos netos.

Solución al problema

De acuerdo a lo discutido anteriormente, los datos de razón ventas/búsquedas, así como la cantidad de producto en stock y el monto de ingresos netos que aportan permitirá dar una lista en primer lugar de productos que se recomienda deben ser retirados del negocio porque no se venden y no son buscados, luego una lista recomendada de productos que requieren ser más difundidos para su venta, luego una lista de productos que deben ser más provistos de stock y finalmente una lista de productos que deben mantenerse en las condiciones con que se han manejado hasta ahora.

Productos a retirar porque no se venden, no son buscados y tienen poco stock, su impacto en los ingresos netos es mínimo:

- 19 - Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile
- 24 - Tarjeta de Video PNY NVIDIA GeForce RTX 2080
- 34 - Tarjeta Madre ASUS ATX ROG STRIX B550-F GAMING WI-FI
- 43 - Tarjeta Madre ASUS ATX ROG STRIX Z390-E GAMING
- 53 - SSD Addlink Technology S70
- 56 - SSD para Servidor Lenovo Thinksystem S4500
- 61 - Kit Memoria RAM Corsair Vengeance LPX DDR4
- 62 - Makena Smart TV LED 32S2 32"
- 65 - Samsung Smart TV LED UN70RU7100FXZX 70
- 71 - Samsung Smart TV LED UN32J4290AF 32
- 73 - Samsung Smart TV LED UN55TU7000FXZX 55
- 77 - Verbatim Bocina Portátil Mini
- 78 - Ghia Bocina Portátil BX300

Productos a considerar para mayor difusión de venta u otro mecanismo para consolidar compras porque se acumula demasiado en stock y no tiene ventas y búsquedas:

- 41 - Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM

- 68 - Makena Smart TV LED 40S2 40"
- 26 - Tarjeta de Video VisionTek AMD Radeon HD 5450
- 63 - Seiki TV LED SC-39HS950N 38.5
- 93 - Ginga Audífonos con Micrófono GI18ADJ01BT-RO
- 39 - ASUS T. Madre uATX M4A88T-M
- 69 - ASUS T. Madre uATX M4A88T-M
- 64 - Samsung TV LED LH43QMREBGCXGO 43
- 37 - Tarjeta Madre ASRock ATX Z490 STEEL LEGEND
- 30 - Tarjeta Madre AORUS ATX Z390 ELITE
- 27 - Tarjeta de Video VisionTek AMD Radeon HD5450
- 14 - Tarjeta de Video EVGA NVIDIA GeForce GT 710
- 9 - Procesador Intel Core i3-8100
- 82 - Ghia Bocina Portátil BX400
- 79 - Naceb Bocina Portátil NA-0301
- 35 - Tarjeta Madre Gigabyte micro ATX Z390 M GAMING
- 86 - ASUS Audífonos Gamer ROG Theta 7.1
- 81 - Ghia Bocina Portátil BX900
- 76 - Acteck Bocina con Subwoofer AXF-290
- 91 - Genius GHP-400S Audífonos
- 83 - Ghia Bocina Portátil BX500
- 58 - SSD para Servidor Lenovo Thinksystem S4510

Para elegir estos productos se consideró que estuvieran en stock más de 15 artículos de cada producto, pues el promedio de ventas mensuales aproximado ha sido de 31, de modo que no es recomendable tener ese producto dos meses más y esperar que la mitad de las ventas por mes sean de alguno de ellos, pues nadie los busco y mucho menos los compra.

No tienen ventas, pocas búsquedas, pero hay poco en stock, se puede retirar si no presenta problema:

- 87 - Acer Audífonos Gamer Galea 300
- 95 - logear Audífonos Gamer GHG601
- 96 - Klip Xtreme Audífonos Blast

Los productos a mantener hasta ahora, por su buen desempeño en el mercado son los siguientes:

- 3 - Procesador AMD Ryzen 5 2600
- 7 - Procesador Intel Core i7-9700K
- 2 - Procesador AMD Ryzen 5 3600
- 5 - Procesador Intel Core i3-9100F
- 6 - Procesador Intel Core i9-9900K
- 42 - Tarjeta Madre ASRock Micro ATX B450M Steel Legend
- 4 - Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8
- 54 - SSD Kingston A400
- 57 - SSD Adata Ultimate SU800
- 47 - SSD XPG SX8200 Pro

Conclusión

Se recomienda a la empresa LifeStore hacer un estudio para conocer mejor a sus clientes, pues existen productos en su mayoría bocinas, pantallas y varias tarjetas madres, así como ciertos modelos de audífonos de los cuales se tienen en demasiada cantidad y su distribución no ha sido efectiva, siendo las pantallas y tarjetas madres los productos en promedio de mayor precio que no se están vendiendo. Así como algunas tarjetas de video que son gran precio, pero no se venden. De modo que un mejor conocimiento de sus clientes le va a permitir una mejor estrategia para promover sus productos que no han sido buscados y mucho menos concluido en una venta. Otra estrategia sería analizar qué productos pueden abaratare sin el decremento de los ingresos promedio del negocio.

Finalmente, para los productos sin ventas ni búsquedas y poco stock, se pueden retirar de inmediato si su costo no es muy elevado o promoverlos en una sección o temporada especial para que lleguen precisamente al publico que desean, buscar activamente la atención e interés del cliente con estos productos.

Por otra parte, se ha evidenciado y demostrado con resultados cuantitativos y precisos la efectividad del análisis de datos aplicado a este negocio, que, bajo la correcta manipulación e interpretación de los datos, así como la asistencia de conocedores del negocio puede conducir a mejoras notables en su valor como empresa y eficiencia en sus operaciones.

