

PROYECTO 1 INTRODUCCIÓN A PYTHON



LIFESTORE - EMTECH INSTITUTE
ALFREDO GOMEZ CHAVEZ

INDICE

1. Introducción	Pág. 3
2. Definición del código	Pág. 4
2.1. Ventas de los productos	Pág. 5
2.2. Ventas por categoría-.....	Pág. 5
2.3. Productos por búsqueda	Pág. 6
2.4. Productos por Score	Pág. 7
2.5. Total de ingresos	Pág. 7
2.6. Total anual	Pág. 8
2.7. Ventas e ingresos mensuales	Pág. 9
2.8. Menu	Pág. 10
2.9. Impresiones	Pág. 11
3. Solución del problema	Pág. 13
4. Conclusión	Pág. 16

INTRODUCCIÓN

LifeStore es una tienda virtual que maneja una gran cantidad de componentes (hardware) para computadoras. La gerencia de ventas, tras realizar un análisis de los productos, se percató que la empresa tiene una importante acumulación de inventario. De igual manera, se ha identificado una reducción en las búsquedas de un grupo importante de productos, siendo este motivo de una disminución sustancial de sus ventas del último trimestre.

Para poder darle una solución a este problema es importante conocer el impacto que tiene cada producto de acuerdo a las búsquedas y ventas durante el último año, así como identificar cual es la categoría con mayor cantidad de productos rezagados.

Se ha hecho un análisis de los productos para poder descartar a aquellos que no se han vendido a lo largo de los meses en los que hubo ventas, calculando la cantidad de búsquedas para conocer si realmente existe interés por estos.

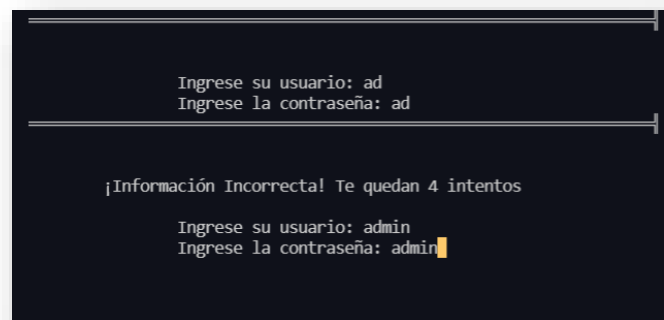
Se detallarán los métodos utilizados para la obtención de dichos cálculos, así como los ingresos obtenidos durante el tiempo de ventas activo.

Definición del código

Comencé inicializando las variables de separador, el usuario, la contraseña y los intentos, en este caso como el login apenas arrancará lo definí con 1, todas estas variables son para utilizarse durante el primer bucle while, el cual es el login.

Al ejecutar el programa el login es el que inicia, teniendo como función el comprobar que has introducido correctamente los valores de usuario y contraseña, si alguno de estos dos valores es incorrecto el ciclo while te dará otros 4 intentos para poder colocar los datos correctos, de igual manera esta información te la muestra en la pantalla. Al introducir correctamente los datos podrás entrar al menú correspondiente, de lo contrario una vez que se termine la cantidad de intentos el programa terminará.

Enseguida se tiene a las listas correspondientes con las que cuenta la tienda, se muestran los productos con su costo y datos adicionales



```
"""
This is the Lifestore_Saleslist data:
lifestore_searches = [id_search, id product]
lifestore_sales = [id_sale, id_product, score(1-5), date, refund(1-0)]
lifestore_products = [id_product, name, price, category, stock]
"""
> lifestore_products = [ ...
]
> lifestore_sales = [ ...
]
> lifestore_searches = [ ...
]
"""
```

como categoría y stock, al igual se tienen las ventas realizadas de la tienda a lo largo del último año, y por otro lado tenemos registradas las búsquedas realizadas de cada producto

Terminando las listas se tiene el inicio del código que se realizó para cada método, el código está separado por líneas de comentario en cada apartado, al igual está correctamente comentado y con los datos

de cada arreglo que se ocupará en dicho apartado.

