

Diego Daniel Muñoz Batista

## Análisis de LifeStore



**EMTECH**  
Emerging Technologies Institute

# Índice

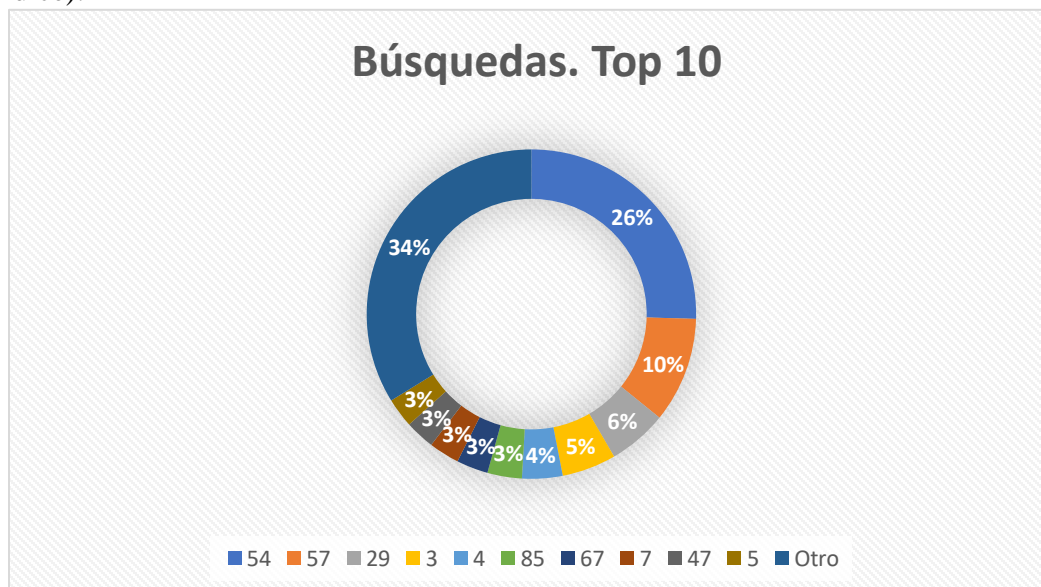
Página 3.....	Análisis de búsquedas
Página 4.....	Análisis de ventas
Página 6 .....	Análisis mensual
Página 7.....	Análisis reseña de servicio
Página 9.....	Análisis inventario
Página 11 .....	Recomendaciones

LifeStore es una empresa que vende diversos productos de tecnología. Los productos están divididos en ocho categorías principales: procesadores, tarjetas de video, tarjetas madre, usb, discos duros, pantallas, audífonos, y bocinas. La empresa ha identificado un problema: están acumulando demasiados productos en el inventario. Para dar una recomendación es necesario analizar los datos que tenemos de LifeStore; ventas, inventario, búsqueda por producto, y la valoración que los clientes le dan a los productos que compran. No obstante, es pertinente mencionar que para hacer un análisis más acertado sería preferible tener un horizonte más amplio de los datos. Con los datos actuales no es posible saber si la tienda ha estado disminuyendo sus ventas debido a un rendimiento por debajo de lo esperado o porque la tienda suele vender menos en estos meses (no sabemos si hay estacionalidad en las ventas de la tienda).

Como la seguridad de datos es una prioridad para todas las compañías hemos desarrollado un sistema de inicio de sesión para que solo los administradores de LifeStore puedan acceder a la información.

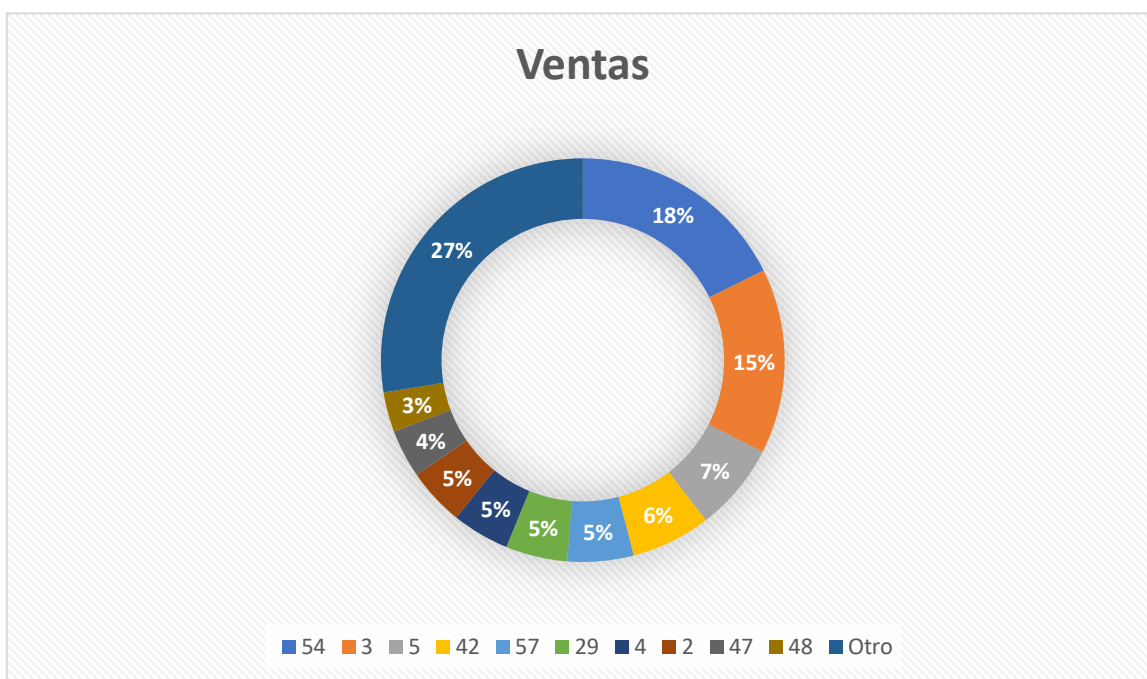
## ANÁLISIS DE BÚSQUEDAS

Ahora bien, analizando los datos podemos observar distintas características. En primer lugar, observamos que las búsquedas están concentradas en pocos productos. Los consumidores realizan más de una tercera parte de las búsquedas solo en dos productos y más de dos terceras partes en 10 productos (Ver tabla completa de frecuencias en Apéndice).