Código del programa comentado

```

1429
1430 #En primer lugar se establecerán los usuarios permitidos con sus respectivas contraseñas, pero además se dará la opción de agregar nue
1431 #La primera entrada de cada elemento de la lista es el nombre de usuario, y al segunda entrada es la contraseña.
1432 users_regist = [['Administrador','adm'],['Equipo ventas','equipvent'],['Equipo inventario','equipinvent'],['Invitado','invit']]
1433 #El usuario Administrador tiene opciones extra que pueden ser consultadas en el reporte.
1434
1435 print('*'*110)
1436 print('{:^110s}'.format('BIENVENIDO AL SISTEMA DE ANÁLISIS DE LIFESTORE'))
1437 print('*'*110)
1438
1439 ans1 = False
1440
1441 while ans1 == False:
1442     status1 = input('¿Es un usuario registrado en el sistema? (si/no): ')
1443     if status1 == 'si':
1444         user_found = False
1445         user = input('Ingrese su nombre de usuario: ')
1446         passwd = input('Ingrese su contraseña: ')
1447         for u in range(0,len(users_regist)):
1448             if user == users_regist[u][0] and passwd == users_regist[u][1]:
1449                 print('*'*30)
1450                 print('\n Bienvenido ' + users_regist[u][0])
1451                 print('*'*30)
1452                 user_found = True
1453                 ans1 = True
1454                 break
1455             if user_found == False:
1456                 print('\n Nombre de usuario o contraseña incorrectos. Verifique si tiene un nombre de usuario y contraseña registrados.')
1457                 ans1 = False
1458         elif status1 == 'no':
1459             status2 = input('¿Desea crear un nuevo usuario y contraseña? (si/no): ')
1460             if status2 == 'si':
1461                 new_user = input('Nombre de usuario: ')
1462                 new_passwd = input('Contraseña: ')
1463                 users_regist.append([new_user,new_passwd])
1464                 print('*'*30)
1465                 print('\n Bienvenido ' + new_user)
1466                 print('*'*30)
1467                 ans1 = True
1468             elif status2 == 'no':
1469                 print('\n Verifique si cuenta con un nombre de usuario y contraseña permitidos.')
1470                 ans1 = False
1471             else:
1472                 print('\n No ingresó alguna de las respuestas permitidas. Intente de nuevo.')
1473                 ans1 = False
1474
1475         else:
1476             print('\n No ingresó alguna de las respuestas permitidas. Intente de nuevo.')
1477             ans1 = False
1478
1479 #Por practicidad de empleo de variables se definirán ciertos valores que serán usados regularmente:
1480
1481 length_sales_list = len(lifestore_sales)
1482 length_product_list = len(lifestore_products)
1483 length_searches_list = len(lifestore_searches)
1484
1485

```

```

1485
1486
1487 #ALGORITMOS DE BÚSQUEDA Y RECOLECCIÓN DE DATOS.
1488
1489 months = []
1490 months_names=['enero','febrero','marzo','abril','mayo','junio','julio','agosto','septiembre','octubre','noviembre','diciembre']
1491 for j in range(1,12):
1492     for i in range(0,length_sales_list):
1493         if int(lifstore_sales[i][3][4]) == j:
1494             months.append([j,months_names[j-1]])
1495             break
1496
1497 length_months_list = len(months)
1498
1499 print('\n Existe registro de ventas de los siguientes meses:')
1500 for m in range(0,length_months_list):
1501     print(str(months[m][0]) + '-' + months[m][1])
1502
1503
1504
1505 #Ahora se crearán algunas listas útiles para extraer la información necesaria en cada caso.
1506
1507 #Antes se explicará que la función split() sobre listas permite separar una cadena de caracteres cada vez que el caracter
1508 #elegido como argumento de la función aparece en la cadena.
1509 #Así, si el caracter aparece, pro ejemplo, dos veces en la cadena, ésta se separará en otras tres cadenas dentro de una
1510 #lista y si se desea recuperar uno de esos fragemntos basta con seleccionar
1511 #el elemento de la cadena resultante.
1512 #Ejemplo: si lista = 'Hola, me llamo Hugo, soy amigable' y usar la función split para separar cada vez que aparece el
1513 #caracter ',', se obtiene lo siguiente:
1514 #lista.split(',') = ['Hola','me llamo Hugo','soy amigable'] y si sólo se desea un fragmento de la cadena, por ejemplo
1515 #'Hola' entonces se puede escribir lista.split(',')[0] = 'Hola'
1516
1517 #Esta primera lista contiene casi toda la información necesaria por producto (id del producto, nombre reducido, categoria y ventas por mes).
1518 product_month_sales = []
1519
1520 for n in range(1,length_product_list +1):
1521     product_month_sales.append([n,lifstore_products[n-1][1].split(',')[0],lifstore_products[n-1][3]])
1522     for m in range(1,length_months_list +1):
1523         sales = 0
1524         refunds = 0
1525         for i in range(0,length_sales_list):
1526             if n == lifstore_sales[i][1] and m == int(lifstore_sales[i][3][4]):
1527                 sales += 1
1528                 if 1 == lifstore_sales[i][4]:
1529                     refunds += 1
1530         product_month_sales[n-1].append([m,sales,refunds])
1531
1532 #La lista tiene la siguiente forma:
1533 #product_month_sales = [[id_product, reduced_name, category, [month_number, sales, refunds]]]
1534
1535 length_product_month_sales = len(product_month_sales)
1536
1537 #print(product_month_sales)
1538
1539
1540
1541 #Ahora se crearán algunas listas útiles para extraer la información necesaria en cada caso.
1542
1543 #Antes se explicará que la función split() sobre listas permite separar una cadena de caracteres cada vez que el caracter
1544 #elegido como argumento de la función aparece en la cadena.
1545 #Así, si el caracter aparece, pro ejemplo, dos veces en la cadena, ésta se separará en otras tres cadenas dentro de una
1546 #lista y si se desea recuperar uno de esos fragemntos basta con seleccionar
1547 #el elemento de la cadena resultante.
1548 #Ejemplo: si lista = 'Hola, me llamo Hugo, soy amigable' y usar la función split para separar cada vez que aparece el
1549 #caracter ',', se obtiene lo siguiente:
1550 #lista.split(',') = ['Hola','me llamo Hugo','soy amigable'] y si sólo se desea un fragmento de la cadena, por ejemplo
1551 #'Hola' entonces se puede escribir lista.split(',')[0] = 'Hola'
1552
1553 #Esta primera lista contiene casi toda la información necesaria por producto (id del producto, nombre reducido, categoria y ventas por mes).
1554 product_month_sales = []
1555
1556 for n in range(1,length_product_list +1):
1557     product_month_sales.append([n,lifstore_products[n-1][1].split(',')[0],lifstore_products[n-1][3]])
1558     for m in range(1,length_months_list +1):
1559         sales = 0
1560         refunds = 0
1561         for i in range(0,length_sales_list):
1562             if n == lifstore_sales[i][1] and m == int(lifstore_sales[i][3][4]):
1563                 sales += 1
1564                 if 1 == lifstore_sales[i][4]:
1565                     refunds += 1
1566         product_month_sales[n-1].append([m,sales,refunds])
1567
1568 #La lista tiene la siguiente forma:
1569 #product_month_sales = [[id_product, reduced_name, category, [month_number, sales, refunds]]]
1570
1571 length_product_month_sales = len(product_month_sales)
1572
1573 #print(product_month_sales)
1574
1575
1576
1577 #Ahora, una lista que tendrá el número de ventas por producto (de todo el tiempo que se tiene registros, en este caso del año 2020).
1578 #Se puede construir directamente de las listas de la base de datos:
1579 product_sales = []
1580
1581 for n in range(1,length_product_list +1):
1582     sales = 0
1583     for i in range(0,length_sales_list):
1584         if n == lifstore_sales[i][1]:
1585             sales += 1
1586     product_sales.append([n,lifstore_products[n-1][1].split(',')[0],lifstore_products[n-1][3],sales])
1587
1588 #O bien construirse a partir de la lista creada al inicio (product_month_sales) con el siguiente algoritmo:

```

```

1554 '''product_sales = []
1555
1556 for i in range(0, length_product_month_sales):
1557     sales = 0
1558     for k in range(0, len(product_month_sales[i][3:])):
1559         sales += product_month_sales[i][k+3][1]
1560     ... product_sales.append([i+1, lifestore_products[n-1][1].split(',')[0], lifestore_products[n-1][3], sales])
1561
1562
1563 #En cualquier caso, la lista tiene la siguiente forma:
1564 #product_sales = [[id_product, reduced_name, category, sales]
1565
1566 #print(product_sales)
1567
1568
1569
1570 #La siguiente lista tendrá los datos de la cantidad de búsquedas por producto.
1571 product_searches = []
1572
1573 for n in range(1, length_product_list + 1):
1574     searches = 0
1575     for i in range(0, length_searches_list):
1576         if n == lifestore_searches[i][1]:
1577             searches += 1
1578     product_searches.append([n, lifestore_products[n-1][1].split(',')[0], lifestore_products[n-1][3], searches])
1579
1580 #La lista tiene la siguiente forma:
1581 #product_searches = [[id_product, reduced_name, category, searches]
1582
1583 #print(product_searches)
1584
1585
1586
1587 #Después, se creará una lista con número de cada calificación de reseña por producto. Los productos con devolución son igualmente considerados.
1588 product_scores = []
1589
1590 for n in range(1, length_product_list + 1):
1591     scores_1 = 0
1592     scores_2 = 0
1593     scores_3 = 0
1594     scores_4 = 0
1595     scores_5 = 0
1596     for i in range(0, length_sales_list):
1597         if n == lifestore_sales[i][1] and 1 == lifestore_sales[i][2]:
1598             scores_1 += 1
1599         elif n == lifestore_sales[i][1] and 2 == lifestore_sales[i][2]:
1600             scores_2 += 1
1601         elif n == lifestore_sales[i][1] and 3 == lifestore_sales[i][2]:
1602             scores_3 += 1
1603         elif n == lifestore_sales[i][1] and 4 == lifestore_sales[i][2]:
1604             scores_4 += 1
1605         elif n == lifestore_sales[i][1] and 5 == lifestore_sales[i][2]:
1606             scores_5 += 1
1607     product_scores.append([n, lifestore_products[n-1][1].split(',')[0], [5, scores_5], [4, scores_4], [3, scores_3], [2, scores_2], [1, scores_1]])
1608
1609 #La lista tiene la siguiente forma:
1610 #product_scores = [[id_product, reduced_name, category, [score, number_scores]]]
1611

```

```

1615
1616 #Luego, una lista de la cantidad de ventas por mes creada directamente de las listas de la base de datos.
1617 month_sales = []
1618
1619 for k in range(1,length_months_list +1):
1620     #for n in range(1,len(lifystore_products)+1):
1621     sales = 0
1622     for i in range(0,length_sales_list):
1623         if k == int(lifystore_sales[i][3][4]):
1624             sales += 1
1625     month_sales.append([k,months[k-1][1],sales])
1626
1627 #print(month_sales)
1628 #len(product_month_sales[0][3:]) esto se puede reemplazar por len(length_months_list) pues siempre tienen la misma cantidad de elementos
1629 # y se refiere precisamente al número de meses ambos.
1630
1631 #También se puede crear a partir de la primera lista creada (product_month_sales) con el siguiente algoritmo:
1632 '''month_sales = []
1633
1634 for m in range(1,length_months_list +1):
1635     sales = 0
1636     for i in range(0,length_product_month_sales):
1637         for k in range(0,len(product_month_sales[i][3:])):
1638             if m == product_month_sales[i][k+3][0]:
1639                 sales += product_month_sales[i][k+3][1]
1640     month_sales.append([m,months[m-1][1],sales])
1641
1642 print(month_sales)'''
1643
1644 #En cualquier caso, la lista tiene la siguiente forma:
1645 #months_sales = [[month_number, month_name, sales]
1646
1647
1648 #También será de utilidad la lista con las categorías existentes y un respectivo código de identificación.
1649 #Notar que primero se crea un programa que identifique automáticamente todos los nombres de las categorías existentes
1650
1651 categ_names = []
1652 for i in range(0,len(lifystore_products)):
1653     if lifystore_products[i][3] not in categ_names:
1654         categ_names.append(lifystore_products[i][3])
1655
1656 #print(categ_names)
1657
1658 categories = []
1659
1660 for j in range(0,len(categ_names)):
1661     categories.append([j+1,categ_names[j]])
1662
1663 #Se creará luego una lista que agrupa por categoría a los productos y su correspondiente número de ventas.
1664
1665 categ_prod_sales = []
1666
1667 for c in range(0,len(categories)):
1668     categ_prod_sales.append([categories[c][0],categories[c][1]])
1669     for i in range(0,len(product_sales)):
1670         if categories[c][1] == product_sales[i][2]:
1671             categ_prod_sales[c].append([product_sales[i][0],product_sales[i][1],product_sales[i][3]])
1672

```

```

1672
1673 #La lista es de la forma:
1674 #categ_prod_sales = [[categ_number, categ_name, [id_product, reduced_name, sales]]]
1675
1676 #Se creará luego una lista que agrupa por categoria a los productos y su correspondiente número de búsquedas.
1677
1678 categ_prod_searches = []
1679
1680 for c in range(0,len(categories)):
1681     categ_prod_searches.append([categories[c][0],categories[c][1]])
1682     for i in range(0,len(product_searches)):
1683         if categories[c][1] == product_searches[i][2]:
1684             categ_prod_searches[c].append([product_searches[i][0],product_searches[i][1],product_searches[i][3]])
1685
1686 #La lista es de la forma:
1687 #categ_prod_searches = [[categ_number, categ_name, [id_product, reduced_name, searches]]]
1688
1689
1690
1691 #ALGORITMOS DE ORDENAMIENTO, CLASIFICACIÓN Y DEPURACIÓN DE DATOS.
1692
1693 #PARA OPCIÓN EXTRA 4 (Se pone aquí para evitar que el orden de las listan siguientes afecte el resultado)
1694 #Forma parte de las opciones extra del Administrador.
1695
1696 #Se creará una lista con datos de razón ventas/búsquedas por producto.
1697 product_rate_sale_search=[]
1698
1699 for n in range(0,len(lifystore_products)):
1700     if product_searches[n][3] != 0:
1701         product_rate_sale_search.append([n+1,product_searches[n][1],round(product_sales[n][3]/product_searches[n][3],2)])
1702
1703 #La lista es de la forma: product_rate_sale_search = [[id_product,reduced_name,rate]]
1704
1705
1706
1707 #Se va a ordenar la lista
1708 for i in range(0,len(product_rate_sale_search)):
1709     for j in range(i+1,len(product_rate_sale_search)):
1710         if product_rate_sale_search[i][2] <= product_rate_sale_search[j][2]:
1711             temp = product_rate_sale_search[i]
1712             product_rate_sale_search[i] = product_rate_sale_search[j]
1713             product_rate_sale_search[j] = temp
1714
1715
1716
1717 product_rate_search_sale=[]
1718
1719 for n in range(0,len(lifystore_products)):
1720     if product_sales[n][3] != 0:
1721         product_rate_search_sale.append([n+1,product_searches[n][1],round(product_searches[n][3]/product_sales[n][3],2)])
1722
1723 #La lista es de la forma: product_rate_sale_search = [[id_product,reduced_name,rate]]
1724
1725 #Se creará una lista con datos de razón búsquedas/ventas por producto.
1726