Se comenzó declarando un arreglo de productos llamado "products", se planea llenar este arreglo con los siguientes atributos de cada producto como el id, nombre, ventas, devoluciones, score, etc.

Este arreglo primero fue llenado en los apartados de id, nombre, ventas y categoría, provenientes del arreglo de lifestore_products. Se agregó producto por producto.

VENTAS DE LOS PRODUCTOS

Luego tenemos el código empleado para calcular las ventas de los productos.

Comencé realizando un for, recorriendo cada venta del arreglo de ventas de la lista ya proporcionada, enseguida recorri cada producto del arreglo que hice anteriormente, el de products, esto para comparar que el id del producto de la venta sea igual al id del producto del arreglo de products, de ser que los id coincidan se realizará la inserción de los datos de la venta en el producto, estos datos son los puntos de Score, se suma 1 a la cantidad de scores, se suman las devoluciones, y si no tiene devolución alguna también se le suma 1 a la cantidad de ventas del producto.

```
# Sumamos la venta, score, cantidad a
for sale in lifestore_sales:
    for product in products:
        if sale[1] == product[0]:
            product[5] += sale[2]
            product[6] += 1
            product[7] += sale[4]
            if sale[4] == 0:
                product[3] += 1
```

```
for i in range(len(products)):
    for j in range(len(products)-1):
        if products[j][3] < products[j+1][3]:
            aux = products[j+1]
            products[j+1] = products[j]
            products[j] = aux

# Los primeros 15 se van a BEST_SALES
best_sales = [] # más vendidos
worst_sales = [] # menos vendidos -- Aquí solo guardaremos e
for i in range(15):
    best_sales.append(products[i])
for i in range(len(products)):
    worst_sales.append(products[len(products)-i-1])

MasVendidos = "\t<< 15 Productos con mayores ventas >>\n\nIDV
for sale in best_sales:
    MasVendidos += "%d\t%.25s...\t %d\t %d\n" % (
        sale[0], sale[1], sale[3], sale[7])
```

Después se realiza un método burbuja para poder ordenar el arreglo de products con respecto a las ventas de los productos.

Enseguida, en 2 arreglos colocó los primeros 15 productos con mayores ventas, y todo el arreglo invertido de products pero invertido. Este segundo arreglo (llamado worst_sales) será utilizado en la siguiente sección para ordenarlos por categoría.

Al final solo agregué algunas líneas de texto con cierto formato para poder llamarlo una vez que queramos imprimir estos datos.

VENTAS POR CATEGORIA

En el apartado para calcular por categoría se utilizará el arreglo antes definido como "worst_sales" el cual tendrá los mismos valores.

Entonces, se realiza un nuevo arreglo llamado "categorias" el cual tendrá el nombre de las distintas categorías sin repetirse, recorriendo un for se puede comprobar si la categoría de cada producto ya existe en el arreglo que acabamos de definir, si ya existe no se agrega, de lo contrario se agrega.

Enseguida hacemos instancia de 2 arreglos, uno para los más buscados y otro para los menos buscados (most y least). Con el arreglo de products agregamos los más buscados, y con el mismo arreglo pero de manera invertida agregamos los menos buscados, al final solo agregamos a las 2 variables de texto cada valor de cada productos con sus búsquedas utilizando iteraciones con for.

```

# agregamos los productos que tienen cada producto
for search in lifestore_searches:
    for product in products:
        if search[1] == product[0]:
            product[8] += 1

# hacemos metodo burbuja (otra vez) para ordenar
for i in range(len(products)):
    for j in range(len(products)-1):
        if products[j][8] < products[j+1][8]:
            aux = products[j+1]
            products[j+1] = products[j]
            products[j] = aux

# acomodamos en dos arreglos, uno de mayor a menor y otro de menor a mayor
most_searches = [] # productos más buscados
least_searches = [] # productos menos buscados
for i in range(20):
    most_searches.append(products[i])

for i in range(20):
    least_searches.append(products[len(products)-i-1])

MasBuscados = "\t<< 20 Productos más buscados >>\n\nID\tNombre del producto\tBúsquedas\n"
for search in most_searches:
    MasBuscados += "%d\t%.25s...\t %d\n" % (search[0], search[1], search[8])

MenosBuscados = "\t<< 20 Productos menos buscados >>\n\nID\tNombre del producto\tBúsquedas\n"
for search in least_searches:
    MenosBuscados += "%d\t%.25s...\t %d\n" % (search[0], search[1], search[8])

#

