

# **LifeStore**

## **Reporte-01**



**Hecho por:**

**Daniel Omar Morán Carrizal**

**Febrero 2022**

# **LIFESTORE - REPORTE 01**

Hecho por: Daniel Omar Morán Carrizal  
Febrero 2022

## **Índice**

<b>Introducción.....</b>	<b>3</b>
<b>Desarrollo y Resultados .....</b>	<b>3</b>
1. Los 5 productos más vendidos .....	4
2. Los 10 productos más buscados .....	5
3. Productos (por categoría) con menores ventas .....	6
4. Productos (por categoría) con menores búsquedas.....	7
5. Mejores y Peores reseñas por producto .....	9
6. Información financiera mensual y anual de LifeStore .....	9
<b>Solución al Problema .....</b>	<b>11</b>
<b>Conclusión.....</b>	<b>12</b>
<b>Anexos .....</b>	<b>12</b>
1. Login de LifeStore.....	12
2. Código empleado .....	13

## Introducción

La Gerencia de Ventas de la empresa **LifeStore** identificó una reducción en las búsquedas de un grupo importante de productos propiciado una disminución sustancial de sus ventas en el último trimestre.

Dado lo anterior, en el presente reporte de identifican y dan a conocer diferentes productos, así como la información financiera correspondiente a los siguientes rubros:

1. Productos más vendidos y productos rezagados.
2. Productos por reseña en el servicio.
3. Total de ingresos y ventas mensual de LifeStore
4. Total de ingresos anual de LifeStore

Finalmente, acorde a un análisis se proponen diferentes soluciones para atender el problema de disminución de ventas y así, mejorar la situación actual de la empresa.

## Desarrollo y Resultados

Para la filtración de la base de datos de LifeStore, se realizaron 6 funciones diferentes en Python: *top\_sales\_product()*, *top\_searches\_product()*, *category\_bottom\_sales()*, *category\_bottom\_searches()*, *product\_review()*, *inf\_sales\_LifeStore()*. En cada una de ellas se almacenan los resultados de cada uno de los apartados solicitados.

Para cada una de las funciones anteriores se aplicó el mismo principio, el cual consiste en la creación de diferentes diccionarios que ayudarán a brincar entre las 3 listas dadas, gracias al id del producto, obteniendo cada una de las filtraciones solicitadas por la Gerencia de Ventas. A continuación, en el siguiente enlace de GitHub, se encuentra el código en Python, donde es posible verificar lo estipulado anteriormente. Cada paso se encuentra comentado para mayor entendimiento.

<https://github.com/danielmoraan/EmTech-python.git>

Para acceder a la información, es necesario ingresar un usuario y una contraseña. A continuación, se muestran los datos:

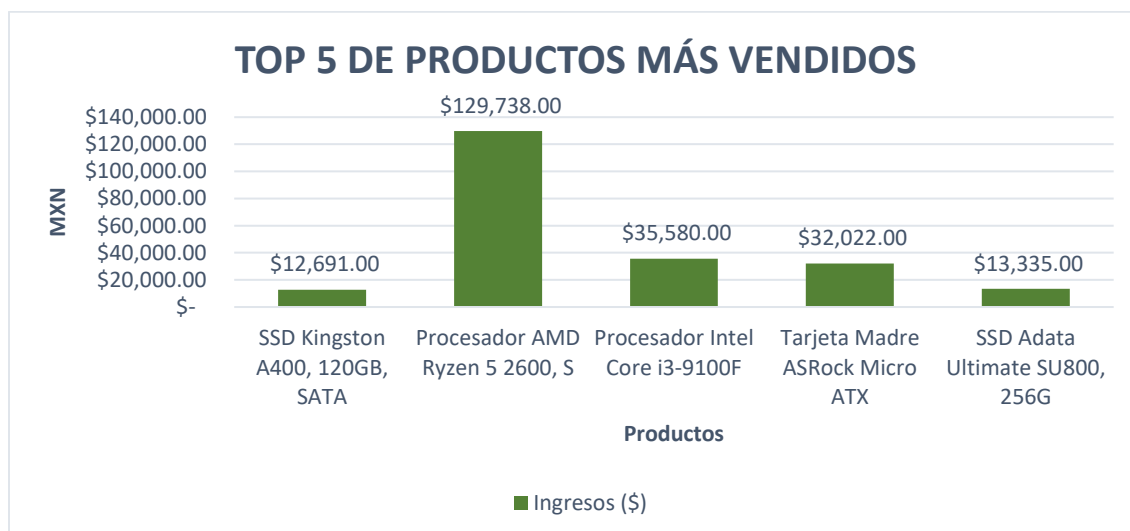
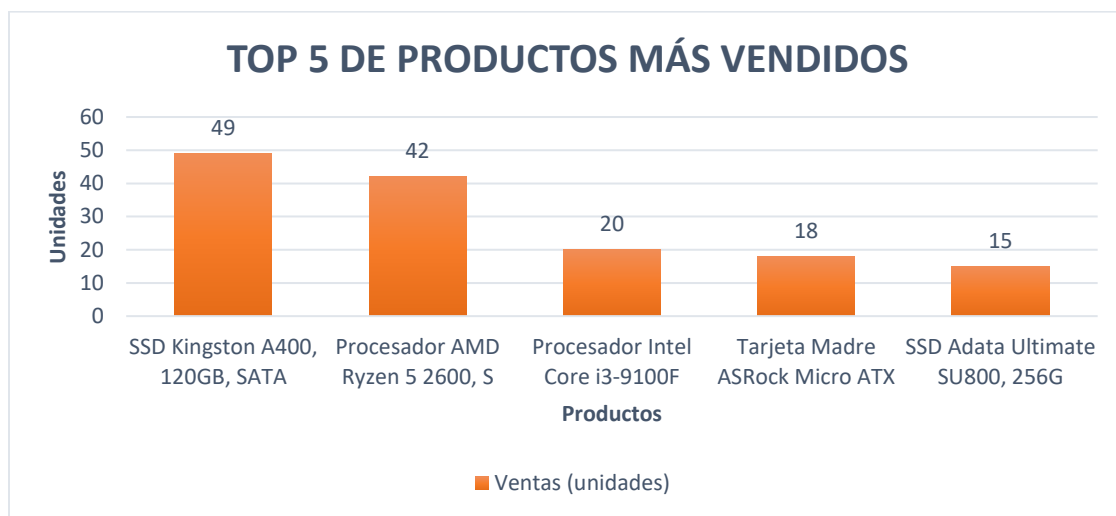
No.	Usuario	Contraseña
1	danielmoraan	Pyth0n3mTech
2	jimmyalonso	3mTechD4t4

**Nota:** Solo se cuentan con 3 oportunidades para ingresar, de lo contrario, el programa finalizará y no se podrá visualizar la información.