```



```

1724
1725 #Se creará una lista con datos de razón búsquedas/ventas por producto.
1726
1727 #Se va a ordenar la lista
1728
1729 for i in range(0,len(product_rate_search_sale)):
1730     for j in range(i+1,len(product_rate_search_sale)):
1731         if product_rate_search_sale[i][2] <= product_rate_search_sale[j][2]:
1732             temp = product_rate_search_sale[i]
1733             product_rate_search_sale[i] = product_rate_search_sale[j]
1734             product_rate_search_sale[j] = temp
1735
1736
1737 #PARA TAREA 1
1738
1739 #El siguiente algoritmo es para ordenar la lista product_sales de mayor a menor número de ventas.
1740 for i in range(0,len(product_sales)):
1741     for j in range(i+1,len(product_sales)):
1742         if product_sales[i][3] <= product_sales[j][3]:
1743             temp = product_sales[i]
1744             product_sales[i] = product_sales[j]
1745             product_sales[j] = temp
1746             #product_sales[i],product_sales[j] = product_sales[j],product_sales[i]
1747
1748 #El siguiente algoritmo es para tener una lista con los productos ordenados de mayor a menor número de ventas.
1749 #Además se omiten los productos que no tienen ventas.
1750 prod_greatsales = []
1751 for i in range(0,len(product_sales)):
1752     if product_sales[i][3] != 0:
1753         prod_greatsales.append(product_sales[i])
1754
1755 #El siguiente algoritmo es para tener una lista con los productos ordenados de menor a mayor número de ventas.
1756 prod_lesssales = []
1757 for i in range(-1,-(len(product_sales)+1),-1):
1758     if product_sales[i][3] == 0:
1759         prod_lesssales.append(product_sales[i])
1760
1761 #print(prod_lesssales)
1762
1763
1764 #PARA TAREA 2
1765
1766 #El siguiente algoritmo es para ordenar la lista product_searches de mayor a menor número de búsquedas.
1767 for i in range(0,len(product_searches)):
1768     for j in range(i+1,len(product_searches)):
1769         if product_searches[i][3] <= product_searches[j][3]:
1770             temp = product_searches[i]
1771             product_searches[i] = product_searches[j]
1772             product_searches[j] = temp
1773             #product_searches[i],product_searches[j] = product_searches[j],product_searches[i]
1774
1775 #El siguiente algoritmo conserva en una lista sólo los productos con ventas diferente de cero.
1776 #Además se omiten los productos que no tienen búsquedas.
1777 prod_greatsearches = []
1778 for i in range(0,len(product_searches)):
1779     if product_searches[i][3] != 0:
1780         prod_greatsearches.append(product_searches[i])
1781

```

```

1781
1782 #El siguiente algoritmo es para tener una lista con los productos ordenados de menor a mayor número de ventas.
1783 prod_lessearches = []
1784 for i in range(-1,-(len(product_searches)+1),-1):
1785     if product_searches[i][3] == 0:
1786         prod_lessearches.append(product_sales[i])
1787
1788 #print(prod_lessearches)
1789
1790
1791 #PARA TAREA 3
1792
1793 #Se va ordenar la lista categ_prod_sales de menor a mayor número de ventas.
1794 for n in range(0,len(categ_prod_sales)):
1795     for i in range(0,len(categ_prod_sales[n][2:])):
1796         for j in range(i+1,len(categ_prod_sales[n][2:])):
1797             if categ_prod_sales[n][i+2][2] >= categ_prod_sales[n][j+2][2]:
1798                 temp = categ_prod_sales[n][i+2]
1799                 categ_prod_sales[n][i+2] = categ_prod_sales[n][j+2]
1800                 categ_prod_sales[n][j+2] = temp
1801                 #categ_prod_sales[n][i+2],categ_prod_sales[n][j+2] = categ_prod_sales[n][j+2],categ_prod_sales[n][i+2]
1802
1803
1804 #PARA TAREA 4
1805
1806 #Se va ordenar la lista categ_prod_searches de menor a mayor número de búsquedas.
1807 for n in range(0,len(categ_prod_searches)):
1808     for i in range(0,len(categ_prod_searches[n][2:])):
1809         for j in range(i+1,len(categ_prod_searches[n][2:])):
1810             if categ_prod_searches[n][i+2][2] >= categ_prod_searches[n][j+2][2]:
1811                 temp = categ_prod_searches[n][i+2]
1812                 categ_prod_searches[n][i+2] = categ_prod_searches[n][j+2]
1813                 categ_prod_searches[n][j+2] = temp
1814                 #categ_prod_searches[n][i+2],categ_prod_searches[n][j+2] = categ_prod_searches[n][j+2],categ_prod_searches[n][i+2]
1815
1816
1817 #PARA TAREA 5
1818
1819 #Antes de comenzar se hará una copia de la lista product_scores que será ocupada en la opción 1 de la tarea 6,
1820 #esto porque la original será modificada en la opción 1 de la tarea 5.
1821 product_scores2 = []
1822 for i in range(0,len(product_scores)):
1823     product_scores2.append(product_scores[i])
1824
1825 #Algoritmo para la opción 1
1826
1827 #Este algoritmo ordena los productos en la lista de acuerdo a mayor cantidad de calificaciones de valor 5 (el máximo).
1828 for i in range(0,len(product_scores)):
1829     for j in range(i+1,len(product_scores)):
1830         if product_scores[i][2][1] <= product_scores[j][2][1]:
1831             temp = product_scores[i]
1832             product_scores[i] = product_scores[j]
1833             product_scores[j] = temp
1834             #product_scores[i],product_scores[j] = product_scores[j],product_scores[i]
1835
1836

