



LIFESTORE

ANÁLISIS DE INVENTARIO

PROYECTO 1

SEPTEMBRE 2020

REALIZADO POR:

Angélica Samantha Álvarez Herrera

ÍNDICE

I. INTRODUCCIÓN

2

II. OBJETIVO

3

III. SISTEMA DE ANÁLISIS

3

IV. ANÁLISIS

6

V. SOLUCIÓN

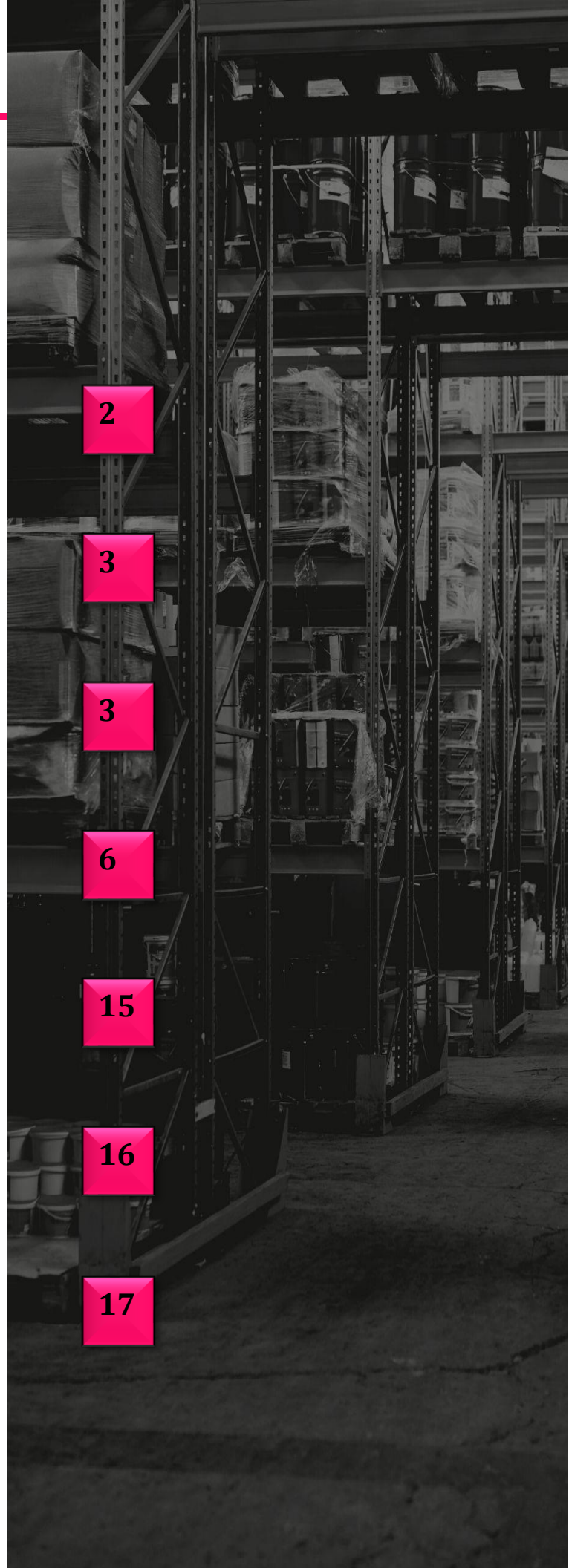
15

VI. CONCLUSIÓN

16

VII. ANEXOS

17





I. INTRODUCCIÓN

El inventario se define como todo el dinero que el sistema invierte en comprar bienes que pretende vender. Su manejo involucra guardar, ordenar y monitorear los niveles que este toma, siempre buscando el mismo objetivo: tener siempre el nivel correcto de inventario para llevar a cabo las ventas en cualquier momento.

En el caso de los productos tecnológico, el inventario corre el riesgo de quedar obsoleto, ya que constantemente se están creando nuevos productos. Esto implica que, de no realizar las ventas a tiempo, la empresa eventualmente los tendrá que vender a un valor depreciado y reemplazarlo por nuevos productos.

Es por lo anterior que se recalca la importancia de someter de forma regular a un análisis exhaustivo la información de la empresa de su inventario.

El proceso de analizar el inventario debe ser continuo y se debe buscar automatizarlo en la mayor medida, para en cualquier momento poder responder con un listado, detallado y actualizado de los recursos de la compañía.

En el presente documento se realizará un análisis del inventario de LifeStore, tomando como guía los datos recabados en el 2020 de las ventas, búsquedas y calificación dada por el usuario de los distintos productos que maneja la empresa.

LINK DE Github:

<https://github.com/Sam-Alvz/PROYECTO-01--ALVAREZ-ANGELICA.git>



LIFESTORE

LifeStore es una tienda virtual que maneja una amplia gama de artículos electrónicos, entre los que se encuentra productos de las más grandes marcas en computo, entretenimiento y electrónica. La empresa cuenta con mercancía dentro de las categorías de:

- ◆ Procesadores.
- ◆ Tarjetas de video.
- ◆ Tarjetas madre.
- ◆ Discos duros.
- ◆ Memorias USB.
- ◆ Pantallas.
- ◆ Bocinas.
- ◆ Audífonos



II. OBJETIVO

Se busca desarrollar un sistema de análisis en la que los administradores de la compañía puedan acceder, facilitando la observación de los resultados de la empresa en diferentes áreas, para tomar decisiones mejor informadas.

III. SISTEMA DE ANÁLISIS

CÓDIGO

La explicación detallada del código se encuentra en el Anexo 1.

ACERCA DEL CÓDIGO

El código se divide en tres partes esenciales:

- Login de Usuario
- Conformación de listas
- Presentación de los resultados

LOGIN DE USUARIO

En este apartado el usuario ingresa su información y el programa verifica si se trata de un administrador, cliente o una persona no registrada. Dependiendo de la respuesta a lo anterior, el programa le permite ver los reportes de la empresa en caso de que se administrador, o le niega el acceso y lo invita a la página de la empresa, si este resulta se cliente (Imagen 2). El usuario tiene hasta 3 oportunidades de intentar ingresar.

Imagen 1. Captura de la salida en el caso de poner la contraseña incorrecta.

```
Usuario: Admin1
Contraseña: ***
Contraseña incorrecta. Te quedan:2 intentos
```

Los usuarios registrados hasta el momento se muestran en la tabla 1.



Tabla 1. Usuarios registrados.

| Posición | Nombre de usuario | Contraseña |
|---------------|-------------------|------------|
| Administrador | Admin1 | 123 |
| Cliente | Cliente1 | 567 |
| Cliente | Cliente2 | 678 |
| Administrador | Admin2 | 234 |
| Administrador | Admin3 | 345 |

Imagen 2. Captura de la salida en el caso de que el usuario sea un cliente.

```
Usuario: Cliente1
Contraseña: 567
No tiene permitido el acceso.
Visita la pagina www.life\_store.com para conocer nuestros productos
```

CONFORMACIÓN DE LISTAS

Esta sección cumple con dos propósitos. En ella se llevan a cabo los conteos de frecuencia de ventas y búsquedas y las sumas de las calificaciones dadas por los clientes, y posteriormente, se generan listas con la información obtenida, para que el proceso de ordenamiento se realice más fácilmente.

PRESENTACIÓN DE RESULTADOS

Una vez que se haya verificado la identidad del administrador, este podrá escoger en el presente apartado la información de la compañía que desee ver.