```

PRODUCTOS POR SCORE

Para calcular los Score de los productos se realizó lo siguiente:

- Declaramos un arreglo con los productos comprados (bought_products).
- Con interacciones comparamos si las ventas de cada productos del arreglo de products es distinto a 0.
- Si es así dividimos la suma de Score que tiene entre la cantidad de Scores dados (practicamente realizamos un promedio).
- Agregamos dicho producto con solo algunos atributos al nuevo arreglo bought_products.
- Una vez finalizado el for, realizamos un metodo burbuja para ordenar el arreglo bought_products de acuerdo al score, irá del mejor al peor.
- Agregamos 2 arreglos, uno para los mejores y otro para los peores.
- Asignamos los productos a dichos arreglos dos for para darle mas claridad al codigo.
- Terminamos agregando los valores a las variables de texto que tenemos para imprimir los elementos.

```
bought_products = []
for product in products:
    if product[6] != 0:
        product[5] /= product[6]
        bought_products.append(
            [product[0], product[1], product[5], product[7]])
# Realizamos el metodo de la burbuja para ordenarlos por busquedas
for i in range(len(bought_products)):
    for j in range(len(bought_products)-1):
        if bought_products[j][2] < bought_products[j+1][2]:
            aux = bought_products[j+1]
            bought_products[j+1] = bought_products[j]
            bought_products[j] = aux
# Pasamos los primeros 10 al arreglo best y los 10 ultimos al arreglo worst
best_scores = [] # productos mejor rankeados
worst_scores = [] # productos peor rankeados
for i in range(10):
    best_scores.append(bought_products[i])
for i in range(10):
    worst_scores.append(bought_products[len(bought_products)-i-1])

MejoresScore = "\t<< 10 Productos con mejor Score >>\n\nID\tNombre del producto\tScore"
for product in best_scores:
    MejoresScore += "%d\t%.25s...\t %.1f\t\t%d\n" % (
        product[0], product[1], product[2], product[3])
PeoresScore = "\t<< 10 Productos con peor Score >>\n\nID\tNombre del producto\tScore"
for product in worst_scores:
    PeoresScore += "%d\t%.25s...\t %.1f\t\t%d\n" % (
        product[0], product[1], product[2], product[3])
```

Con esto hemos terminado los primeros 2 puntos que se han solicitado.

TOTAL DE INGRESOS

Para calcular el total de ingresos y ventas mensuales se hizo de la siguiente manera.

Declaramos un arreglo de meses, una variable de ingresos y otra de ventas para el año 2020.

Dentro de un arreglo sacamos cada venta del arreglo de ventas, extraemos la variable que tiene de la fecha a una nueva variable para darl mejor visibilidad, enseguida esa fecha se compone para extraer solo el año de la venta, luego comparamos si dicho año es igual al 2020, y de ser así se realizan alguna intrucciones:

- Comparamos si no tiene devoluciones, para poder sumar 1 a las ventas a la variable de ventas antes hecha.
- Recorremos cada producto del arreglo de productos para comparar si el id de los productos, tanto de la venta como del producto es igual y así sumar lo que cuesta dicho producto.
- Se compara si el mes de la venta no está dentro del arreglo de meses antes hecho.

```

meses = [] # Declaro un arreglo de meses
ingreso_2020 = 0
ventas_2020 = 0
for sale in lifestore_sales:
    fecha = sale[3]
    anoSale = (fecha[-4]+fecha[-3]+fecha[-2]+fecha[-1])
    if anoSale == '2020':
        if sale[4] == 0:
            ventas_2020 += 1
        for product in products:
            if sale[1] == product[0]:
                ingreso_2020 += product[4]
        mesSale = (fecha[3]+fecha[4])
        if mesSale not in meses:
            meses.append(mesSale)

promedio_ventas = (ventas_2020/len(meses))
promedio_ingresos = (ingreso_2020/len(meses))
VentasPromedioMensuales = "\t<< Ventas promedio mensuales
                             promedio_ventas, promedio_ingresos)