```

```

1835 |
1836 #El siguiente algoritmo es para conservar sólo los productos con calificaciones, es decir, se quitan los que no tienen reseñas.
1837 prod_bestreviews = []
1838 for i in range(0,len(product_scores)):
1839     if product_scores[i][2][1]!=0 or product_scores[i][3][1]!=0 or product_scores[i][4][1]!=0 or product_scores[i][5][1]!=0 or product_scores[i][6][1]!=0:
1840         prod_bestreviews.append(product_scores[i])
1841
1842 #Algoritmo para la opción 2
1843
1844 #Este algoritmo calcula la calificación promedio para cada producto, evidentemente sólo de los que tienen al menos una calificación; es decir al menos una venta.
1845 product_avgscore = []
1846 for i in range(0,len(product_scores)):
1847     sales = product_scores[i][2][1]+product_scores[i][3][1]+product_scores[i][4][1]+product_scores[i][5][1]+product_scores[i][6][1]
1848     if sales != 0:
1849         avg_score=(5*product_scores[i][2][1]+4*product_scores[i][3][1]+3*product_scores[i][4][1]+2*product_scores[i][5][1]+1*product_scores[i][6][1])/sales
1850         avg_score=round(avg_score,1)
1851         product_avgscore.append([product_scores[i][0],product_scores[i][1],sales,avg_score])
1852
1853 #print(product_avgscore)
1854 #Luego se ordenan de mayor a menor calificación.
1855 for i in range(0,len(product_avgscore)):
1856     for j in range(i+1,len(product_avgscore)):
1857         if product_avgscore[i][3] <= product_avgscore[j][3]:
1858             temp = product_avgscore[i]
1859             product_avgscore[i] = product_avgscore[j]
1860             product_avgscore[j] = temp
1861             #product_avgscore[i],product_avgscore[j] = product_avgscore[j],product_avgscore[i]
1862
1863
1864 #PARA TAREA 6
1865
1866 #Algoritmo para la opción 1
1867
1868 #Este algoritmo ordena los productos en la lista de acuerdo a mayor cantidad de calificaciones de valor 1 (el mínimo).
1869 for i in range(0,len(product_scores2)):
1870     for j in range(i+1,len(product_scores2)):
1871         if product_scores2[i][6][1] <= product_scores2[j][6][1]:
1872             temp = product_scores2[i]
1873             product_scores2[i] = product_scores2[j]
1874             product_scores2[j] = temp
1875             #product_scores2[i],product_scores2[j] = product_scores2[j],product_scores2[i]
1876
1877 #El siguiente algoritmo es para conservar sólo los productos con calificaciones, es decir, se quitan los que no tienen reseñas.
1878 prod_worstreviews = []
1879 for i in range(0,len(product_scores2)):
1880     if product_scores2[i][2][1]!=0 or product_scores2[i][3][1]!=0 or product_scores2[i][4][1]!=0 or product_scores2[i][5][1]!=0 or product_scores2[i][6][1]!=0:
1881         prod_worstreviews.append(product_scores2[i])
1882
1883 #Algoritmo para la opción 2 es igual al de la tarea 5 pero se imprime al revés.
1884
1885
1886 #PARA TAREA 7
1887
1888 #El siguiente algoritmo calcula la cantidad de ingresos neto (ventas menos devoluciones) por mes y por producto.
1889 product_month_incomes = []
1890 for i in range(0,len(product_month_sales)):
1891     product_month_incomes.append([product_month_sales[i][0],product_month_sales[i][1],product_month_sales[i][2]])
1892     for m in range(0,len(months)):
1893         product_month_incomes[i].append([m+1,(product_month_sales[i][m+3][1] - product_month_sales[i][m+3][2])*lifestore_products[i][2]])

```

```

1894
1895 #La lista tiene la siguiente forma:
1896 #product_month_incomes = [[id_product, reduced_name, category, [month_number, incomes]]]
1897 #print(product_month_incomes)
1898
1899 #Ahora, se construirá una lista que sume los ingresos netos de todos los productos por mes, esto resultará en los ingresos netos por mes.
1900 month_incomes = []
1901 for m in range(1, length_months_list + 1):
1902     incomes = 0
1903     for i in range(0, len(product_month_incomes)):
1904         for k in range(0, len(product_month_incomes[i][3:])):
1905             if m == product_month_incomes[i][k+3][0]:
1906                 incomes += product_month_incomes[i][k+3][1]
1907     month_incomes.append([m, months[m-1][1], incomes])
1908
1909 #La lista tiene la siguiente forma:
1910 #month_incomes = [[month_number, month_name, incomes]]
1911 #print(month_incomes)
1912
1913
1914
1915 #PARA TAREA 8 y TAREA 9
1916
1917 #Instrucción para calcular la cantidad total de ventas durante todo el registro de ventas.
1918 annual_sales = 0
1919 for i in range(0, len(month_sales)):
1920     annual_sales += month_sales[i][2]
1921
1922 avg_salesmonth = annual_sales/len(months)
1923
1924 annual_incomes = 0
1925 for i in range(0, len(month_incomes)):
1926     annual_incomes += month_incomes[i][2]
1927
1928 avg_incomesmonth = annual_incomes/len(months)
1929
1930
1931 #PARA TAREA 10
1932
1933 #La lista de ventas por mes ya está creada (month_sales), sólo resta ordenarla de mayor a menor cantidad de ventas.
1934 for i in range(0, len(month_sales)):
1935     for j in range(i+1, len(month_sales)):
1936         if month_sales[i][2] <= month_sales[j][2]:
1937             temp = month_sales[i]
1938             month_sales[i] = month_sales[j]
1939             month_sales[j] = temp
1940             #month_sales[i], month_sales[j] = month_sales[j], month_sales[i]
1941
1942
1943 #PARA OPCIÓN EXTRA 1
1944
1945 #Primero se creara una lista de ingresos netos por producto.
1946 product_incomes = []
1947
1948 for i in range(0, len(product_month_incomes)):
1949     incomes = 0
1950     for k in range(0, len(product_month_incomes[i][3:])):
1951         incomes += product_month_incomes[i][k+3][1]
1952     product_incomes.append([product_month_incomes[i][0], product_month_incomes[i][1], product_month_incomes[i][2], incomes])