Aquí se puede encontrar los siguientes datos:

1) Productos.

a) *Productos más vendidos.*

- i) Individual: Incluye producto, unidades vendidas (total y porcentaje con respecto al total de ventas), ingreso (total y porcentaje con respecto al total de ingresos).
- ii) Categoría: Por categoría, incluye productos más vendidos con sus unidades vendidas (total) e ingreso (total), participación de la categoría en las unidades vendidas y las ventas (ambos porcentajes).

b) *Productos menos vendidos.*

- i) Individual: Incluye producto, unidades vendidas (total), ingreso (total).



ii) Categoría: Por categoría, incluye productos menos vendidos con sus unidades vendidas (total) e ingreso (total) y el porcentaje de productos de cada categoría que no se vendió.

c) Productos más buscados.

i) Individual: Incluye producto, búsquedas realizadas, unidades vendidas y el porcentaje de búsquedas que concluyó en el usuario comprando el producto.

ii) Categoría: Por categoría, incluye productos con sus búsquedas realizadas y unidades vendidas, el porcentaje de búsquedas que concluyó en el usuario comprando un producto de la categoría y las búsquedas realizadas en total de la categoría.

d) Productos menos buscados.

i) Individual: Incluye producto, búsquedas realizadas, unidades vendidas.

ii) Categoría: Por categoría, incluye productos con sus búsquedas realizadas y unidades vendidas.

2) Reseñas.

a) Mejor calificados: Incluye los 20 productos mejor calificados con calificación y unidades devueltas.

b) Peor calificados: Incluye los 20 productos peor calificados con calificación y unidades devueltas.

3) Ingresos.

a) Ingresos mensuales: Mes, ventas, devoluciones y total.

b) Ventas v.s. Inventario: Para los 10 productos más vendidos y los 10 productos con mayor stock de los productos menos vendidos se incluye nombre del producto con sus unidades vendidas y unidades en stock.

c) Resumen de ingresos: Incluye ventas al año (unidades e ingreso), devoluciones al año (unidades y pérdida), ventas totales, ventas mensuales promedio y los 5 meses con mayores ventas.



IV. ANÁLISIS

ACERCA DE LOS DATOS.

Se cuenta con información de la empresa recopilada en el año 2020, conformada por tres listas.

1. Lifestore_products: Contiene la información de los productos dentro del inventario de la compañía en el presente año. Entre los datos tenemos el precio, la categoría y el stock actual del producto.
2. Lifestore_sale: Registro de las ventas realizadas, mencionando si el producto fue o no devuelto.
3. Lifestore_searches: Registro de las búsquedas realizadas en la página de la empresa.

Se obtuvo que durante el 2020:

- La empresa comercializó 96 productos.
- Se realizaron 283 ventas con 9 devoluciones.
- Se registraron 1033 búsquedas en el sitio web de la empresa.

Consideraciones:

Las fechas se trabajaron todas con el año 2020.

ACERCA DE LAS VENTAS:

De los registros correspondientes a las ventas, se tiene que 42 productos tuvieron al menos una venta.

Los productos más unidades vendidos fueron:

- SSD Kingston A400, 120GB, SATA III, 2.5", 7mm
- Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth

Con 50 y 45 unidades vendidas respectivamente.

Sin embargo, son los productos:

- Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth (Posición 2 en unidades vendidas)



- Procesador Intel Core i7-9700K S-1151 3.60GHz 8-Core 12MB Smart Cache (9na. Generación Coffee Lake) (Posición 12 en unidades vendidas)
- Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC 6GB 192-bit GDDR6 PCI Express x16 3.0 (Posición 11 en unidades vendidas)
- Procesador AMD Ryzen 5 3600 S-AM4 3.60GHz 32MB L3 Cache con Disipador Wraith Stealth (Posición 7 en unidades vendidas).

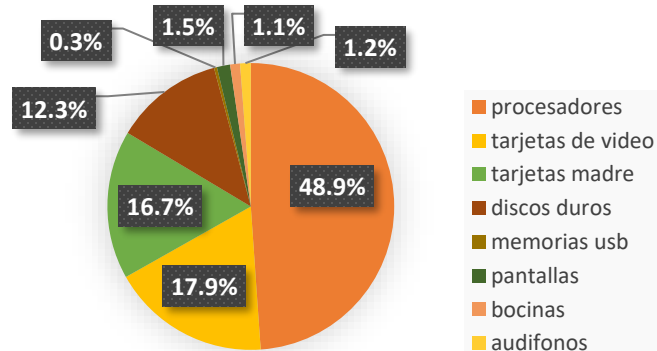
Tabla 2. 5 Productos con más unidades vendidas.

| | Producto | Unidades Vendidas | % del total | Total | % del total |
|---|---|-------------------|-------------|--------------|-------------|
| 1 | SSD Kingston A400 120GB SATA III 2.5" 7mm | 50 | 17.67% | \$12,950.00 | 1.70% |
| 2 | Procesador AMD Ryzen 5 2600 S-AM4 3.40GHz Six-Core 16MB L3 Cache | 42 | 14.84% | \$129,738.00 | 17.07% |
| 3 | Procesador Intel Core i3-9100F S-1151 3.60GHz Quad-Core 6MB Cache (9na. Generación - Coffee Lake) | 20 | 7.07% | \$35,580.00 | 4.68% |
| 4 | Tarjeta Madre ASRock Micro ATX B450M Steel Legend S-AM4 AMD B450 HDMI 64GB DDR4 para AMD | 18 | 6.36% | \$32,022.00 | 4.21% |
| 5 | SSD Adata Ultimate SU800 256GB SATA III 2.5" 7mm | 15 | 5.30% | \$13,335.00 | 1.75% |
| | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |

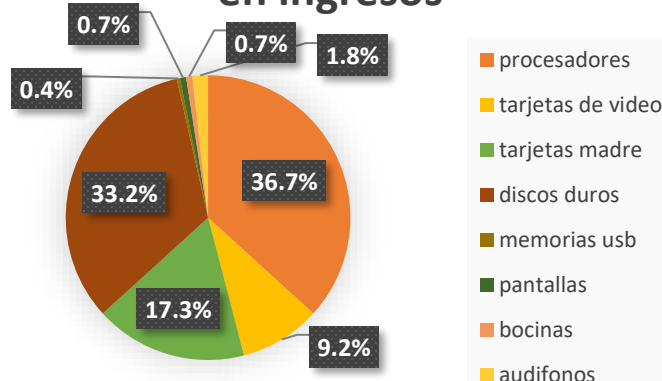
Con \$129,738, \$59,913, \$59,571 y \$54,717 de ganancias respectivamente, los que mayor aportaron a los ingresos de la empresa durante 2020.

Imagen 3. Participación de las categorías en las ventas e ingresos.

Porcentaje de participación en unidades vendidas



Porcentaje de participación en ingresos



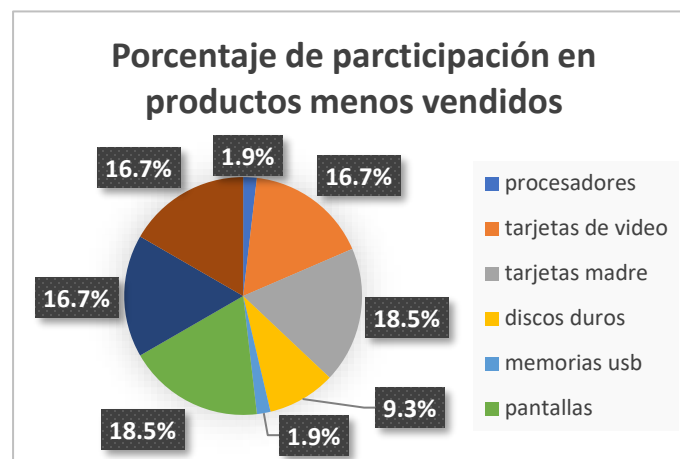


También se observa que con 13 de los productos obtenemos el 80% de las ventas y casi el 70% de las ganancias. Estos nos indica que el resto de los productos no aportan significativamente a las ventas y que estos productos que siempre deben estar disponibles para su venta.