## 1. Los 5 productos más vendidos

Para obtener los 5 productos con mayores ventas, no se tomaron en cuenta las ventas con devolución, ya que, el producto al ser devuelto, se regresa el dinero, no concretando la venta como se esperaba en un principio.

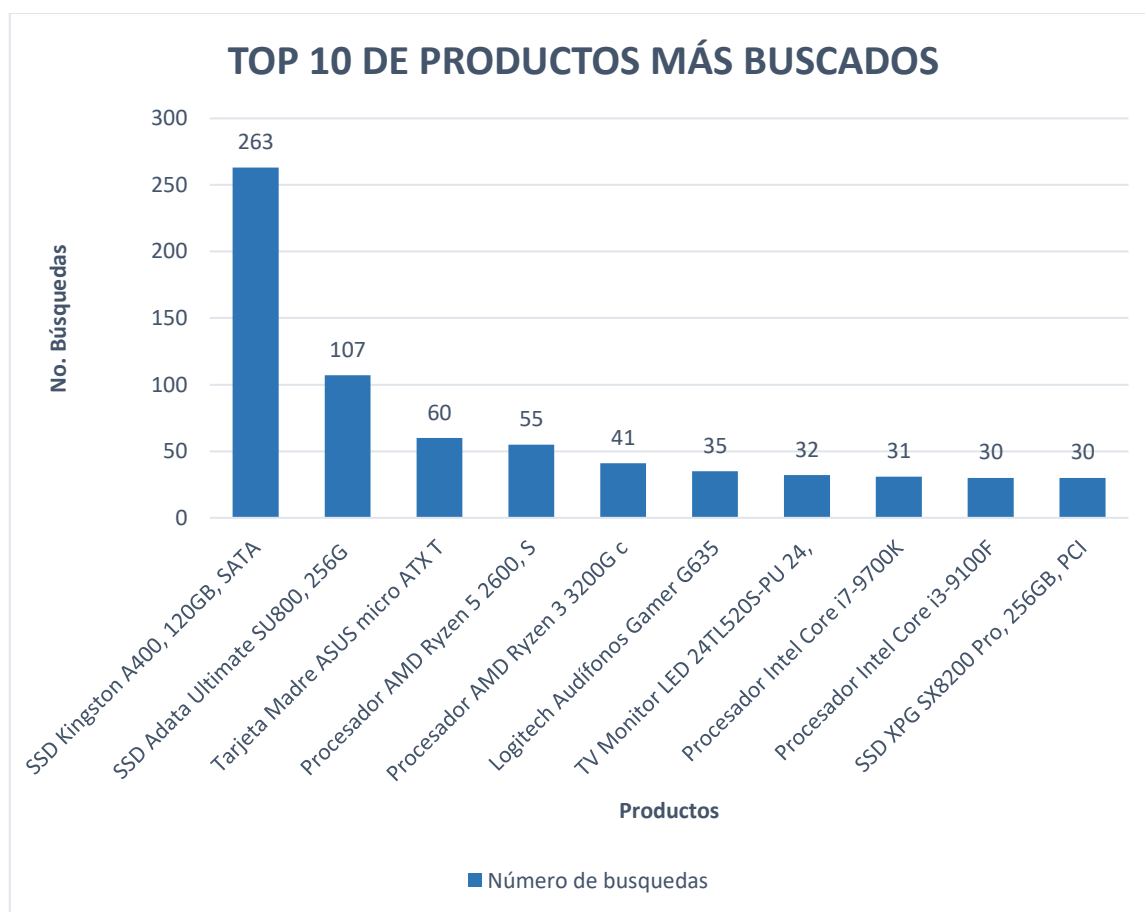
LISTA GENERAL DE 5 PRODUCTOS CON MAYORES VENTAS			
ID	Nombre del Producto	Unidades Vendidas (u)	Ganancia Total (\$)
54	SSD Kingston A400, 120GB, SATA	49	12691
3	Procesador AMD Ryzen 5 2600, S	42	129738
5	Procesador Intel Core i3-9100F	20	35580
42	Tarjeta Madre ASRock Micro ATX	18	32022
57	SSD Adata Ultimate SU800, 256G	15	13335



En las gráficas anteriores es posible observar que el producto más vendido es la SSD Kingston A400, 120 GB, SATA, sin embargo, no por ser la más vendida genera la mayor ganancia. Tal es el caso del Procesador AMD Ryzen 5 2600, ya que tiene menor cantidad de ventas, pero un ingreso mucho mayor.

## 2. Los 10 productos más buscados

LISTA GENERAL DE 10 PRODUCTOS CON MAYORES BUSQUEDAS		
[ID]	Nombre del Producto	Busquedas Totales
54	SSD Kingston A400, 120GB, SATA	263
57	SSD Adata Ultimate SU800, 256G	107
29	Tarjeta Madre ASUS micro ATX T	60
3	Procesador AMD Ryzen 5 2600, S	55
4	Procesador AMD Ryzen 3 3200G c	41
85	Logitech Audífonos Gamer G635	35
67	TV Monitor LED 24TL520S-PU 24,	32
7	Procesador Intel Core i7-9700K	31
5	Procesador Intel Core i3-9100F	30
47	SSD XPG SX8200 Pro, 256GB, PCI	30



En la gráfica anterior es posible observar que el producto más buscado es la SSD Kingston A400, 120GB, SATA lo cual hace mucha coherencia, ya que se trata del producto más vendido de LifeStore. Posteriormente, el producto más buscado es la SSD Adata Ultimate SU800 256 GB, sin embargo, este corresponde al 5 producto más vendido, teniendo ventas mucho menores que el primero, pero con ingresos muy cercanos.