```

Una vez hecho esto, procedemos a sacar el promedio de ventas dividiendo las ventas del 2020 entre los meses activos, y de igual manera dividimos los ingresos del 2020 entre los meses activos.

Despues solo agregamos a nuestra variable de texto dicha información.

TOTAL ANUAL

Para calcular el total anual se realizó lo siguiente:

- Inicializamos la variables de ingreso_anual con 0.
- Realizamos un for sacando cada venta del arreglo de ventas.
- Asignamos la fecha de la venta en otra variable para darle mejor visibilidad al código.
- Descomponemos dicha fecha para extraer el año.
- Comparamos el año con '2020' y si no existe devolucion.
- Si la comparación es exitosa realizamos otro for para sacar el producto del arreglo de productos.
- Comparamos los id de los productos.
- Si se coincide procedemos a sumar el costo del producto a la variable Ingreso_anual.

- Al final solo agregamos el texto a nuestra variable que será visible en el menu.

```

ingreso_anual = 0

for sale in lifestore_sales:
    fecha = sale[3]
    anoSale = (fecha[-4]+fecha[-3]+fecha[-2]+fecha[-1])
    if(anoSale == '2020' and sale[4] != 1):
        for product in lifestore_products:
            if(sale[1] == product[0]):
                ingreso_anual += product[2]
IngresoAnual = '<< Ingreso Anual 2020: %.2f >>' % (ingreso_anual)

```

VENTAS E INGRESOS MENSUALES

Al final, para calcular las ventas mensuales se tuvo que iniciar con un arreglo vacío de ventas mensuales y otro lleno de datos con los meses (con nombre y número de mes).

Se realiza un for para extraer el mes del arreglo con los meses ya hecho, se compara si existe el número del mes con el arreglo de meses que ya existía en pasos anteriores, y si existe se le asigna al arreglo de ventas mensuales. Eso se hace con el fin de colocar los meses que estuvieron activos durante el año 2020.

Después se realiza un for que extrae la venta del arreglo de ventas, se le asigna la fecha de la venta a otra variable para poder extraer el año de la venta y así poder comparar si el año es 2020, si cumple esta condición se procederá a asignarle el mes de la venta a otra variable y enseguida realizar un for que extrae cada producto del arreglo de products, luego compara si los id coinciden (el del producto de la venta y del producto), y si es así se realiza otro for que extrae la venta mensual del arreglo de ventas mensuales para poder comparar si el mes de la venta coincide con algún mes y así asignarle los valores correspondientes.

```

ventas_mensuales = []
nombres_meses = [['Enero', '01'], ['Febrero', '02'], ['Marzo', '03'], ['Abril', '04'],
                  'Julio', '07'], ['Agosto', '08'], ['Septiembre', '09'], ['Octubre', '10'], ['Novien
for mes in nombres_meses:
    if mes[1] in meses:
        ventas_mensuales.append([mes[0], mes[1], 0, 0])

for sale in lifestore_sales:
    fecha = sale[3]
    anoSale = (fecha[-4]+fecha[-3]+fecha[-2]+fecha[-1])
    if anoSale == '2020':
        mesSale = (fecha[3]+fecha[4])
        for product in products:
            if sale[1] == product[0]:
                for vm in ventas_mensuales:
                    if mesSale == vm[1]:
                        vm[2] += 1
                        vm[3] += product[4]

# Aquí solo realizo el metodo burbuja de acuerdo al a los ingresos de cada mes
for i in range(len(ventas_mensuales)):
    for j in range(len(ventas_mensuales)-1):
        if ventas_mensuales[j][3] < ventas_mensuales[j+1][3]:
            aux = ventas_mensuales[j+1]
            ventas_mensuales[j+1] = ventas_mensuales[j]
            ventas_mensuales[j] = aux

VentasMensuales = "\n\t<< Meses con más Ingresos 2020>>\n\n\tVentas\tIngresos\tMes\n"
for vm in ventas_mensuales:
    VentasMensuales += "\t%d\t$%.2f\t$%.10s\n" % (vm[2], vm[3], vm[0])