Analizando la Imagen 3, donde se encuentran los datos de las ventas por categoría vemos que los procesadores, las tarjetas de video y las tarjetas madre son las que más contribuyen en las ventas, mientras que los procesadores, los discos duro y las tarjetas madre son las categorías que más contribuyen a las ganancias las ganancias

Se observó en los datos que 54 productos no registraron ninguna venta durante el 2020. Estos productos es importante analizar su inventario porque este no está rotando y puede resultar en pérdidas para la empresa. Las categorías con más productos en los menos vendidos son las tarjetas madre y las pantallas de televisión.

Imagen 4. Participación de las categorías en los productos menos vendidos



En el caso de las búsquedas, durante el 2020 se registraron 1033 búsquedas que se repartieron en 56 productos.

Los dos productos más buscados pertenecen a la categoría de discos duros y son:

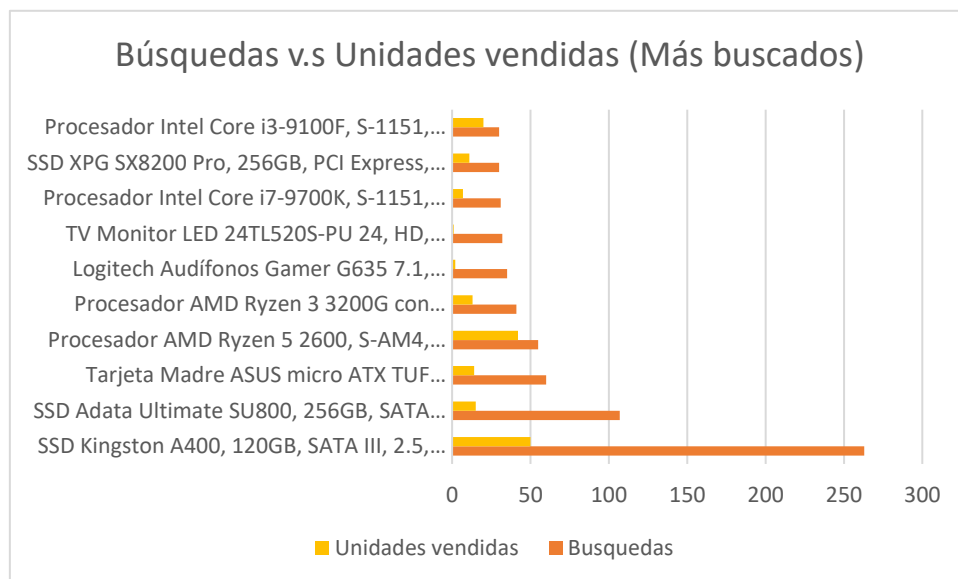
- SSD Kingston A400, 120GB, SATA III, 2.5, 7mm, con 263 búsquedas.
- SSD Adata Ultimate SU800, 256GB, SATA III, 2.5, 7mm, con 107 búsquedas.



Sin embargo, solo el 19% y el 14%, respectivamente, de las personas que los buscaron terminaron comprándolo.

En general, vemos que los productos más buscados no concluyen en una compra por parte de los clientes.

Imagen 5. Búsquedas realizadas comparadas con las ventas hechas.



De los productos que se encuentran en el top 20 de búsquedas, son los productos:

- Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth.
- Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake).
- Tarjeta madre, ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD.

los que tienden a producir más ventas después de ser buscados, con un 76%, 67% y 78% respectivamente.

Si hablamos en el nivel de las categorías, son los procesadores y las tarjetas madre los que son más comprados después de ser buscados.

Tabla 3. Búsquedas que concluyen en venta por categoría

| Categoría | Porcentaje |
|-------------------|------------|
| procesadores | 46.847% |
| tarjetas madre | 34.307% |
| tarjetas de video | 31.707% |
| bocinas | 22.222% |
| discos duros | 20.302% |
| audifonos | 7.813% |
| pantallas | 3.571% |
| procesadores | 46.847% |



Imagen 6. Cantidad de productos no buscados por categoría.



Por otro lado, 40 productos no registraron búsqueda alguna. Siendo las tarjetas madre, las pantallas y las bocinas las categorías en donde sobre más productos no se realizó ninguna búsqueda.

ACERCA DE LAS CALIFICACIONES:

Tabla 4. 5 primeros productos con mejor score.

De los 42 productos vendidos, se registraron las calificaciones dadas por los usuarios y se calculó el promedio. Se presentaron los 20 productos con mayor calificación promedio y 20 productos con menor calificación promedio.

Los 20 mejores promedios no presentaron ninguna devolución y las categorías con más productos con mejor calificación son tarjetas de video y los procesadores.

| | Producto | calificación |
|---|--|--------------|
| 1 | Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti, 6GB 192-bit GDDR6, PCI Express x16 3.0 | 5.00 |
| 2 | Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming, 8GB 128-bit GDDR6, PCI Express 4.0 | 5.00 |
| 3 | Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC, 4GB 128-bit GDDR5, PCI Express x16 3.0 | 5.00 |
| 4 | Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC, 8GB 128-bit GDDR6, PCI Express 4.0 | 5.00 |
| | | |
| 5 | Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Express 3.0 | 5.00 |
| | : | : |

Por otro lado, entre los productos con peores calificaciones destacan las tarjetas madre, que también son las que presentan más devoluciones, siendo la Tarjeta Madre AORUS micro ATX B450



AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD la que tiene más devoluciones, con 3 productos devueltos.

Tabla 5. 5 primero productos con peor score.

| | Producto | calificación | devoluciones |
|---|---|--------------|--------------|
| 1 | Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel | 1.00 | 1 |
| 2 | Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0 | 1.00 | 1 |
| 3 | Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD | 1.83 | 3 |
| 4 | Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel | 2.00 | 1 |
| 5 | Cougar Audífonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro. | 3.00 | 0 |
| | : | : | : |

ACERCA DE LOS INGRESOS

Durante el año, se vendieron un total de 283 unidades de los 96 productos que tiene la empresa, obteniendo un ingreso de \$760,177.00. De igual forma, se obtuvo un total de 9 devoluciones, lo que tuvo un costo para la empresa de \$22,261.00, obteniendo así una ganancia total de \$737,916.00 durante el año 2020.

Las ventas mensuales han ido disminuyendo a lo largo del año, teniendo un mínimo en los meses de octubre y diciembre en donde no se registro ninguna venta. El pico de las ventas se tuvo en el



mes de abril, en donde obtuvimos ventas de \$193,295.00, seguido del mes de marzo, en donde tuvimos un ingreso de \$164,729.00 en ventas.