### 3. Productos (por categoría) con menores ventas

LISTA DE PRODUCTOS CON MENORES VENTAS POR CATEGORÍA			
[ID]	Nombre del Producto	Unidades Vendidas (u)	Ganancia Total (\$)
Categoría: PROCESADORES			
9	Procesador Intel Core i3-8100,	0	0
1	Procesador AMD Ryzen 3 3300X S	2	6038
6	Procesador Intel Core i9-9900K	3	35427
8	Procesador Intel Core i5-9600K	4	21596
7	Procesador Intel Core i7-9700K	7	59913
Categoría: TARJETAS DE VIDEO			
14	Tarjeta de Video EVGA NVIDIA G	0	0
15	Tarjeta de Video EVGA NVIDIA G	0	0
16	Tarjeta de Video EVGA NVIDIA G	0	0
17	Tarjeta de Video Gigabyte AMD	0	0
19	Tarjeta de Video Gigabyte NVID	0	0
Categoría: TARJETAS MADRE			
30	Tarjeta Madre AORUS ATX Z390 E	0	0
32	Tarjeta Madre ASRock Z390 Phan	0	0
34	Tarjeta Madre ASUS ATX ROG STR	0	0
35	Tarjeta Madre Gigabyte micro A	0	0
36	Tarjeta Madre Gigabyte micro A	0	0
Categoría: DISCOS DUROS			
53	SSD Addlink Technology S70, 51	0	0
55	SSD para Servidor Supermicro S	0	0
56	SSD para Servidor Lenovo Think	0	0
58	SSD para Servidor Lenovo Think	0	0
59	SSD Samsung 860 EVO, 1TB, SATA	0	0
Categoría: MEMORIAS USB			
61	Kit Memoria RAM Corsair Vengea	0	0
60	Kit Memoria RAM Corsair Domina	1	2519
Categoría: PANTALLAS			
62	Makena Smart TV LED 32S2 32'',	0	0
63	Seiki TV LED SC-39HS950N 38.5,	0	0
64	Samsung TV LED LH43QMREBGCXG0	0	0
65	Samsung Smart TV LED UN70RU710	0	0
68	Makena Smart TV LED 40S2 40'',	0	0
Categoría: BOCINAS			
75	Lenovo Barra de Sonido, Alámbr	0	0
76	Acteck Bocina con Subwoofer AX	0	0
77	Verbatim Bocina Portátil Mini,	0	0
78	Ghia Bocina Portátil BX300, Bl	0	0
79	Naceb Bocina Portátil NA-0301,	0	0
Categoría: AUDIFONOS			
86	ASUS Audífonos Gamer ROG Theta	0	0
87	Acer Audífonos Gamer Galea 300	0	0
88	Audífonos Gamer Balam Rush Orp	0	0
90	Energy Sistem Audífonos con Mi	0	0
91	Genius GHP-400S Audífonos, Alá	0	0

En la tabla anterior, es posible observar los 5 productos con menores ventas por cada categoría. Hay una cantidad inmensa de productos sin ninguna venta y por ende ningún ingreso. Esto genera un problema, ya que el stock de cada producto ha estado intacto durante un año aproximadamente.

#### 4. Productos (por categoría) con menores búsquedas

LISTA DE PRODUCTOS CON MENORES BÚSQUEDAS POR CATEGORÍA		
ID	Nombre del Producto	Busquedas Totales
Categoría: PROCESADORES		
9	Procesador Intel Core i3-8100,	1
1	Procesador AMD Ryzen 3 3300X S	10
6	Procesador Intel Core i9-9900K	10
8	Procesador Intel Core i5-9600K	20
2	Procesador AMD Ryzen 5 3600, S	24
5	Procesador Intel Core i3-9100F	30
7	Procesador Intel Core i7-9700K	31
4	Procesador AMD Ryzen 3 3200G c	41
3	Procesador AMD Ryzen 5 2600, S	55
Categoría: TARJETAS DE VIDEO		
14	Tarjeta de Video EVGA NVIDIA G	0
16	Tarjeta de Video EVGA NVIDIA G	0
19	Tarjeta de Video Gigabyte NVID	0
20	Tarjeta de Video Gigabyte NVID	0
23	Tarjeta de Video MSI Radeon X1	0
24	Tarjeta de Video PNY NVIDIA Ge	0
10	MSI GeForce 210, 1GB GDDR3, DV	1
27	Tarjeta de Video VisionTek AMD	1
13	Tarjeta de Video Asus NVIDIA G	2
17	Tarjeta de Video Gigabyte AMD	3
Categoría: TARJETAS MADRE		
30	Tarjeta Madre AORUS ATX Z390 E	0
32	Tarjeta Madre ASRock Z390 Phan	0
33	Tarjeta Madre ASUS ATX PRIME Z	0
34	Tarjeta Madre ASUS ATX ROG STR	0
36	Tarjeta Madre Gigabyte micro A	0
37	Tarjeta Madre ASRock ATX Z490	0
38	Tarjeta Madre Gigabyte Micro A	0
41	Tarjeta Madre ASUS micro ATX P	0
43	Tarjeta Madre ASUS ATX ROG STR	0
35	Tarjeta Madre Gigabyte micro A	1
Categoría: DISCOS DUROS		
53	SSD Addlink Technology S70, 51	0
55	SSD para Servidor Supermicro S	0
58	SSD para Servidor Lenovo Think	0
59	SSD Samsung 860 EVO, 1TB, SATA	1
56	SSD para Servidor Lenovo Think	2
52	SSD Western Digital WD Blue 3D	5
50	SSD Crucial MX500, 1TB, SATA I	7
49	Kit SSD Kingston KC600, 1TB, S	10
51	SSD Kingston UV500, 480GB, SAT	11
48	SSD Kingston A2000 NVMe, 1TB,	27



Categoría: MEMORIAS USB

60	Kit Memoria RAM Corsair Domina	0
61	Kit Memoria RAM Corsair Vengea	0

Categoría: PANTALLAS

62	Makena Smart TV LED 32S2 32'',	0
64	Samsung TV LED LH43QMREBGCXG0	0
65	Samsung Smart TV LED UN70RU710	0
68	Makena Smart TV LED 40S2 40'',	0
69	Hisense Smart TV LED 40H5500F	0
71	Samsung Smart TV LED UN32J4290	0
72	Hisense Smart TV LED 50H8F 49.	0
70	Samsung Smart TV LED 43, Full	1
63	Seiki TV LED SC-39HS950N 38.5,	4
73	Samsung Smart TV LED UN55TU700	4

Categoría: BOCINAS

75	Lenovo Barra de Sonido, Alámbro	0
77	Verbatim Bocina Portátil Mini,	0
78	Ghia Bocina Portátil BX300, Bl	0
79	Naceb Bocina Portátil NA-0301,	0
81	Ghia Bocina Portátil BX900, Bl	0
82	Ghia Bocina Portátil BX400, Bl	0
83	Ghia Bocina Portátil BX500, Bl	0
80	Ghia Bocina Portátil BX800, Bl	1
76	Acteck Bocina con Subwoofer AX	2
74	Logitech Bocinas para Computad	6