```

```

1953
1954 #La lista resulta de la forma:
1955 #product_incomes = [[id_product, reduced_name, category, incomes]]
1956
1957 #Se creará una lista que agrupa por categoria a los productos y su correspondiente número de ingresos netos.
1958
1959 categ_prod_incomes = []
1960 for c in range(0,len(categories)):
1961     categ_prod_incomes.append([categories[c][0],categories[c][1]])
1962     for i in range(0,len(product_incomes)):
1963         if categories[c][1] == product_incomes[i][2]:
1964             categ_prod_incomes[c].append([product_incomes[i][0],product_incomes[i][1],product_incomes[i][3]])
1965
1966 #La lista es de la forma:
1967 #categ_prod_incomes = [[categ_number, categ_name, [id_product, reduced_name, incomes]]]
1968
1969 #Se va ordenar de acuerdo a menores ingresos netos la lista categ_prod_incomes.
1970 for n in range(0,len(categ_prod_incomes)):
1971     for i in range(0,len(categ_prod_incomes[n][2:])):
1972         for j in range(i+1,len(categ_prod_incomes[n][2:])):
1973             if categ_prod_incomes[n][i+2][2] >= categ_prod_incomes[n][j+2][2]:
1974                 temp = categ_prod_incomes[n][i+2]
1975                 categ_prod_incomes[n][i+2] = categ_prod_incomes[n][j+2]
1976                 categ_prod_incomes[n][j+2] = temp
1977                 #categ_prod_incomes[n][i+2],categ_prod_incomes[n][j+2] = categ_prod_incomes[n][j+2],categ_prod_incomes[n][i+2]
1978
1979
1980 #PARA OPCIÓN EXTRA 2
1981
1982 #El siguiente algoritmo es para ejecutar permutaciones entre elementos sucesivos bajo el criterio de orden descendente.
1983 for i in range(0,len(product_incomes)):
1984     for j in range(i+1,len(product_incomes)):
1985         if product_incomes[i][3] <= product_incomes[j][3]:
1986             temp = product_incomes[i]
1987             product_incomes[i] = product_incomes[j]
1988             product_incomes[j] = temp
1989             #product_incomes[i],product_incomes[j] = product_incomes[j],product_incomes[i]
1990
1991 #El siguiente algoritmo es para tener una lista con los productos ordenados de mayor a menor número de ingresos netos.
1992 #Además se omiten los productos que no tienen ingresos netos.
1993 prod_greatincomes = []
1994 for i in range(0,len(product_incomes)):
1995     if product_incomes[i][3] != 0:
1996         prod_greatincomes.append(product_incomes[i])
1997
1998 #El siguiente algoritmo es para tener una lista con los productos ordenados de menor a mayor monto de ingresos netos.
1999 prod_lessincomes = []
2000 for i in range(-1,-(len(product_incomes)+1),-1):
2001     if product_incomes[i][3] == 0:
2002         prod_lessincomes.append(product_incomes[i])
2003
2004
2005 #PARA OPCIÓN EXTRA 3
2006
2007 #Se creará una lista que contenga el valor de stock por producto.
2008 product_stock = []
2009 for i in range(0,len(lifestore_products)):
2010     product_stock.append([i+1,lifestore_products[i][1].split(',')[0],lifestore_products[i][3],lifestore_products[i][4]])
2011

```

```

2011
2012 #La lista resulta de la siguiente forma:
2013 #product_stock = [[id_procut, reduced_name, category, stock]]
2014
2015 #La siguiente instrucción es para ordenar de mayor a menor valor de stock.
2016 for i in range(0,len(product_stock)):
2017     for j in range(i+1,len(product_stock)):
2018         if product_stock[i][3] <= product_stock[j][3]:
2019             temp = product_stock[i]
2020             product_stock[i] = product_stock[j]
2021             product_stock[j] = temp
2022             #product_stock[i],product_stock[j] = product_stock[j],product_stock[i]
2023
2024
2025
2026
2027
2028 #PRESENTACIÓN DE RESULTADOS
2029 #Aquí inicia la presentación de resultados
2030
2031 ans2 = False
2032
2033 while ans2 == False:
2034
2035     print('\n Las siguientes opciones estan disponibles para este usuario:')
2036     print('***100)
2037     print('')
2038     -Productos más vendidos y rezagados:-
2039     1 Productos con mayores ventas.
2040     2 Productos con mayores búsquedas.
2041     3 Productos con menores ventas por categoría.
2042     4 Productos con menores búsquedas por categoría.
2043
2044     -Productos por reseña en el servicio:-
2045     5 Productos con mejores reseñas.
2046     6 Productos con peores reseñas.
2047
2048     -Relación sobre ingresos y ventas:-
2049     7 Total de ingresos mensuales.
2050     8 Ventas promedio mensuales.
2051     9 Total de ingresos anual.
2052     10 Meses con más ventas al año.'')
2053     print('***100)
2054
2055     option_chosen = int(input('Ingrese el número de opción que desea visualizar: '))
2056
2057     if option_chosen == 1:
2058         print('\n A continuación los productos ordenados de mayor a menor cantidad de ventas:\n')
2059         #La siguiente instrucción es para presentar los resultados.
2060         print('-'*110)
2061         print('ID                               Nombre                               No.ventas')
2062         print('-'*110)
2063         for m in range(0,len(prod_greatsales)):
2064             print('{:2d} {:<68s} {:4d}'.format(prod_greatsales[m][0],prod_greatsales[m][1],prod_greatsales[m][3]))
2065

```



```

2065     admflag1 = False
2066     while user == "Administrador" and admflag1 == False:
2067         opt1 = input('¿Desea ver la lista con los productos con menores ventas? (si/no): ')
2068         if opt1 == 'si':
2069             print('\n A continuación Los productos que no tuvieron ventas:\n')
2070             #La siguiente instrucción es para presentar los resultados.
2071             print('-'*110)
2072             print('ID                Nombre                No.ventas')
2073             print('-'*110)
2074             for m in range(0,len(prod_lessales)):
2075                 print('{:2d} {:<68s} {:4d}'.format(prod_lessales[m][0],prod_lessales[m][1],prod_lessales[m][3]))
2076             admflag1 = True
2077         elif opt1 == 'no':
2078             admflag1 = True
2079         else:
2080             print('No ingresó alguna de las respuestas permitidas. Intente de nuevo.')
2081
2082     elif option_chosen == 2:
2083         print('\n A continuación Los productos ordenados de mayor a menor cantidad de búsquedas:\n')
2084         #La siguiente instrucción es para presentar los resultados.
2085         print('-'*110)
2086         print('ID                Nombre                No.búsquedas')
2087         print('-'*110)
2088         for m in range(0,len(prod_greatsearches)):
2089             print('{:2d} {:<68s} {:4d}'.format(prod_greatsearches[m][0],prod_greatsearches[m][1],prod_greatsearches[m][3]))
2090
2091     admflag2 = False
2092     while user == "Administrador" and admflag2 == False:
2093         opt2 = input('¿Desea ver la lista con los productos con menores búsquedas? (si/no): ')
2094         if opt2 == 'si':
2095             print('\n A continuación Los productos que no tuvieron búsquedas:\n')
2096             #La siguiente instrucción es para presentar los resultados.
2097             print('-'*110)
2098             print('ID                Nombre                No.búsquedas')
2099             print('-'*110)
2100             for m in range(0,len(prod_lessearches)):
2101                 print('{:2d} {:<68s} {:4d}'.format(prod_lessearches[m][0],prod_lessearches[m][1],prod_lessearches[m][3]))
2102             admflag2 = True
2103         elif opt2 == 'no':
2104             admflag2 = True
2105         else:
2106             print('No ingresó alguna de las respuestas permitidas. Intente de nuevo.')
2107
2108     elif option_chosen == 3:
2109         print('\n A continuación Los productos ordenados de menor a mayor cantidad de ventas por categoria:\n')
2110         #Ahora se va a imprimir con el formato adecuado a todas las categorías y sus productos con menores ventas.
2111         for c in range(0,len(categ_prod_sales)):
2112             print('')
2113             print('+*50')
2114             print('Categoría','[' ,categ_prod_sales[c][0],']',categ_prod_sales[c][1])
2115             print('-'*110)
2116             print('ID                Nombre                No.ventas')
2117             print('-'*110)
2118             for i in range(0,len(categ_prod_sales[c][2:])):
2119                 print('{:2d} {:<68s} {:4d}'.format(categ_prod_sales[c][i+2][0],categ_prod_sales[c][i+2][1],categ_prod_sales[c][i+2][2]))
2120
2121