```

Al final se realiza un metodo burbuja para poder ordenar el arreglo de ventas mensuales de acuerdo a los ingresos obtenidos, así mismo asignamos los valores de texto a las variables que se mostrarán en el menú.

MENÚ

Por ultimo tenemos el menú, el cual hará visible todos los datos que hemos generado anteriormente.

Se comenzó inicializando algunas variables, ya sabemos que tenemos la variable del separador al inicio del programa, entonces aquí definimos un arreglo con las opciones disponibles para limitar al usuario a solo introducir datos ya establecidos, luego asignamos

otros separador en donde se colocará el input, tambien tenemos un texto de salida, un espaciado de 30 lineas y un menú con las opciones disponibles.

Despues definimos una variable de opcion, la cual dará una entrada en un input, y enseguida tenemos un while, este while por defecto está en true, ya que se debe entrar automaticamente a este.

```
opciones = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']
separador_op = '\n|'
salida = '\n{ Si desea salir presione 9, de lo contrario presione 0 }'
espacio = '\n'*30
menu = """
.....: 0 1 2 3 4 5 6 7 8 9 - TIENDA VIRTUALA :.....

    Elige la opción que deseas ver:

    1 - Productos más vendidos
    2 - Productos menos vendidos por categoria
    3 - Productos con mayores busquedas
    4 - Productos con menos busquedas
    5 - Productos mejor calificados
    6 - Productos peor calificados
    7 - Ventas e ingresos promedio mensuales
    8 - Ingreso anual
    9 - Salir
"""
```

IMPRESIONES

```
opcion = 0
while True:
    print(espacio)
    print(separador)
    if opcion == 0:
        print(menu)
    else:
        if opcion == 1:
            print(MasVendidos)
        if opcion == 2:
            print(MenosVendidos)
        if opcion == 3:
            print(MenosBuscados)
        if opcion == 4:
            print(MejoresScore)
        if opcion == 5:
            print(MenosVendidos)
        if opcion == 6:
            print(PeoresScore)
        if opcion == 7:
            print(VentasPromedioMensuales)
            print(VentasMensuales)
        if opcion == 8:
            print(IngresoAnual)
        print(salida)
    opcion = input(separador_opt+' ')
while opcion not in opciones:
    print("\t\t\tSelecciona una opción valida!")
    opcion = input(separador_opt+' ')
opcion = int(opcion)
if opcion == 9:
    break
```

Dentro del while tenemos 2 impresiones, la de espacio y el separador, esto para darle cierto formato y diseño a las impresiones. Enseguida tenemos la opcion 0 que es la del menu, a esta se accede directamente ya que es el valor inicial de la variable opcion, despues están todas las demas opciones de 1 al 8, y para el 9 se tiene asignado un break para salir del menu. Para hacer uso de este menú lo primero es asignar un valor a input, luego se realiza una condicion en un while que exige que la opcion dada no esté dentro de las opciones establecidas para ejecutar un print "Opción no valida" y enseguida otro input hasta que se asigne un valor correcto. Una vez que se asigna un valor correcto se para a int y se ejecuta dentro del while.