Categoría: AUDIFONOS

86	ASUS Audífonos Gamer ROG Theta	0
87	Acer Audífonos Gamer Galea 300	0
88	Audífonos Gamer Balam Rush Orp	0
90	Energy Sistem Audífonos con Mi	0
92	Getttech Audífonos con Micrófo	0
96	Klip Xtreme Audífonos Blast, B	0
93	Ginga Audífonos con Micrófono	1
91	Genius GHP-400S Audífonos, Alá	2
95	Iogear Audífonos Gamer GHG601,	3
94	HyperX Audífonos Gamer Cloud F	6

En la tabla anterior, es posible observar que existen demasiados productos en cada categoría que nunca se han buscado. Es importante verificar que sea un problema de desinterés por el producto y no por desconocimiento de su existencia en la empresa. Es un problema muy grande, ya que, al no existir ninguna búsqueda, quiere decir que tampoco hay venta, por ende, el stock está intacto. Los procesadores son los productos que tienen mayor número de ventas, seguido por los discos duros.



## 5. Mejores y Peores reseñas por producto

LISTA DE LOS 5 PRODUCTOS CON MEJORES RESEÑAS				
[ID]	Nombre del Producto	Reseña Promedio	Unidades Vendidas (u)	Ganancia Total (\$)
6	Procesador Intel Core i9-9900K	5.0	3	35427
7	Procesador Intel Core i7-9700K	5.0	7	59913
8	Procesador Intel Core i5-9600K	5.0	4	21596
11	Tarjeta de Video ASUS AMD Rade	5.0	3	22197
49	Kit SSD Kingston KC600, 1TB, S	5.0	3	9417

LISTA DE LOS 5 PRODUCTOS CON PEORES RESEÑAS				
[ID]	Nombre del Producto	Reseña Promedio	Unidades Vendidas (u)	Ganancia Total (\$)
17	Tarjeta de Video Gigabyte AMD	1.0	1	4199
45	Tarjeta Madre ASRock ATX H110	1.0	1	2869
31	Tarjeta Madre AORUS micro ATX	1.83	6	13374
46	Tarjeta Madre Gigabyte micro A	2.0	1	1539
89	Cougar Audífonos Gamer Phontum	3.0	1	859

En la tabla anterior es posible percatarse que existen productos con reseñas altas. La cantidad de ventas es pequeña porque se tratan de productos con un precio elevado. Para la filtración se tomaron en cuenta los productos con mas de dos unidades vendidas, ya que así sería posible cerciorarse que las reseñas son confiables para los productos con mayores ventas.

Además, se puede observar que existen productos con reseña muy baja y con pocas unidades vendidas. Esto genera un problema ya que genera desconfianza al cliente sobre si el producto en cuestión funciona o presenta problemas técnicos.

## 6. Información financiera mensual y anual de LifeStore

Para obtener las ventas por mes, no se tomaron en cuenta las ventas con devolución ya que no se concretó al 100% la venta. El ingreso total anual de LifeStore fue de: **\$737,916.00**

TOTAL DE INGRESOS Y VENTAS PROMEDIO MENSUALES			
No.Mes	Mes	Ventas(u)	Ingresos(\$)
01	- January -	52	117738
02	- February -	40	107270
03	- March -	49	162931
04	- April -	74	191066
05	- May -	34	91936
06	- June -	11	36949
07	- July -	11	26949
08	- August -	3	3077
09	- September -	0	0
10	- October -	0	0
11	- November -	0	0
12	- December -	0	0

5 MESES CON MÁS VENTAS AL AÑO			
No.Mes	Mes	Ventas(u)	Ingresos(\$)
04	- April -	74	191066
01	- January -	52	117738
03	- March -	49	162931
02	- February -	40	107270
05	- May -	34	91936

El Total Anual de LifeStore es: \$ 737916

En la tabla anterior, se presenta la cantidad de productos vendidos en su respectivo mes, así como los ingresos que generaron. Es preocupante que durante el ultimo trimestre del año no existan ventas y por ende ingresos.

Tomando en cuenta la cantidad de unidades vendidas y todos los meses, el promedio de ventas mensual es de aproximadamente 23 productos, en otras palabras, idealmente se venden 23 productos al mes. Si la situación de la empresa continua de la misma manera podría caer en quiebra en cualquier momento.



El primer trimestre del año es donde se generaron mas ventas y mas ingresos, siendo abril el mejor mes. En el segundo trimestre las ventas comenzaron a caer considerablemente hasta el último trimestre donde no se vendió nada.

## **Solución al Problema**

---

El principal problema que presenta LifeStore, es la gran cantidad de productos en Stock, generando consecuencias como: problemas de flujo de efectivo, reducción de márgenes, es decir, terminar ofreciendo los productos a un menor precio, también se presentan gastos adicionales para el almacenamiento del sobre stock y posteriormente gastos adicionales para eliminarlos.

Para solucionar el problema, se propone lo siguiente:

- Difundir la empresa en redes sociales como Facebook, Instagram, Twitter, propiciando que más personas conozcan quien es LifeStore y que ofrece.
- Difundir cada uno de los productos en diferentes redes sociales. Es importante difundir los productos más vendidos, así como los productos menos vendidos, buscando que las personas se enteren que en LifeStore pueden adquirirlos.
- Verificar con los proveedores que los productos se encuentren en buenas condiciones y funcionen correctamente, mejorando la experiencia de quien los consume.
- Mostrar de primera instancia en la pagina web los productos que nunca se han buscado, esperando que esta situación se trate por el desconocimiento del cliente sobre si se vendía en LifeStore o no.
- Poner algún descuento a cada uno de los productos rezagados que se encuentran en inventario, buscando sacarlos lo más pronto posible.
- Adquirir más productos de los más vendidos, como lo son los procesadores, sin embargo, primero teniendo un estudio de mercado que avale que se venderán. Esto propiciaría más ingresos para ayudar a solventar el sobre stock.
- Dejar de adquirir productos que nunca se han buscado y por ende vendido. Al tener acceso a los intereses de las personas por medio de la página web, es importante tomar en consideración ya no vender productos como lo son algunas tarjetas de video, tarjetas madre, pantallas, memorias USB.
- Armar algunos paquetes de promoción en donde se puedan juntar algunos productos más vendidos y productos rezagados, propiciando la venta del sobre stock.

## Conclusión

---

Es importante para LifeStore que se hagan estudios de mercado sobre los productos que ofrece antes de adquirirlos. También podría convertirse en una empresa dedicada solo a la venta de procesadores y discos duros, que son los productos que más se venden.