```

```

2121
2122
2123     print('\n A continuación los productos ordenados de menor a mayor cantidad de ventas por categoría:\n')
2124     #Ahora se va a imprimir con el formato adecuado a todas las categorías y sus productos con menores búsquedas
2125     for c in range(0,len(categ_prod_searches)):
2126         print('')
2127         print('*50')
2128         print('Categoría', '[' ,categ_prod_searches[c][0], ']', categ_prod_searches[c][1])
2129         print('*110')
2130         print('ID'                                     Nombre                                     No. búsquedas')
2131         print('*110')
2132         for i in range(0,len(categ_prod_searches[c][2])):
2133             print('{:2d} {:<68s} {:4d}'.format(categ_prod_searches[c][i+2][0],categ_prod_searches[c][i+2][1],categ_prod_searches[c][i+2][2]))
2134
2135     elif option_chosen == 5:
2136         print('\n Opciones para visualización de productos con mejores reseñas:
2137
2138     1 Con mayor cantidad de calificaciones de valor 5 (el máximo).
2139     2 Mayor calificación promedio.'')
2140
2141     opt5 = int(input('Ingrese el número de la opción deseada: '))
2142
2143     if opt5 == 1:
2144         #La siguiente instrucción es para presentar los resultados.
2145         print('\n A continuación se muestran sólo los productos que cuentan con al menos una calificación y ordenados de acuerdo a mayor cantidad de calificaciones 5 (La máxima). \n')
2146         print('*110')
2147         print('ID'                                     Nombre                                     No. calificaciones:          5   4   3   2   1')
2148         print('*110')
2149         for m in range(0,len(prod_bestreviews)):
2150             print('{:2d} {:<68s} {:3d} {:3d} {:3d} {:3d} {:3d}'.format(prod_bestreviews[m][0],prod_bestreviews[m][1],prod_bestreviews[m][2][1],prod_bestreviews[m][3][1],prod_bestreviews[m][4][1],pr
2151
2152     elif opt5 == 2:
2153         #La siguiente instrucción es para presentar los resultados.
2154         print('\n A continuación se muestran sólo los productos que cuentan con calificación distinta de cero: \n')
2155         print('*110')
2156         print('ID'                                     Nombre                                     Calificación promedio      No.ventas')
2157         print('*110')
2158         for m in range(0,len(product_avgscore)):
2159             print('{:2d} {:<68s} {:1f} {:>10d}'.format(product_avgscore[m][0],product_avgscore[m][1],product_avgscore[m][3],product_avgscore[m][2]))
2160
2161     else:
2162         print('No ingresó alguna de las respuestas permitidas. Intente de nuevo.')
2163
2164     elif option_chosen == 6:
2165         print('\n Opciones para visualización de productos con peores reseñas:
2166
2167     1 Con mayor cantidad de calificaciones de valor 1 (el mínimo).
2168     2 Menor calificación promedio.'')
2169
2170     opt6 = int(input('Ingrese el número de la opción deseada: '))
2171
2172     if opt6 == 1:
2173         #La siguiente instrucción es para presentar los resultados.
2174         print('\n A continuación se muestran sólo los productos que cuentan con al menos una calificación y ordenados de acuerdo a mayor cantidad de calificaciones 1 (La mínima). \n')
2175         print('*110')
2176         print('ID'                                     Nombre                                     No. calificaciones:          1   2   3   4   5')
2177         print('*110')
2178         for m in range(0,len(prod_worstreviews)):
2179             print('{:2d} {:<68s} {:3d} {:3d} {:3d} {:3d} {:3d}'.format(prod_worstreviews[m][0],prod_worstreviews[m][1],prod_worstreviews[m][6][1],prod_worstreviews[m][5][1],prod_worstreviews[m][4][

```

```

2180
2181     elif opt6 == 2:
2182         #La siguiente instrucción es para presentar los resultados.
2183         print('\n A continuación se muestran sólo los productos que cuentan con calificación distinta de cero: \n')
2184         print('-'*110)
2185         print('ID          Nombre          Calificación promedio          No.ventas')
2186         print('-'*110)
2187         for m in range(-1,-(len(product_avgscore)+1),-1):
2188             print('{:2d} {:<68s} {:1f} {:>10d}'.format(product_avgscore[m][0],product_avgscore[m][1],product_avgscore[m][3],product_avgscore[m][2]))
2189
2190     elif option_chosen == 7:
2191         print('\n A continuación los ingresos netos por mes: \n')
2192         print('-'*50)
2193         print('No.          Mes          Ingresos netos ($)')
2194         print('-'*50)
2195         for m in range(0,len(month_incomes)):
2196             print('{:2d} {:^20s} {:7d}'.format(month_incomes[m][0],month_incomes[m][1],month_incomes[m][2]))
2197
2198     admflag7 = False
2199     while user == "Administrador" and admflag7 == False:
2200         opt7 = input('¿Desea ver la lista de ingresos netos por mes ordenados de mayor a menor? (si/no): ')
2201         if opt7 == 'si':
2202             #Con esta instrucción se va a ordenar la lista de ingresos por mes.
2203             for i in range(0,len(month_incomes)):
2204                 for j in range(i+1,len(month_incomes)):
2205                     if month_incomes[i][2] <= month_incomes[j][2]:
2206                         temp = month_incomes[i]
2207                         month_incomes[i] = month_incomes[j]
2208                         month_incomes[j] = temp
2209                     #month_incomes[i],month_incomes[j] = month_incomes[j],month_incomes[i]
2210             #La siguiente instrucción es para presentar los resultados.
2211             print('\n A continuación los ingresos netos mensuales ordenados:\n')
2212             print('-'*50)
2213             print('No.          Mes          Ingresos netos ($)')
2214             print('-'*50)
2215             for m in range(0,len(month_incomes)):
2216                 print('{:2d} {:^20s} {:7d}'.format(month_incomes[m][0],month_incomes[m][1],month_incomes[m][2]))
2217             admflag7 = True
2218         elif opt7 == 'no':
2219             admflag7 = True
2220         else:
2221             print('No ingresó alguna de las respuestas permitidas. Intente de nuevo.')
2222
2223     elif option_chosen == 8:
2224         print('\n Cantidad de ventas totales:',annual_sales)
2225         print('\n Monto promedio de ventas mensuales:',round(avg_salesmonth,2))
2226
2227     elif option_chosen == 9:
2228         print('\n Monto anual de ingresos netos (sin contar productos con devolución):','$',annual_incomes)
2229         print('\n Monto promedio de ingresos netos mensuales:', '$',avg_incomesmonth)
2230
2231     elif option_chosen == 10:
2232         print('\n A continuación las ventas por mes ordenadas de mayor a menor:\n')
2233         print('-'*50)
2234         print('No.          Mes          No. ventas')
2235         print('-'*50)
2236         for m in range(0,len(month_sales)):
2237             print('{:2d} {:^20s} {:7d}'.format(month_sales[m][0],month_sales[m][1],month_sales[m][2]))
2238