El programa ejecutandose se mira de la siguiente manera:

Login:

```
Ingrese su usuario: admin
Ingrese la contraseña: admin
```

Menu:

```
.....: INFORME - TIENDA VIRTUAL :.....

Elige la opción que deseas ver:

1 - Productos más vendidos
2 - Productos menos vendidos por categoria
3 - Productos con mayores busquedas
4 - Productos con menos busquedas
5 - Productos mejor calificados
6 - Productos peor calificados
7 - Ventas e ingresos promedio mensuales
8 - Ingreso anual
9 - Salir
```

Opcion:

```
<< 15 Productos con mayores ventas >>

ID      Nombre                               Ventas  Devoluciones
54      SSD Kingston A400, 120GB,...           49      1
3       Procesador AMD Ryzen 5 26...           42      0
5       Procesador Intel Core i3-...           20      0
42      Tarjeta Madre ASRock Micr...           18      0
57      SSD Adata Ultimate SU800,...          15      0
4       Procesador AMD Ryzen 3 32...           13      0
29      Tarjeta Madre ASUS micro ...           13      1
2       Procesador AMD Ryzen 5 36...           12      1
47      SSD XPG SX8200 Pro, 256GB...           11      0
12      Tarjeta de Video ASUS NVI...            9      0
48      SSD Kingston A2000 NVMe, ...            9      0
7       Procesador Intel Core i7-...             7      0
44      Tarjeta Madre MSI ATX B45...             6      0
18      Tarjeta de Video Gigabyte...             5      0
8       Procesador Intel Core i5-...             4      0

{ Si desea salir presione 9, de lo contrario presione 0 }
```

Solución

Para darle una solución al problema se siguió una serie de pasos:

- Lista de productos.
- a) PUNTO 1
 - Lista de 15 - MÁS VENDIDOS.
 - Lista de 20 – MÁS BUSCADOS.
 - Lista de 5 * categoría – CATEGORIA MENOS VENDIDOS.
 - Lista de 20 – MENOS BUSCADOS.
- b) PUNTO 2
 - Lista de 10 – MEJORES SCORE.
 - Lista de 10 – PEORES SCORE.
- c) PUNTO 3
 - Total de ingresos y ventas mensuales.
 - Total anual.
 - Lista de 12 - Ventas mensuales de mayor a menor.

1. Productos más vendidos

ID	Nombre	Ventas	Devoluciones
54	SSD Kingston A400, 120GB...	49	1
3	Procesador AMD Ryzen 5 26...	42	0
5	Procesador Intel Core i3-...	20	0
42	Tarjeta Madre ASRock Micr...	18	0
57	SSD Adata Ultimate SU800...	15	0
4	Procesador AMD Ryzen 3 32...	13	0
29	Tarjeta Madre ASUS micro...	13	1
2	Procesador AMD Ryzen 5 36...	12	1
47	SSD XPG SX8200 Pro, 256GB...	11	0
12	Tarjeta de Video ASUS NVI...	9	0
48	SSD Kingston A2000 NVMe...	9	0
7	Procesador Intel Core i7-...	7	0
44	Tarjeta Madre MSI ATX B45...	6	0
18	Tarjeta de Video Gigabyte...	5	0
8	Procesador Intel Core i5-...	4	0