Imagen 7. Ventas en el año 2020

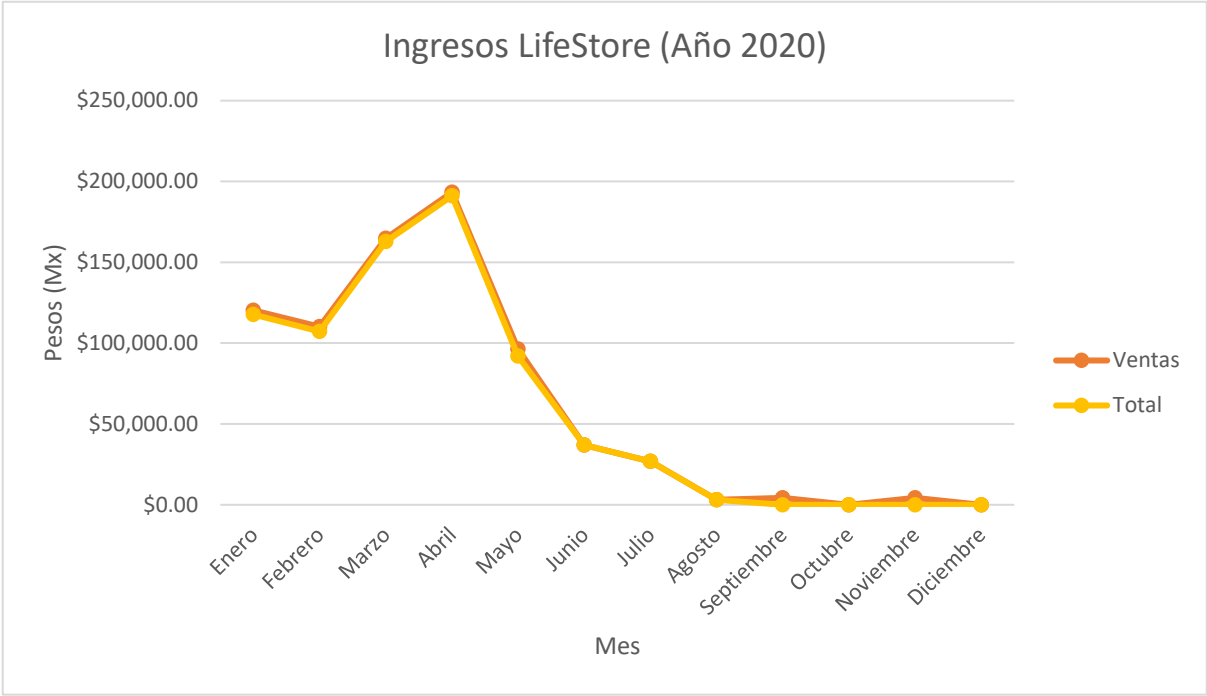


Tabla 5. 5 meses con mayores ventas.

| 5 Meses con mayores ventas: | | |
|-----------------------------|---------|---------------|
| | Mes | Ingreso |
| 1 | Abril | \$ 193,295.00 |
| 2 | Marzo | \$ 164,729.00 |
| 3 | Enero | \$ 120,237.00 |
| 4 | Febrero | \$ 110,139.00 |
| 5 | Mayo | \$ 96,394.00 |

Finalmente se realizó un análisis de las unidades vendidas comparándolo con su stock, de los 8 productos más vendidos y los 8 productos con mayor stock de los productos con 0 ventas.

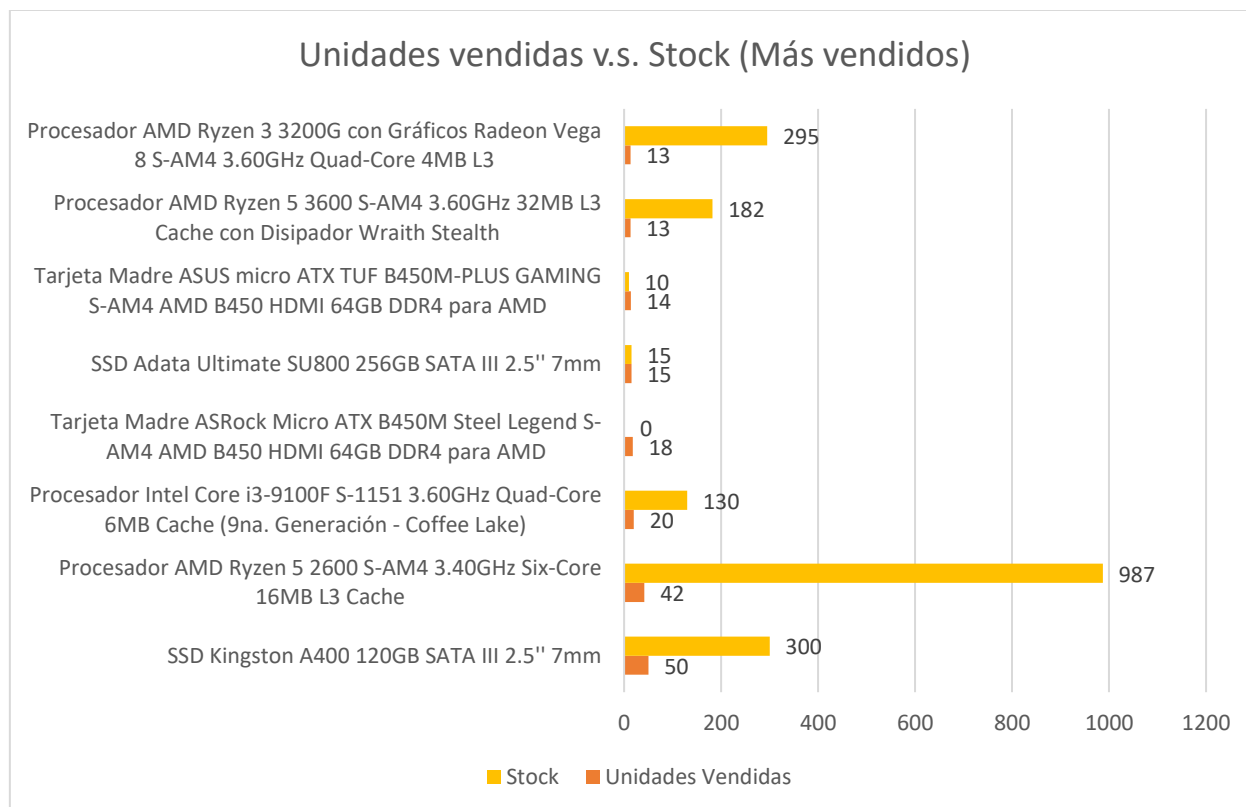
En el primer caso, vemos que mientras para unos productos hay exceso de stock, para



- Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING S-AM4 AMD B450 HDMI 64GB DDR4 para AMD
- SSD Adata Ultimate SU800 256GB SATA III 2.5" 7mm
- Tarjeta Madre ASRock Micro ATX B450M Steel Legend S-AM4 AMD B450 HDMI 64GB DDR4 para AMD

Las ventas están superando al stock. Esto es perjudicial para la empresa, ya que estos productos están entre los más vendidos por lo que deben estar disponibles en todo momento para los posibles clientes.