Para que una empresa tenga éxito, es de suma importancia contar con un plan de negocios que le permita saber sus limitaciones, por lo que, para LifeStore es necesario reconsiderar su plan actual, ya que, en cualquier momento, podría caer en quiebra.

Utilizar los datos de búsqueda de las personas ayuda a saber sus intereses, poniendo a la empresa en ventaja para que adquieran con ella, sin embargo, es de suma importancia no sobre adquirir, ya que podría suceder lo mismo que con LifeStore.

## Anexos

---

### 1. Login de LifeStore

```
LIFESTORE - LOGIN
Ingrese su usuario: fgdsgsd
Ingrese su contraseña: gfyf
El usuario y/o contraseña son incorrectos
Tienes 1 intentos más
```

```
BIENVENIDE A LIFESTORE DANIELMORAAN

1. Lista de los productos con mayores ventas
2. Lista de los productos con mayores búsquedas
3. Lista (por categoría) de los productos con menores ventas
4. Lista (por categoría) de los productos con menores búsquedas
5. Lista de los productos por reseña
6. Información financiera mensual y anual de LifeStore
7. Salir

Selecciona una opción: █
```

## 2. Código empleado

```

3. """
4. This is the LifeStore_SalesList data:
5.
6. lifestore_searches = [id_search, id_product]
7. lifestore_sales = [id_sale, id_product, score (from 1 to 5), date,
   refund (1 for true or 0 to false)]
8. lifestore_products = [id_product, name, price, category, stock]
9. """
10.
11.     from lifestore_file import lifestore_products,
   lifestore_sales, lifestore_searches
12.     from os import system
13.     from time import sleep
14.     import datetime
15.
16.     '''-----LISTA DE 5 PRODUCTOS CON MAYORES VENTAS-----'''
17.     def top_sales_product():
18.         #Diccionario con los id por producto
19.         product_id = {}
20.         for product in lifestore_products:
21.             id_product = product[0]
22.             name = product[1]
23.             if name not in product_id.keys():
24.                 product_id[name] = []
25.                 product_id[name].append(id_product)
26.
27.         #Diccionario con las ventas(unidades) por id
28.         product_sales = {}
29.         for sales in lifestore_sales:
30.             id_product_sales = sales[1]
31.             refund = sales[4]
32.             if id_product_sales not in product_sales.keys():
33.                 product_sales[id_product_sales] = []
34.                 product_sales[id_product_sales].append(refund)
35.                 #Si la venta tuvo devolución, entonces no se toma en
   cuenta
36.                 if refund != 0:
37.                     product_sales[id_product_sales].remove(1)
38.
39.         #Diccionario con el total de ventas(unidades) por cada
   producto
40.         #Asignación de ventas y ganancias a cada producto por
   medio del ID.
41.         product_results = {}
42.         for name, list_id in product_id.items():
43.             total_sales = 0
44.             total_profit = 0
45.             for id_product_sales in list_id:
46.                 if id_product_sales not in product_sales.keys():
47.                     continue
48.                 no_refund_sales = product_sales[id_product_sales]
49.                 price = lifestore_products[id_product_sales-1][2]
50.                 sales_amount = len(no_refund_sales)
51.
52.                 profit_amount = price*sales_amount
53.

```

```

54.         total_sales += sales_amount
55.         total_profit += profit_amount
56.
57.         #Si se imprime aquí dentro solo serían los
vendidos
58.
59.         #Si se imprime afuera serían todos los productos
60.         product_results[name] = {
61.             'ID': id_product_sales,
62.             'Total Sales': total_sales,
63.             'Total Profit': total_profit }
64.
65.         #Ordenando el diccionario resultante
66.         product_results_ord = sorted(product_results.items(), key
= lambda x: x[1]['Total Sales'], reverse= True)
67.         print(f'\n\t\tLISTA DE LOS 5 PRODUCTOS CON MAYORES
VENTAS')
68.         print(f'|ID| \t\t Nombre del Producto \t Unidades
Vendidas (u) \t Ganancia Total ($) ')
69.
70.         #Imprimiendo solo 5 productos
71.         for product, attributes in product_results_ord[0:5]:
#Lista
72.             print(f'{attributes["ID"]} \t {product[0:30]} \t
{attributes["Total Sales"]} \t\t\t {attributes["Total Profit"]} ')
73.
74.
75.         '''-----LISTA DE 10 PRODUCTOS CON MAYORES BÚSQUEDAS-----'''
76.         def top_searches_product():
77.             #Diccionario con el id por producto
78.             product_id = {}
79.             for product in lifestore_products:
80.                 id_product = product[0]
81.                 name = product[1]
82.                 if name not in product_id.keys():
83.                     product_id[name] = []
84.                     product_id[name].append(id_product)
85.
86.             #Diccionario con las busquedas(cantidad) por id
87.             product_searches = {}
88.             for searches in lifestore_searches:
89.                 id_product_searches = searches[1]
90.                 search = searches[0]
91.                 if id_product_searches not in
product_searches.keys():
92.                     product_searches[id_product_searches] = []
93.                     product_searches[id_product_searches].append(search)
94.
95.             #Diccionario con el total de busquedas por cada producto
96.             #Asignación de busquedas a cada producto por medio del ID
97.             searches_results = {}
98.             for name, list_id in product_id.items():
99.                 total_searches = 0
100.                 for id_product_searches in list_id:
101.                     if id_product_searches not in
product_searches.keys():
102.                         continue