Productos rezagados

				procesadores
9	Procesador Intel Core i3-...	0	0	
1	Procesador AMD Ryzen 3 33...	2	0	
6	Procesador Intel Core i9-...	3	0	
8	Procesador Intel Core i5-...	4	0	
7	Procesador Intel Core i7-...	7	0	
				tarjetas de video
27	Tarjeta de Video VisionTe...	0	0	
26	Tarjeta de Video VisionTe...	0	0	
24	Tarjeta de Video PNY NVID...	0	0	
23	Tarjeta de Video MSI Rade...	0	0	
20	Tarjeta de Video Gigabyte...	0	0	
				tarjetas madre
46	Tarjeta Madre Gigabyte mi...	0	1	
45	Tarjeta Madre ASRock ATX ...	0	1	
43	Tarjeta Madre ASUS ATX RO...	0	0	
41	Tarjeta Madre ASUS micro ...	0	0	
39	ASUS T. Madre uATX M4A88T...	0	0	
				discos duros
59	SSD Samsung 860 EVO, 1TB,...	0	0	
58	SSD para Servidor Lenovo ...	0	0	
56	SSD para Servidor Lenovo ...	0	0	
55	SSD para Servidor Supermi...	0	0	
53	SSD Addlink Technology S7...	0	0	
				memorias usb
61	Kit Memoria RAM Corsair V...	0	0	
60	Kit Memoria RAM Corsair D...	1	0	
				pantallas
73	Samsung Smart TV LED UN55...	0	0	
72	Hisense Smart TV LED 50H8...	0	0	
71	Samsung Smart TV LED UN32...	0	0	
70	Samsung Smart TV LED 43, ...	0	0	
69	Hisense Smart TV LED 40H5...	0	0	
				bocinas
83	Ghia Bocina Portátil BX50...	0	0	
82	Ghia Bocina Portátil BX40...	0	0	
81	Ghia Bocina Portátil BX90...	0	0	
80	Ghia Bocina Portátil BX80...	0	0	
79	Naceb Bocina Portátil NA-...	0	0	
				audifonos
96	Klip Xtreme Audifonos Bla...	0	0	
95	Iogear Audifonos Gamer GH...	0	0	
93	Ginga Audifonos con Micró...	0	0	
92	Getttech Audifonos con Mi...	0	0	
91	Genius GHP-400S Audifonos...	0	0	

2. Productos con mejor reseña

<< 10 Productos con mejor Score >>			
ID	Nombre del producto	Score	Devoluciones
85	Logitech Audifonos Gamer ...	5.0	0
67	TV Monitor LED 24TL520S-P...	5.0	0
7	Procesador Intel Core i7-...	5.0	0
8	Procesador Intel Core i5-...	5.0	0
21	Tarjeta de Video MSI AMD ...	5.0	0
66	TCL Smart TV LED 55S425 5...	5.0	0
6	Procesador Intel Core i9-...	5.0	0
49	Kit SSD Kingston KC600, 1...	5.0	0
1	Procesador AMD Ryzen 3 33...	5.0	0
25	Tarjeta de Video Sapphire...	5.0	0

3. Sugerencia

Como solución a la consigna realmente se deben de tomar en cuenta los productos cuyas reseñas fueron las peores, ya que estos productos por muchas ventas que cuenten realmente no tienen un impacto positivo en cuanto a los ingresos, ya que,

aunque no sean devueltos no consiguen tener una buena experiencia con los clientes. En cuanto a los productos con mejores reseñas es posible considerar obtener más productos de su misma categoría, lo más viable sería considerar tener en stock productos similares o una versión más actual de dicho producto en lugar de volver a tener el stock lleno, ya que el mundo de la tecnología cambia constantemente y LifeStore no sería competencia para aquellas empresas actualizadas en cuanto a tecnología.

Por otro lado, podría considerarse retirar del mercado la categoría de Memorias USB, ya que, como se muestra en el programa realmente no existe una demanda de dichos productos, en 9 meses solamente se logró vender 1 pieza, apenas representa ingresos.

Conclusión

La tienda virtual LifeStore podría obtener mayor número de ventas cambiando los productos que no tienen búsquedas con alguna versión actual de los que tienen mayor número de ventas y búsquedas. Existen categorías casi intactas que realmente no aportan mucho a los ingresos de la empresa, quizá se podría compensar con un mejor manejo de marketing para no dar por hecho que los productos sin ventas son pérdidas totales.

Las empresas de venta de hardware normalmente cuentan con una gran cantidad de productos que se van considerando como rezagados al paso del tiempo, y estos nunca tendrán una fecha de caducidad, por lo que se puede aprovechar esto para considerar dar algunos productos en ofertas ya sea individualmente o en conjunto.

El programa realizado cuenta con la comparación de ingresos por cada mes, esto puede ayudar a considerar alguna temporada con mayor demanda, quizá al inicio de año se venden más ciertos productos y a mediados del año otros, realmente se puede dar un diagnóstico completo si partimos de los meses ahora que tenemos todas estas listas, no hay duda que se pueden dar en oferta los productos rezagados en dichos meses.