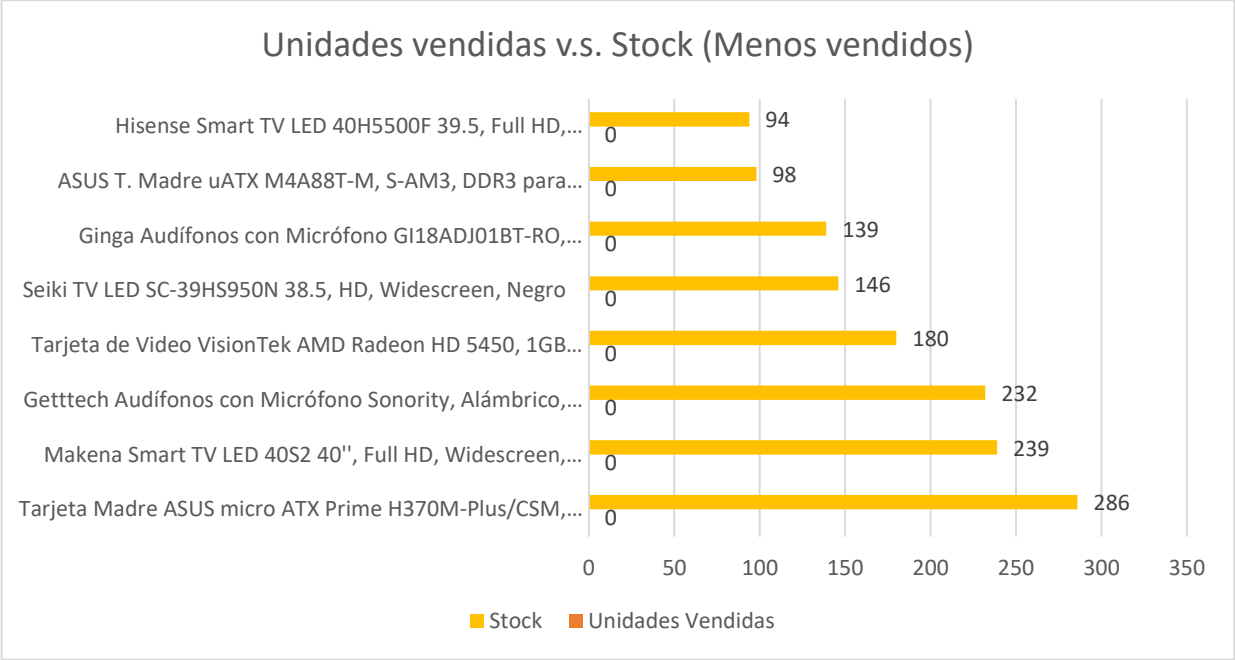
Imagen 8.



En el segundo caso, se observa una gran cantidad de stock para estos productos que no registran ventas. Esto resulta en la acumulación de inventario que presenta la empresa y termina resultando en un costo para mantenerlas guardadas.



Imagen 9.





V. SOLUCIÓN AL PROBLEMA

Para determinar que productos sacar del mercado nos fijamos en aquellos productos que tienen poco movimiento o que representan muy poco margen de utilidad, es decir, la lista de productos con menos ventas. No podemos eliminar todos estos, dado que aun necesitamos proveer de opciones a los usuarios. De ella nos quedaremos únicamente con los productos que pertenezcan a una categoría en la que sus búsquedas se traduzcan en ventas, es decir, procesadores, tarjetas madre y tarjetas de video.

En particular, la venta de pantallas, audífonos y bocinas no está resultando rentable para la empresa, ya que representan una mínima de las ventas y los ingresos, sin embargo, tenemos una cantidad significativa de ellos en el stock, por lo que aun vendiendo estos productos no recuperamos el costo de conservarlos en buen estado. Por lo que se pueden retirar del mercado o tenerlos bajo pedido con el proveedor. De esta forma al menos reduciremos el costo de mantenerlos.

Para reducir la acumulación de inventario identificamos inicialmente dos tipos de productos:

Aquellos que tienen un valor elevado pero una baja frecuencia de ventas. Entre ellos están los procesadores. Tienen un alto impacto financiero, pero no debemos tener un gran volumen en el almacén. En este caso, podremos reducir el nivel de inventario de estos productos con ofertas o reducir nuestra demanda de ellos, hasta disminuir su cantidad.

Aquellos con un valor bajo con niveles altos de ventas. Entre ellos están los discos duros. Estos tienen poco impacto financiero, pero están en constante rotación por lo que queremos siempre tenerlos disponibles.



VI. CONCLUSIÓN

Se realizó un análisis acerca de las ventas de los productos de LifeStore. Primeramente, notamos un mal manejo del inventario. Tiende a haber un exceso de stock en la mayoría de los productos, y al mismo tiempo se ha mostrado insuficiente para algunos de los productos con mayor demanda de los clientes. Esto es importante, porque la mayoría de las ventas de la empresa son de un número reducido de productos y perder una venta por no tenerlo disponible le da una mala imagen a la empresa.

Finalmente, se destaca la importancia del programa creado para el análisis, ya que facilitó acceder a la información relevante, lo que nos permitió observar a mayor detalle los problemas que se están presentando en la compañía.



VII. ANEXO

LOGIN DE USUARIO

```
6 #Usuarios registrados
7 Usuarios_Admin=[["Admin1","123"],["Admin2","234"],["Admin3","345"]] #Lista de administradores
8 Usuarios_Clientes=[["cliente1","567"],["cliente2","678"]] #Lista de clientes
9
```

Lista que contiene los usuarios que ya registrados.

```
10 ##LOGIN USUARIO/ADMINISTRADOR
11 intento=0 #Inicio contador de intentos
12 admin=False #bandera
13 cliente=False #bandera
14 while intento<3:
15     Usuario_inp=input("Usuario: ") #Solicita usuario
16     Contraseña_inp=input("Contraseña: ") #Solicita contraseña
17     #Verifica si el ingreso es de un Administrador
18     for usuario_a in Usuarios_Admin:
19         if usuario_a[0]==Usuario_inp:
20             if usuario_a[1]==Contraseña_inp:
21                 admin=True
22                 intento=3
23                 break
24             else:
25                 print("Contraseña incorrecta."+" Te quedan:"+str(3-intento-1)+" intentos")
26                 admin=True
27                 break
28         else:
29             admin=False
30             continue
31     #Verifica si el ingreso es de un Cliente
32     if admin==False:
33         for usuario_c in Usuarios_Clientes:
34             if usuario_c[0]==Usuario_inp:
35                 if usuario_c[1]==Contraseña_inp:
36                     print("No tiene permitido el acceso.")
37                     print("Visita la pagina www.life_store.com para conocer nuestros productos")
38                     cliente=True
39                     break
40             else:
41                 print("Contraseña incorrecta."+" Te quedan:"+str(3-intento-1)+" intentos")
42                 cliente=True
43                 break
44             else:
45                 cliente=False
46                 continue
47     #Avisa si no se encuentra el usuario
48     if cliente==False and admin==False:
49         print("No se encontro al usuario."+" Te quedan:"+str(3-intento-1)+" intentos")
50         intento+=1
51     #Avisa si ya se acabaron los intentos
52     if intento==3 and cliente==False:
53         admin=False
54         print("Ya se realizaron 3 intentos. Vuelva mas tarde")
55
```

Se inicia contador de intento y se repite el ciclo mientras estos sean menores a 3.

Se solicita al usuario sus datos.

Verifica si el usuario está en la lista de administradores. Si es así verifica si su contraseña es correcta, y hace admin=True y termina el ciclo si lo es o le señala que se equivocó y termina el ciclo. Sino es administrador, lo indica para que el ciclo continúe.

Si el usuario no es administrador, verifica si el usuario es tá la lista de clientes. Si es así verifica si su contraseña es correcta, y hace cliente=True y termina el ciclo si lo es o le señala que se equivocó y termina el ciclo. Sino es cliente, lo indica para que el ciclo continúe.