```

```

103.         search_list_lifestore =
        product_searches[id_product_searches]
104.         search_amount = len(search_list_lifestore)
105.         total_searches += search_amount
106.
107.         searches_results[name] = {
108.             'ID': id_product_searches,
109.             'Total Searches': total_searches,
110.         }
111.
112.         #Ordenando el diccionario de búsquedas resultante
113.         searched_results_ord = sorted(searches_results.items(),
        key = lambda x: x[1]['Total Searches'], reverse= True)
114.         print(f'\n\tLISTA DE LOS 10 PRODUCTOS CON MAYORES
        BUSQUEDAS')
115.         print(f'|ID| \t\t Nombre del Producto \t\t Búsquedas
        Totales ')
116.
117.         #Imprimiendo solo 10 productos
118.         for product, attributes in searched_results_ord[0:10]:
        #Lista
119.             print(f'{attributes["ID"]} \t {product[0:30]} \t\t
        {attributes["Total Searches"]}')
120.
121.
122.         '''-----LISTA DE PRODUCTOS CON MENORES VENTAS POR CATEGORÍA--
        -----'''
123.         def category_bottom_sales():
124.             #Diccionario con los nombres por categoría
125.             category_product = {}
126.             for product in lifestore_products:
127.                 name = product[1]
128.                 category = product[3]
129.                 if category not in category_product.keys():
130.                     category_product[category] = []
131.                     category_product[category].append(name)
132.
133.             #Diccionario con los id por producto
134.             product_id = {}
135.             for product in lifestore_products:
136.                 id_product = product[0]
137.                 name = product[1]
138.                 if name not in product_id.keys():
139.                     product_id[name] = []
140.                     product_id[name].append(id_product)
141.
142.             #Diccionario con las ventas(unidades) por id
143.             #En este diccionario se eliminan las ventas con
        devolución
144.             product_sales = {}
145.             for sales in lifestore_sales:
146.                 id_product_sales = sales[1]
147.                 refund = sales[4]
148.                 if id_product_sales not in product_sales.keys():
149.                     product_sales[id_product_sales] = []
150.                     product_sales[id_product_sales].append(refund)
151.                 if refund != 0:

```



```

152.         product_sales[id_product_sales].remove(1)
153.
154.         #Diccionario con el total de ventas(unidades) por cada
        producto
155.         #Asignación de ventas y ganancias a cada producto por
        medio del ID.
156.         #Name: Id ----> Id : Sales
157.         product_results = {}
158.         for name, list_id in product_id.items():
159.             total_sales = 0
160.             total_profit = 0
161.             for id_product_sales in list_id:
162.                 if id_product_sales not in product_sales.keys():
163.                     continue
164.                 no_refund_sales = product_sales[id_product_sales]
165.                 price = lifestore_products[id_product_sales-1][2]
166.                 sales_amount = len(no_refund_sales)
167.
168.                 profit_amount = price*sales_amount
169.
170.                 total_sales += sales_amount
171.                 total_profit += profit_amount
172.
173.                 product_results[name] = {
174.                     'ID': id_product_sales,
175.                     'Total Sales': total_sales,
176.                     'Total Profit': total_profit }
177.
178.         #Diccionario con el total de productos por cada categoría
179.         #División de productos por categoría por medio del
        nombre,
180.         # Category:Name ----> Name:Id
181.         category_results = {}
182.         for cat, list_name in category_product.items():
183.             names_product = []
184.             for name in list_name:
185.                 if name not in product_results.keys():
186.                     continue
187.
188.             names_product.append([name[0:30],product_results[name]])
189.
190.             category_results[cat] = {
191.                 'Name' : names_product }
192.
193.         #Indagando en el diccionario para imprimir la categoría y
194.         #los atributos de cada producto (ID, Ventas y Ganancias)
195.         print(f'\n\t\t\t\tLISTA DE 5 PRODUCTOS CON MENORES VENTAS
        POR CATEGORÍA')
196.         print(f'|ID| \t\t\t Nombre del Producto \t\t Unidades
        Vendidas (u) \t\t\t Ganancia Total ($) ')
197.         for categories, attributes in category_results.items():
198.             print(f'\nCategoría: {categories.upper()}')
199.             for key, list_product in attributes.items():
200.                 #Imprimiendo solo 5 productos
201.                 list_product = sorted(list_product, key = lambda
        x: x[1]['Total Sales'], reverse= False)
        for product_name, attr in list_product[0:5]:

```

```

202.             #if attr["Total Sales"] != 0:
203.                 print(f'{attr["ID"]} \t
    {product_name} \t\t {attr["Total Sales"]} \t\t\t\t {attr["Total
    Profit"]}')
204.
205.
206.     '''-----LISTA DE PRODUCTOS CON MENORES BÚSQUEDAS POR
    CATEGORÍA-----'''
207.     def category_bottom_searches():
208.         #Diccionario con los nombres por categoría
209.         category_product = {}
210.         for product in lifestore_products:
211.             name = product[1]
212.             category = product[3]
213.             if category not in category_product.keys():
214.                 category_product[category] = []
215.                 category_product[category].append(name)
216.
217.         #Diccionario con los id por producto
218.         product_id = {}
219.         for product in lifestore_products:
220.             id_product = product[0]
221.             name = product[1]
222.             if name not in product_id.keys():
223.                 product_id[name] = []
224.                 product_id[name].append(id_product)
225.
226.         #Diccionario con las ventas(unidades) por id
227.         #En este diccionario se eliminan las ventas con
    devolución
228.         product_searches = {}
229.         for searches in lifestore_searches:
230.             id_product_searches = searches[1]
231.             search = searches[0]
232.             if id_product_searches not in
    product_searches.keys():
233.                 product_searches[id_product_searches] = []
234.                 product_searches[id_product_searches].append(search)
235.
236.         #Diccionario con el total de ventas(unidades) por cada
    producto
237.         #Asignación de ventas y ganancias a cada producto por
    medio del ID
238.         #Name: Id ----> Id : Sales
239.         searches_results = {}
240.         for name, list_id in product_id.items():
241.             total_searches = 0
242.             for id_product_searches in list_id:
243.                 if id_product_searches not in
    product_searches.keys():
244.                     continue
245.                 search_list_lifestore =
    product_searches[id_product_searches]
246.                 search_amount = len(search_list_lifestore)
247.                 total_searches += search_amount
248.
249.                 searches_results[name] = {

```

```

250.         'ID': id_product_searches,
251.         'Total Searches': total_searches,
252.     }
253.
254.     #Diccionario con el total de productos por cada categoría
255.     #División de productos por categoría por medio del
    nombre,
256.     # Category:Name ----> Name:Id
257.     category_results = {}
258.     for cat, list_name in category_product.items():
259.         names_product = []
260.         for name in list_name:
261.             if name not in searches_results.keys():
262.                 continue
263.
264.         names_product.append([name[0:30],searches_results[name]])
265.
266.         category_results[cat] = {
267.             'Name' : names_product }
268.
269.     #Indagando en el diccionario para imprimir la categoría y
270.     #los atributos de cada producto (ID y Búsquedas)
271.     print(f'\n\tLISTA DE 10 PRODUCTOS CON MENORES BÚSQUEDAS
    POR CATEGORÍA')
272.     print(f'|ID| \t\t Nombre del Producto \t\t Busquedas
    Totales ')
273.     for categories, attributes in category_results.items():
274.         print(f'\nCategoría: {categories.upper()}')
275.         for key, list_product in attributes.items():
276.             #Imprimiendo solo 10 productos
277.             list_product = sorted(list_product, key =
    lambda x: x[1]['Total Searches'], reverse= False)
278.             for product_name, attr in list_product[0:10]:
279.                 print(f'{attr["ID"]} \t {product_name}
    \t\t {attr["Total Searches"]}')
280.
281.     '''----- LISTAS DE LAS MEJORES Y PEORES RESEÑAS -----'''
282.     def product_review():
283.         #Diccionario con los id por producto
284.         product_id = {}
285.         for product in lifestore_products:
286.             id_product = product[0]
287.             name = product[1]
288.             if name not in product_id.keys():
289.                 product_id[name] = []
290.                 product_id[name].append(id_product)
291.
292.         #Diccionario con las reseñas por id
293.         product_review = {}
294.         for reviews in lifestore_sales:
295.             id_product_review = reviews[1]
296.             review = reviews[2]
297.             if id_product_review not in product_review.keys():
298.                 product_review[id_product_review] = []
299.                 product_review[id_product_review].append(review)
300.