```

```

2239     else:
2240         print('No ingresó algunas de las opciones mostradas. Intente de nuevo')
2241
2242     #Opciones extra para el administrador
2243     admflag = False
2244     while user == "Administrador" and admflag == False:
2245         adm_ans = input('Como Administrador tiene opciones extra disponibles ¿Desea verlas (si/no): ')
2246         if adm_ans == 'si':
2247             print('')
2248             -Opciones extra-
2249             1 Productos con menores ingresos netos por categoria.
2250             2 Productos ordenados de mayor a menor por cantidad de ingresos netos.
2251             3 Productos con mayor cantidad de stock.
2252             4 Razones ventas/búsquedas y búsquedas/ventas.
2253             ''')
2254             extflag = False
2255             while extflag == False:
2256                 optadm = int(input('Ingrese alguna de las opciones mostradas: '))
2257                 if optadm == 1:
2258                     print('\n A continuación Los productos ordenados de menor a mayor monto de ingresos netos por categoria:\n')
2259                     #Ahora se va a imprimir con el formato adecuado a todas las categorias y sus productos con menores ventas.
2260                     for c in range(0,len(categ_prod_incomes)):
2261                         print('')
2262                         print('+'*50)
2263                         print('Categoría', '[', categ_prod_incomes[c][0], ']', categ_prod_incomes[c][1])
2264                         print('-'*110)
2265                         print('ID                               Nombre                               Ingresos netos ($)')
2266                         print('-'*110)
2267                         for i in range(0,len(categ_prod_incomes[c][2:])):
2268                             print('{:2d} {:<68s} {:4d}'.format(categ_prod_incomes[c][i+2][0], categ_prod_incomes[c][i+2][1], categ_prod_incomes[c][i+2][2]))
2269                         extflag = True
2270
2271                 elif optadm == 2:
2272                     print('\n A continuación Los productos ordenados de mayor a menor monto de ingresos netos:\n')
2273                     #La siguiente instrucción es para presentar los resultados.
2274                     print('-'*110)
2275                     print('ID                               Nombre                               Ingresos netos ($)')
2276                     print('-'*110)
2277                     for m in range(0,len(prod_greatincomes)):
2278                         print('{:2d} {:<68s} {:4d}'.format(prod_greatincomes[m][0], prod_greatincomes[m][1], prod_greatincomes[m][3]))
2279
2280                     print('\n A continuación Los productos sin ingresos netos:\n')
2281                     #La siguiente instrucción es para presentar los resultados.
2282                     print('-'*110)
2283                     print('ID                               Nombre                               Ingresos netos ($)')
2284                     print('-'*110)
2285                     for m in range(0,len(prod_lessincomes)):
2286                         print('{:2d} {:<68s} {:4d}'.format(prod_lessincomes[m][0], prod_lessincomes[m][1], prod_lessincomes[m][3]))
2287                     extflag = True
2288
2289                 elif optadm == 3:
2290                     print('\n A continuación Los productos ordenados de mayor a menor stock:\n')
2291                     #La siguiente instrucción es para presentar los resultados.
2292                     print('-'*110)
2293                     print('ID                               Nombre                               Stock')
2294                     print('-'*110)
2295                     for m in range(0,len(product_stock)):
2296                         print('{:2d} {:<68s} {:4d}'.format(product_stock[m][0], product_stock[m][1], product_stock[m][3]))
2297                     extflag = True

```

```

2289         elif optadm == 3:
2290             print('\n A continuación Los productos ordenados de mayor a menor stock:\n')
2291             #La siguiente instrucción es para presentar los resultados.
2292             print('-'*110)
2293             print('ID                               Nombre                               Stock')
2294             print('-'*110)
2295             for m in range(0,len(product_stock)):
2296                 print('{:2d} {:<68s} {:4d}'.format(product_stock[m][0],product_stock[m][1],product_stock[m][3]))
2297             extflag = True
2298
2299         elif optadm == 4:
2300             print('\n A continuación Los productos ordenados de mayor a menor razón ventas/búsqueda:\n')
2301             #La siguiente instrucción es para presentar los resultados.
2302             print('-'*110)
2303             print('ID                               Nombre                               Ventas/Búsquedas')
2304             print('-'*110)
2305             for m in range(0,len(product_rate_sale_search)):
2306                 print('{:2d} {:<68s} {:4f}'.format(product_rate_sale_search[m][0],product_rate_sale_search[m][1],product_rate_sale_search[m][2]))
2307             print('')
2308             print('\n A continuación Los productos ordenados de mayor a menor razón búsquedas/ventas:\n')
2309             #La siguiente instrucción es para presentar los resultados.
2310             print('-'*110)
2311             print('ID                               Nombre                               Búsquedas/Ventas')
2312             print('-'*110)
2313             for m in range(0,len(product_rate_search_sale)):
2314                 print('{:2d} {:<68s} {:4f}'.format(product_rate_search_sale[m][0],product_rate_search_sale[m][1],product_rate_search_sale[m][2]))
2315             extflag = True
2316             extflag = True
2317
2318         else:
2319             print('No ingresó alguna de las respuestas permitidas. Intente de nuevo.')
2320             extflag = False
2321
2322     elif adm ans == 'no':
2323         admflag = True
2324     else:
2325         print('No ingresó alguna de las respuestas permitidas. Intente de nuevo.')
2326         admflag = True
2327
2328     #El siguiente algoritmo es para decidir si se abandona el programa o se continua.
2329     option_close = input('?Desea salir del programa? Escriba "si" para abandonar y "no" para continuar (sin comillas): ')
2330     if option_close == 'si':
2331         ans2 = True
2332     elif option_close == 'no':
2333         ans2 = False
2334     else:
2335         print('No ingreso alguna de las respuestas (si/no). Intente de nuevo')

```

Al inicio de todo el programa se encuentran los usuarios disponibles con sus contraseñas.

El usuario Administrador tiene acceso a opciones extra, precisamente:

Productos ordenados de menor a mayor cantidad de ventas.

Productos ordenados de menor a mayor cantidad de búsquedas.

Ingresos netos por mes ordenados de mayor a menor.

Así como otras cuatro opciones extra a las consignas iniciales recomendadas.

1. Productos con menores ingresos netos por categoría.
2. Productos ordenados de mayor a menor por cantidad de ingresos netos.
3. Productos con mayor cantidad de stock.
4. Razones ventas/búsquedas y búsquedas/ventas.