Si el usuario no es cliente ni admin. se imprime que no se reconoce sus datos.

Si no se reconoce los datos o se equivoca en la contraseña aumente el contador de intentos. Si los intentos son iguales a 3 termina el ciclo.



CONFORMACIÓN DE LISTAS

```
56 while admin==True:
57     #Inicio de listas con contadores
58     vend = [0]*len(lifystore_products) #ventas
59     devol = [0]*len(lifystore_products) #devoluciones
60     reseña= [0]*len(lifystore_products) #reseñas
61     vend_mensual=[0]*12 #ventas mensuales
62     devol_mensual=[0]*12 #devoluciones mensuales
63     unidades_devol=0 #Unidades devueltas
64     busq= [0]*len(lifystore_products) #busquedas
65
66     #Contar ventas, devoluciones y sumar reseñas
67     for sales in lifystore_sales:
68         vend[sales[1]-1] += 1
69         reseña[sales[1]-1] += sales[2]
70         vend_mensual[int(sales[3][3:5])-1]+=lifystore_products[sales[1]-1][2]
71         if sales[4]==1:
72             devol[sales[1]-1] += 1
73             unidades_devol+=1
74             devol_mensual[int(sales[3][3:5])-1]+=lifystore_products[sales[1]-1][2]
```

Se inician vectores con 0 que van a servir para contar las frecuencias de las ventas, las devoluciones, las búsquedas, las unidades devueltas y sumar las reseñas.

Se recorre cada lista de ventas. Si se vendió el producto con id i se le suma 1 a la índice i-1 del vector de ventas. En las reseñas se le suma la calificación dada. Para las ventas mensuales, se vuelve entero el mes de la fecha y se le suma a la índice mes-1 el precio del producto de id i. Si se realizó devolución del producto con id i se le suma 1 a la índice i-1 del vector de devoluciones.

```
76     #Contar busquedas
77     for search in lifystore_searches:
78         busq[search[1]-1] +=1
```

Se recorre cada lista de búsquedas. Si se buscó el producto con id i se le suma 1 a la índice i-1 del vector de búsquedas.

```
80
81     #OPTENCION DE LISTAS
82     categorias=[] #categorias=[categoria]
83     ventas= [] #ventas=[unidades,ventas,producto,ingreso,categoria]
84     busquedas= [] #busquedas=[busquedas,producto,unidades,ventas,categoria]
85     calificacion=[] #calificacion=[calificacion.prom,producto,devoluciones]
86
87     #Agregado de datos para ventas, busquedas y calificacion
88     for productos in lifystore_products:
89         #categorias de los productos
90         if productos[3] not in categorias:
91             categorias.append(productos[3])
92         #ventas
93         lista=[vend[productos[0]-1],productos[1],productos[2]*vend[productos[0]-1],productos[3],productos[4]]
94         ventas.append(lista)
95         #busquedas
96         lista=[busq[productos[0]-1],productos[1],vend[productos[0]-1],productos[3]]
97         busquedas.append(lista)
98         #calificaciones de los productos
99         if vend[productos[0]-1]==0:
100             lista=[0,productos[1],devol[productos[0]-1]]
101             calificacion.append(lista)
102             continue
103         lista=[reseña[productos[0]-1]/vend[productos[0]-1],productos[1],devol[productos[0]-1]]
104         calificacion.append(lista)
```

Se crean listas 0 que van a servir para guardar los datos solicitados para imprimirlos posteriormente.

Se recorren las listas de los productos.

En el caso de las categorías, se verifica si la categoría del producto no está en la lista de categorías y si es así, se añade a la lista.

Seguido a ello se van creando arreglos con los elementos del producto que se desean, procurando colocar primero el elemento con el cual se va a ordenar. El arreglo creado se anexa a la lista que corresponda.

```
114
115     ventas_anual=0 #Ingreso ventas al año
116     devoluciones_anual=0 #Perdida devoluciones al año
117     ventas_mensuales=[] #ventas_mensuales=[ventas,mes,devoluciones,total]
118     meses=['Enero','Febrero','Marzo','Abril','Mayo','Junio','Julio','Agosto','Septiembre','Octubre','Noviembre','Diciembre']
119     #Agregado de datos para ingresos
120     for i in range(12):
121         lista=[vend_mensual[i],meses[i],devol_mensual[i],vend_mensual[i]-devol_mensual[i]]
122         ventas_anual+=vend_mensual[i]
123         devoluciones_anual+=devol_mensual[i]
124         ventas_mensuales.append(lista)
125
126     ventas_totales=ventas_anual-devoluciones_anual #Ingreso total
```

Se inicial contadores de ventas y devoluciones anuales, y se crea una lista para guardar la información de las ventas de cada mes.

Se crea un arreglo con los elementos del mes que se desean, poniendo primero las ventas (para ordenarlo después con ellas). Y se suma a la posición i la venta y la devolución del mes i.

Se calculan las ventas totales con la diferencia de ventas anuales menos devoluciones anuales.



PRESENTACIÓN DE RESULTADOS

Este es la estructura del código para mostrar la información de la empresa.

```
#Mostrar opciones al administrador
mostrar=True
while mostrar==True:
    print("Opciones:\n1) Productos\n2) Reseñas de productos \n3) Ingresos ")
    Opcion=input("Elige una opción: ")
    #PRODUCTOS
    if Opcion=="1":
        #####CODIGO#####
        mostrar=False
    #RESEÑAS
    elif Opcion=="2":
        #####CODIGO#####
        mostrar=False
    #INGRESOS
    elif Opcion=="3":
        #####CODIGO#####
        mostrar=False
    else:
        print("Error. El valor ingresado no se encuentra dentro de las opciones\n")

consultar=True
#Pregunta al usuario si desea consultar otro apartado
while consultar==True:
    consulta=input("Deseas realizar otra consulta (Si/No): ")
    if consulta=="No":
        admin=False
        consultar=False
    elif consulta=="Si":
        admin=True
        consultar=False
    else:
        print("Error. El valor ingresado no se encuentra dentro de las opciones\n")
```

Bandera para que no vuelvan a salir las opciones si no las pides.

Se te solicita la opción que deseas y según tu respuesta se ejecuta el código de esa opción. Al finalizar el proceso se hace mostrar=False para que no salgan las opciones nuevamente.

Te señala si metes un valor equivocado y te vuelven a salir las opciones para que elijas nuevamente.

Pregunta si deseas otra consulta. Si no es así se hace admin=False y se termina el programa. Si deseas seguir la consulta el proceso se reinicia y te vuelve a dar a escoger opciones de información.

Te señala si metes un valor equivocado y te vuelven a salir las opciones para que elijas nuevamente.