```

```

301.         #Diccionario con el total de reviews por cada producto
302.         #Asignación de ventas, ganancias y review promedio a cada
        producto por medio del ID
303.         review_results = {}
304.         for name, list_id in product_id.items():
305.             total_review = []
306.             total_sales = 0
307.             total_profit = 0
308.             for id_product_review in list_id:
309.                 if id_product_review not in
        product_review.keys():
310.                     continue
311.                     no_refund_sales =
        product_review[id_product_review]
312.                     price = lifestore_products[id_product_review-
        1][2]
313.                     sales_amount = len(no_refund_sales)
314.
315.                     profit_amount = price*sales_amount
316.
317.                     total_sales += sales_amount
318.                     total_profit += profit_amount
319.
320.                     no_review_sales =
        product_review[id_product_review]
321.                     total_review += no_review_sales
322.                     #Si se imprime aquí dentro solo serían los
        vendidos
323.
324.                     #Calculando la review promedio por producto
325.                     prom = sum(total_review)/len(total_review)
326.                     prom_round = round(prom,2)
327.
328.                     #Si se imprime afuera serían todos los productos
329.                     review_results[name] = {
330.                         'ID': id_product_review,
331.                         'Total Sales': total_sales,
332.                         'Total Profit': total_profit,
333.                         'Total Review': prom_round }
334.
335.         ##LISTAS MEJORES Y PEORES REVIEWS
336.         '''-----'''
337.
338.         #Ordenando el diccionario de MEJOR REVIEW a PEOR REVIEW,
        sin tomar en cuenta la cantidad de ventas
339.         product_results_ord = sorted(review_results.items(), key
        = lambda x: x[1]['Total Review'], reverse= True)
340.         print(f'\n\t\t\t\tLISTA DE LOS 5 PRODUCTOS CON MEJORES
        RESEÑAS ')
341.         print(f'|ID| \t\t Nombre del Producto \t Reseña Promedio
        \t Unidades Vendidas (u) \t Ganancia Total ($)')
342.
343.         #Imprimiendo solo 5 productos
344.         #Se propone que el producto tenga mas de dos ventas para
        que se considere su reseña promedio

```

```

345.         for product, attributes in product_results_ord[0:20]:
346.             #Lista
347.             if attributes['Total Sales'] != 0 and
348.                 attributes['Total Sales'] > 2 :
349.                 print(f'{attributes["ID"]} \t {product[0:30]} \t
350.                     {attributes["Total Review"]} \t\t\t {attributes["Total Sales"]}
351.                     \t\t\t {attributes["Total Profit"]} ')
352.             #Ordenando el diccionario de PEOR REVIEW a MEJOR REVIEW,
353.             sin tomar en cuenta la cantidad de ventas
354.             product_results_ord = sorted(review_results.items(), key
355.                 = lambda x: x[1]['Total Review'], reverse= False)
356.             print(f'\n\t\t\t\tLISTA DE LOS 5 PRODUCTOS CON PEORES
357.                 RESEÑAS ')
358.             print(f'|ID| \t\t Nombre del Producto \t Reseña Promedio
359.                 \t Unidades Vendidas (u) \t Ganancia Total ($) ')
360.             #Imprimiendo solo 5 productos
361.             for product, attributes in product_results_ord[0:6]:
362.                 #Lista
363.                 if attributes['Total Sales'] != 0:
364.                     print(f'{attributes["ID"]} \t {product[0:30]} \t
365.                         {attributes["Total Review"]} \t\t\t {attributes["Total Sales"]}
366.                         \t\t\t {attributes["Total Profit"]} ')
367.
368.
369.
370.
371.
372.
373.
374.
375.
376.
377.
378.
379.
380.
381.
382.
383.
384.
385.
386.
387.
388.

```

```

def inf_sales_LifeStore():
    #Creación de diccionario donde se guarda la cantidad
    #de ventas por cada mes.
    month_sales = {}
    mon_calendar = [datetime.date(2000, m, 1).strftime('%m')
    for m in range(1, 13)]
    for product in lifestore_sales:
        id_sales = product[0]
        refund = product[4]
        _, mon, year = product[3].split('/')
        if mon not in month_sales.keys():
            month_sales[mon] = []
            month_sales[mon].append(id_sales)

        for mons in mon_calendar:
            if mons not in month_sales.keys():
                month_sales[mons] = []

        #Si hay una devolución, la venta no se concreta
        #por lo que no se contaría dentro del total
        if refund != 0:
            month_sales[mon].remove(id_sales)

    #Diccionario con información de cada uno de los meses
    #Cálculo del total de ventas y el total de ganancia por
    cada MES
    info_month = {}
    for mon in month_sales.keys():
        month_list = month_sales[mon]
        total_profit = 0
        total_sales = 0