Si seleccionaste para ver la información de los **productos**

```
if Opcion=="1":
    error=True #bandera
    while error==True:
        #Opciones de productos
        print("Opciones:\n1) Productos mas vendidos\n2) Productos menos vendidos \n3) Productos mas buscados \n4) Productos menos buscados")
        Opcion_ventas=input("Elige una opción: ")
        if Opcion_ventas=="1":
            print("PRODUCTOS MÁS VENDIDOS: \n")
            sub_error=True #bandera
            while sub_error==True:
                ventas.sort(reverse=True)
                #Opciones de productos mas vendidos
                ventas_select=input("Opciones:\n1) General \n2) Por categoria \n Elige una opcion: ")
                if ventas_select=="1":
                    print("PRODUCTOS MÁS VENDIDOS: \n")
                    n_veces=0
                    while n_veces<50 and ventas[n_veces][0]!=0:
                        print(str(n_veces+1)+" " +ventas[n_veces][1])#Nombre del producto
                        #calcula el % de unidades vendidas con respecto del total del producto
                        porcentaje_unidades=str('{:.2f}'.format(ventas[n_veces][0]*100/len( lifestore_sales)))
                        print("Unidades Vendidas: "+str(ventas[n_veces][0]) + " (% Total: "+porcentaje_unidades+"%)")
                        #calcula el % de ingreso con respecto del total del producto
                        porcentaje_ventas=str('{:.2f}'.format(ventas[n_veces][2]*100/ventas_anual))
                        print("Ventas: $" +str(ventas[n_veces][2])+" (% Total: "+porcentaje_ventas+"%) \n")
                        n_veces+=1
                    sub_error=False
                elif ventas_select=="2":
                    print("PRODUCTOS MÁS VENDIDOS POR CATEGORIA: \n")
                    for categoria in categorias:
                        porcentaje_unidades=0 #Inicio contador de %unidades vendidas por categoria
                        porcentaje_ventas=0 #Inicia contador de %ingresos por categoria
                        n_veces=0
                        print("CATEGORIA: "+categoria)
                        for venta in ventas:
                            if venta[3]==categoria and venta[0]!=0:
                                print(str(n_veces+1)+" "+venta[1]) #Nombre del producto
                                print("Unidades Vendidas: "+str(venta[0]))
                                print("Total: $" +str(venta[2])+"\n")
                                n_veces+=1
                                porcentaje_unidades+=venta[0]/len( lifestore_sales) #unidades.vendidas/total.de.ventas
                                porcentaje_ventas+=venta[2]/ventas_anual #ventas/ingreso.total
                        print("% de unidades vendidas: "+str('{:.2f}'.format(porcentaje_unidades*100))+ "%")
                        print("% de ventas: "+str('{:.2f}'.format(porcentaje_ventas*100))+ "% \n\n")
                        sub_error=False
                else:
                    print("Error. El valor ingresado no se encuentra dentro de las opciones\n")
                    error=False
```

Se te solicita la opción que deseas y según tu respuesta se ejecuta el código de esa opción.

Para cada producto: Mientras haya tenido ventas: Te imprime el producto, las unidades vendidas y su ganancia.
Calcula % de unidades vendidas con respecto al total (unidades vendidas/unidades vendidas totales) y lo imprime
Calcula % de ingreso con respecto al total (ventas/ventas totales) y lo imprime

Para cada categoría: Te imprime sus productos con sus ventas.
Calcula % de unidades vendidas con respecto al total (unidades vendidas/unidades vendidas totales) y lo imprime
Calcula % de ingreso con respecto al total (ventas/ventas totales) y lo imprime



```
elif Opcion_ventas=="2":
    sub_error=True #bandera
    while sub_error==True:
        print("PRODUCTOS MENOS VENDIDOS: \n")
        ventas.sort()
        #Opciones de productos menos vendidos
        ventas_select=input("Opciones:\n1) General \n2) Por categoria \n Elige una opcion: ")
        if ventas_select=="1":
            print("PRODUCTOS MENOS VENDIDOS: \n")
            n_veces=0
            while n_veces<50 or ventas[n_veces][0]==0:
                print(str(n_veces+1)+" " +ventas[n_veces][1])#Nombre del producto
                print("Unidades Vendidas: "+str(ventas[n_veces][0]))
                print("Total: $" +str(ventas[n_veces][2])+"\n")
                n_veces+=1
            sub_error=False
        elif ventas_select=="2":
            print("PRODUCTOS MENOS VENDIDOS POR CATEGORIA: \n")
            i=0 #indice
            for categoria in categorias:
                n_veces=0
                print("CATEGORIA: "+categoria)
                for venta in ventas:
                    if venta[3]==categoria and venta[0]==0:
                        print(str(n_veces+1)+" " +venta[1])#Nombre del producto
                        print("Unidades Vendidas: "+str(venta[0])+"\n")
                        n_veces+=1
                    i+=1
                print("\n")
            sub_error=False
        else:
            print("Error. El valor ingresado no se encuentra dentro de las opciones\n")
    error=False
```

Se te solicita la opción que deseas y según tu respuesta se ejecuta el código de esa opción.

Para cada producto: Mientras NO haya tenido ventas: Te imprime el producto, las unidades vendidas y su ganancia.

Para cada categoría: Te imprime sus productos con sus ventas.



```
elif Opcion_ventas=="3":
    sub_error=True #bandera
    while sub_error==True:
        print("PRODUCTOS MÁS BUSCADOS: \n")
        busquedas.sort(reverse=True)
        #Opciones de productos mas buscados
        busquedas_select=input("Opciones:\n1) General \n2) Por categoria \n Elige una opcion: ")
        if busquedas_select=="1":
            print("PRODUCTOS MÁS BUSCADOS: \n")
            n_veces=0
            while busquedas[n_veces][0]!=0:
                print(str(n_veces+1)+" " +busquedas[n_veces][1])#nombre del producto
                print("Busquedas: "+str(busquedas[n_veces][0]))
                print("Unidades vendidas: "+str(busquedas[n_veces][2])+"\n")
                #calcula las busquedas que concluyen en ventas
                porcentaje_busqueda=busquedas[n_veces][2]*100/busquedas[n_veces][0] #ventas/busquedas
                print("% que concluye en venta: "+ str('{:.2f}'.format(porcentaje_busqueda))+"%\n\n")
                n_veces+=1
            sub_error=False
        elif busquedas_select=="2":
            print("PRODUCTOS MÁS BUSCADOS POR CATEGORIA: \n")
            for categoria in categorias:
                n_veces=0
                numero_ventas=0 #Inicia contador de ventas
                numero_busqueda=0 #Inicia contador de busquedas
                print("CATEGORIA: "+categoria)
                for busqueda in busquedas:
                    if busqueda[3]==categoria and busqueda[0]!=0:
                        print(str(n_veces+1)+" " +busqueda[1])#Nombre del producto
                        print("Busquedas: "+str(busqueda[0]))
                        print("Unidades vendidas: "+str(busqueda[2]))
                        numero_ventas+=busqueda[2]
                        numero_busqueda+=busqueda[0]
                        n_veces+=1
                    if numero_busqueda==0:
                        print("No se realizaron busquedas en esta categoria\n\n")
                        continue
                #calcula las busquedas que concluyen en ventas
                porcentaje_busqueda=numero_ventas*100/numero_busqueda
                print("\n% que concluye en venta: "+str('{:.2f}'.format(porcentaje_busqueda)))
                #calcula la participacion de busquedas de cada categoria
                participacion_busquedas=numero_busqueda
                print("Busquedas realizadas: "+str('{:.2f}'.format(participacion_busquedas))+"\n\n")
            sub_error=False
        else:
```

Se te solicita la opción que deseas y según tu respuesta se ejecuta el código de esa opción.

Para cada producto: Mientras haya búsquedas: Te imprime el producto, sus búsquedas y las unidades vendidas.
Calcula % de búsquedas que termina en ventas (unidades vendidas/búsqueda) y lo imprime

Para cada categoría: Te imprime sus productos con sus búsquedas y ventas. Suma las ventas y búsquedas por categoría.