```

```

389.         for id_sales in month_list:
390.             id_product = lifestore_sales[id_sales-1][1]
391.             price = lifestore_products[id_product-1][2]
392.
393.             total_profit += price
394.             total_sales = len(month_list)
395.
396.             #Se sustituye el número del mes por
397.             #su respectivo nombre, para mejor presentación
398.             if mon == '01':
399.                 mons = '- January -'
400.             elif mon == '02':
401.                 mons = '- February -'
402.             elif mon == '03':
403.                 mons = '- March -'
404.             elif mon == '04':
405.                 mons = '- April -'
406.             elif mon == '05':
407.                 mons = '- May -'
408.             elif mon == '06':
409.                 mons = '- June -'
410.             elif mon == '07':
411.                 mons = '- July -'
412.             elif mon == '08':
413.                 mons = '- August -'
414.             elif mon == '09':
415.                 mons = '- September -'
416.             elif mon == '10':
417.                 mons = '- October -'
418.             elif mon == '11':
419.                 mons = '- November -'
420.             else:
421.                 mons = '- December -'
422.
423.             info_month[mons] = {
424.                 'No': mon,
425.                 'TSales': total_sales,
426.                 'TProfit': total_profit}
427.
428.
429.             #Ordenando el diccionario de información por cada mes
430.             info_month_new = sorted(info_month.items(), key = lambda
x: x[1]['No'], reverse= False)
431.             print(f'\nTOTAL DE INGRESOS Y VENTAS PROMEDIO MENSUALES')
432.             print(f'\nNo.Mes \t Mes \t\t Ventas(u) \t Ingresos($)')
433.
434.             #Imprimiendo la información del ingreso y ventas por cada
mes
435.             total_anual_profit = 0
436.             for mons, attributes in info_month_new: #Lista
437.                 print(f'{attributes["No"]} \t {mons} \t
{attributes["TSales"]} \t\t {attributes["TProfit"]}')
438.                 total_anual_profit += attributes["TProfit"]
439.
440.             #Ordenando el diccionario de información por mes acorde a
la cantidad de ventas

```

```

441.         info_month_new = sorted(info_month.items(), key = lambda
x: x[1]['TSales'], reverse= True)
442.         print(f'\n\t5 MESES CON MÁS VENTAS AL AÑO')
443.         print(f'No.Mes \t Mes \t\t Ventas(u) \t Ingresos($)')
444.
445.         #Imprimiendo la información de los meses con más ventas
en LifeStore
446.         for mons, attributes in info_month_new[0:5]: #Lista
447.             print(f'{attributes["No"]} \t {mons} \t
{attributes["TSales"]} \t\t {attributes["TProfit"]}')
448.
449.             print(f'\nEl Total Anual de LifeStore es: $',
total_anual_profit)
450.
451.
452.         '''-----LOGIN-----
-----'''
453.         users = {'Daniel Morán' : 'danielmoraan' , 'Jaime Alonso' :
'jimmyalonso'}
454.         passwords = {'Daniel Morán' : 'Pyth0n3mTech' , 'Jaime Alonso'
: '3mTechD4t4'}
455.
456.         intentos = 4
457.         while intentos > 0:
458.             print("\t\tLIFESTORE - LOGIN")
459.             user = input('Ingrese su usuario: ')
460.             password = input('Ingrese su contraseña: ')
461.
462.             #Al ingresar con el usuario, borrará el contenido de la
terminal
463.             #y mostrará el menú principal
464.             if user == users['Daniel Morán'] and password ==
passwords['Daniel Morán']:
465.                 sleep(2)
466.                 system("cls")
467.                 print('\tBIENVENIDE A LIFESTORE ' +
str.upper(users['Daniel Morán']))
468.
469.                 print('\n1. Lista de los productos con mayores
ventas')
470.                 print('2. Lista de los productos con mayores
búsquedas')
471.                 print('3. Lista (por categoría) de los productos con
menores ventas')
472.                 print('4. Lista (por categoría) de los productos con
menores búsquedas')
473.                 print('5. Lista de los productos por reseña')
474.                 print('6. Información financiera mensual y anual de
LifeStore')
475.                 print('7. Salir')
476.
477.                 ans = int(input('\nSelecciona una opción: '))
478.
479.                 #Al ingresar a cualquier opción, borrará el contenido
de la terminal
480.                 #y mostrará lo solicitado
481.                 if ans == 1:

```



```

482.         sleep(2)
483.         system("cls")
484.         top_sales_product()
485.
486.     elif ans == 2:
487.         sleep(2)
488.         system("cls")
489.         top_searches_product()
490.
491.     elif ans == 3:
492.         sleep(2)
493.         system("cls")
494.         category_bottom_sales()
495.
496.     elif ans == 4:
497.         sleep(2)
498.         system("cls")
499.         category_bottom_searches()
500.
501.     elif ans == 5:
502.         sleep(2)
503.         system("cls")
504.         product_review()
505.
506.     elif ans == 6:
507.         sleep(2)
508.         system("cls")
509.         inf_sales_LifeStore()
510.
511.     else:
512.         sleep(2)
513.         system("cls")
514.         print('REGRESA PRONTO...')
515.         exit()
516.
517.     break
518.
519.     elif user == users['Jaime Alonso'] and password ==
passwords['Jaime Alonso']:
520.         sleep(2)
521.         system("cls")
522.         print('\tBIENVENIDE A LIFESTORE ' +
str.upper(users['Jaime Alonso']))
523.
524.         print('\n1. Lista de los productos con mayores
ventas')
525.         print('2. Lista de los productos con mayores
búsquedas')
526.         print('3. Lista (por categoría) de los productos con
menores ventas')
527.         print('4. Lista (por categoría) de los productos con
menores búsquedas')
528.         print('5. Lista de los productos por reseña')
529.         print('6. Información financiera mensual y anual de
LifeStore')
530.         print('7. Salir')
531.

```

```

532.         ans = int(input('\nSelecciona una opción: '))
533.
534.         #Al ingresar a cualquier opción, borrará el contenido
de la terminal
535.         #y mostrará lo solicitado
536.         if ans == 1:
537.             sleep(2)
538.             system("cls")
539.             top_sales_product()
540.
541.         elif ans == 2:
542.             sleep(2)
543.             system("cls")
544.             top_searches_product()
545.
546.         elif ans == 3:
547.             sleep(2)
548.             system("cls")
549.             category_bottom_sales()
550.
551.         elif ans == 4:
552.             sleep(2)
553.             system("cls")
554.             category_bottom_searches()
555.
556.         elif ans == 5:
557.             sleep(2)
558.             system("cls")
559.             product_review()
560.
561.         elif ans == 6:
562.             sleep(2)
563.             system("cls")
564.             inf_sales_LifeStore()
565.
566.         else:
567.             sleep(2)
568.             system("cls")
569.             print('REGRESA PRONTO...')
570.             exit()
571.
572.         break
573.     else:
574.         print('El usuario y/o contraseña son incorrectos')
575.         intentos -= 1
576.         print('Tienes ' + str(intentos) + ' intentos más\n')
577.         if intentos == 0:
578.             print("¡Acceso bloqueado! Intentalo más tarde")
579.             exit()
580.     sleep(2)
581.     system("cls")

```