Calcula % de búsquedas que termina en ventas por categoría (unidades vendidas/búsqueda) y lo imprime

Imprime las búsquedas realizadas por categoría.



```
elif Opcion_ventas=="4":
    sub_error=True #bandera
    while sub_error==True:
        print("PRODUCTOS MENOS BUSCADOS: \n")
        busquedas.sort()
        #Opciones de productos menos buscados
        busquedas_select=input("Opciones:\n1) General \n2) Por categoria \n3 Elige una opcion: ")
        if busquedas_select=="1":
            n_veces=0
            while n_veces<50 and busquedas[n_veces][0]==0:
                print(str(n_veces+1)+" " +busquedas[n_veces][1])#Nombre del producto
                print("Busquedas: "+str(busquedas[n_veces][0]))
                print("Unidades vendidas: "+str(busquedas[n_veces][2])+"\n")
                n_veces+=1
            sub_error=False
        elif busquedas_select=="2":
            print("PRODUCTOS MENOS BUSCADOS POR CATEGORIA: \n")
            for categoria in categorias:
                n_veces=0
                print("CATEGORIA: "+categoria)
                for busqueda in busquedas:
                    if busqueda[3]==categoria and busqueda[0]==0:
                        print(str(n_veces+1)+" " +busqueda[1])#Nombre del producto
                        print("Busquedas: "+str(busqueda[0]))
                        print("Unidades vendidas: "+str(busqueda[2])+"\n")
                        n_veces+=1
                print("Productos no buscados: "+ str(n_veces))
            print("\n\n")
            sub_error=False
        else:
            print("Error. El valor ingresado no se encuentra dentro de las opciones\n")
            error=False
    else:
        print("Error. El valor ingresado no se encuentra dentro de las opciones\n")
        mostrar=False
#RESEÑAS
```

Se te solicita la opción que deseas y según tu respuesta se ejecuta el código de esa opción.

Para cada producto: Mientras NO haya tenido búsquedas: Te imprime el producto, las búsquedas y las unidades vendidas.

Para cada categoría: NO haya tenido búsquedas: Te imprime el producto, las búsquedas y las unidades vendidas. Te imprime cuantos productos fueron no buscados de cada categoría.

Te señala si metiste un valor incorrecto al escoger tu búsqueda y te vuelven a salir las opciones.



Si seleccionaste para ver la información de las **reseñas**

```

| mostrar=False
#RESEÑAS
elif Opcion=="2":
    error=True #bandera
    while error==True:
        #Opciones de reseñas
        print("Opciones:\n1) Mejores reseñas\n2) Peores reseñas")
        Opcion_reseñas=input("Elige una opción: ")
        if Opcion_reseñas=="1":
            print("PRODUCTOS MEJOR CALIFICADOS: \n")
            calificacion.sort(reverse=True)
            n_veces=0
            while n_veces<20:
                print(str(n_veces+1)+" " +calificacion[n_veces][1])#Nombre del producto
                print("Calificacion de los usuarios: "+str('{:.2f}'.format(calificacion[n_veces][0])))
                print("Unidades Devueltas: "+str(calificacion[n_veces][2])+"\n")
                n_veces+=1
            error=False
        elif Opcion_reseñas=="2":
            print("PRODUCTOS PEOR CALIFICADOS: \n")
            calificacion.sort()
            n_veces=0
            i=0
            while n_veces<20:
                if calificacion[i][0]==0:
                    i+=1
                    continue
                print(str(i+1)+" " +calificacion[i][1])#Nombre del producto
                print("Calificacion de los usuarios: "+str('{:.2f}'.format(calificacion[i][0])))
                print("Unidades Devueltas: "+str(calificacion[i][2])+"\n")
                i+=1
                n_veces+=1
            error=False
        else:
            print("Error. El valor ingresado no se encuentra dentro de las opciones\n")
            mostrar=False
    mostrar=False
#FIN DEL PROGRAMA
```

Se te solicita la opción que deseas y según tu respuesta se ejecuta el código de esa opción.

Se ordenan los productos de mayor a menor score y se imprimen los 20 mayores, con nombre, calificación y unidades devueltas.

Se ordenan los productos de menor a mayor score y se imprimen los 20 menores, con nombre, calificación y unidades devueltas.



Si seleccionaste para ver la información de los **ingresos**

```
#INGRESOS
elif Opcion=="3":
    print("Ingresos")
    error=True #bandera
    while error==True:
        #Opciones de ingresos
        print("Opciones:\n1) Ingresos mensuales\n2) Ventas vs Inventario \n3) Resumen de ingresos")
        Opcion_ventas_mensuales=input("Elige una opción: ")
        if Opcion_ventas_mensuales=="1":
            n_veces=0
            while n_veces<12:
                print("Mes: "+ ventas_mensuales[n_veces][1])
                print("Ventas: $" + str(ventas_mensuales[n_veces][0]))
                print("Devoluciones: $" + str(ventas_mensuales[n_veces][2]))
                print("TOTAL: $" + str(ventas_mensuales[n_veces][3])+'\n')
                n_veces+=1
            error=False
        elif Opcion_ventas_mensuales=="2":
            #Ventas vs inventarios de productos mas vendidos
            print("MAS VENDIDOS")
            ventas.sort(reverse=True)
            n_veces=0
            while n_veces<8:
                print(str(n_veces+1)+" "+ventas[n_veces][1])#nombre del producto
                print("Ventas: "+ str(ventas[n_veces][0]))
                print("Stock: "+ str(ventas[n_veces][4]))

                n_veces+=1
            print("\n\n MENOS VENDIDOS")
            #Ventas vs inventarios de productos menos vendidos
            menos_vendidos=[] #menos_vendidos[stock,producto,unidades vendidas]
            for venta in ventas:
                if venta[0]==0:
                    lista=[venta[4],venta[1],venta[0]]
                    menos_vendidos.append(lista)
            menos_vendidos.sort(reverse=True)
            n_veces=0
            while n_veces<8:
                print(str(n_veces+1)+" "+menos_vendidos[n_veces][1])#Nombre del produ
                print("Ventas: "+ str(menos_vendidos[n_veces][2]))
                print("Stock: "+ str(menos_vendidos[n_veces][0]))
                n_veces+=1

        elif Opcion_ventas_mensuales=="3":
            ventas_mensuales.sort(reverse=True) #ordena las ventas mensuales
            print("LifeStore 2020")
            print("Ventas al año:\n Unidades:"+str(len(lifestore_sales))+ " Ingreso: $" + str(ventas_anual))
            print("Devoluciones al año:\n Unidades:"+str(unidades_devol)+" Perdida: $" + str(devoluciones_anual))
            print("Ventas Totales: $" + str('{:.2f}'.format(ventas_totales)))
            print("Ventas mensuales promedio: $" + str('{:.2f}'.format(ventas_totales/12)))
            print("5 Meses con mayores ventas: ")
            n_veces=0
            #imprime 5 meses con mayores ventas
            while n_veces<5:
                print(" "+ventas_mensuales[n_veces][1]+ ". Ventas: $" + str(ventas_mensuales[n_veces][0])+'\n')
                n_veces+=1
            error=False
```

Para cada mes se imprime el mes, las ventas, las devoluciones y el total (Ventas menos devoluciones)

Ordena las ventas de mayor a menor y para los 8 productos con más ventas imprime el nombre sus ventas y su stock.

Crea una nueva lista con el stock, el producto y las unidades vendidas de los productos con 0 ventas. Los ordena según su stock de mayor a menor y para los 8 productos con más stock imprime el nombre sus ventas y su stock.

Imprime datos de la empresa ya calculados anteriormente.

Ordena las ventas mensuales e imprime los 5 meses con mayores ventas.