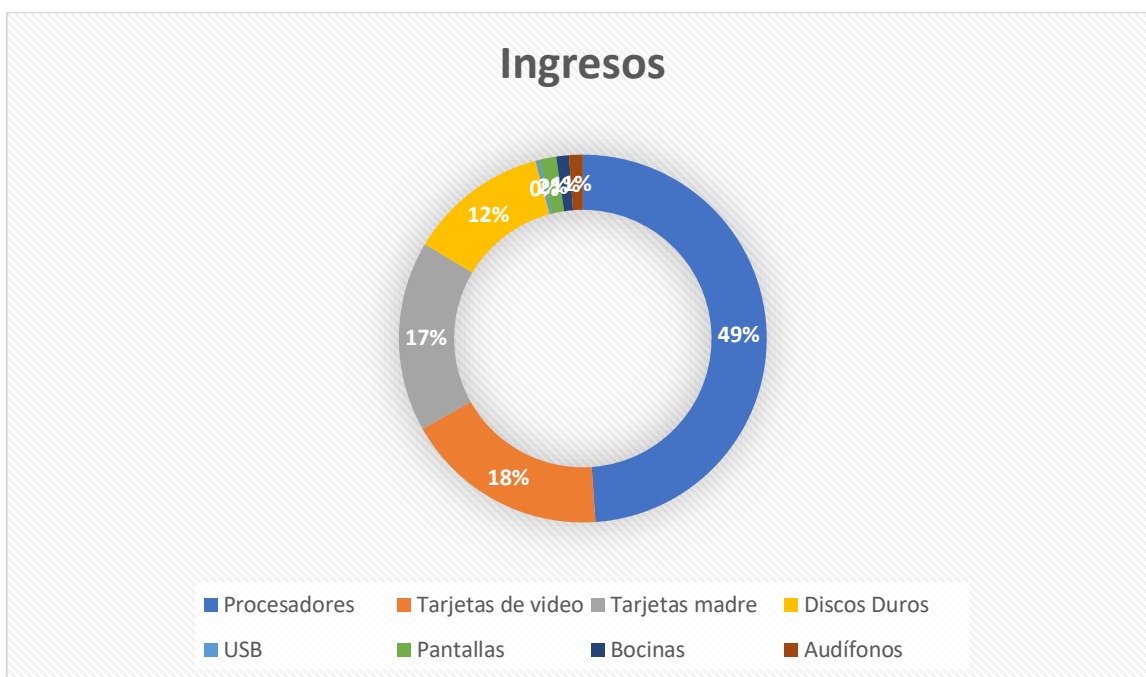
## ANÁLISIS VENTAS

En segundo lugar, las ventas presentan un comportamiento similar, el 82% de las compras están concentradas en 10 productos. Además, el producto más vendido también es el más buscado (producto 54). Si observamos ambas listas encontraremos que 7 de los 10 productos más buscados también se encuentran entre los 10 productos más vendidos. El producto 85 ha sido buscado 35 veces y solo ha logrado 2 ventas en el año, y el producto 67 lo han buscado 32 veces y se ha vendido uno solo de ese producto. Estos productos deben presentar características que no le agradan al consumidor ya que, a pesar de buscarlo y ver sus características, pocos deciden comprarlo.



Para hacer un análisis más completo de las ventas podemos observar cuáles son los productos que más ingresos generan. **Los ingresos totales del año fueron: \$760,177.** Usaremos las 8 categorías principales que vende LifeStore para saber cuáles son las que generan más ingresos. Los procesadores generan alrededor de la mitad de los ingresos de la compañía. Si sumamos los ingresos de procesadores, tarjetas de video, discos duros y tarjetas madre obtenemos el 96% de los ingresos de la compañía.

<b>Categoría</b>	<b>Ingresos</b>
Procesadores	\$371,726
Tarjetas de video	\$136,224
Tarjetas madre	\$127,321
Discos Duros	\$93,496
Pantallas	\$11,278
Bocinas	\$8,478
Audífonos	\$9,135
USB	\$2,519
<b>Total</b>	<b>\$760,177</b>



Ahora podemos hacer un análisis más a fondo de cada categoría. En primer lugar, en los procesadores es notable que el único producto que no genera ingresos es el producto 9. En segundo lugar, en las tarjetas de video los productos 14, 15, 16, 19, 20, 23, 24, 26 y 27 no generan ingresos, son la mitad de productos dentro de la categoría de tarjetas de video. En tercer lugar, en las tarjetas madre los productos 30, 32, 34, 35, 36, 37, 38, 39, 41 y 43 no generan ingresos, son más de la mitad de los productos dentro de esta categoría. En cuarto lugar, en los discos duros los productos 53, 55, 56, 58 y 59 no generan ingresos y son la

mitad de los productos dentro de esta categoría. En quinto lugar, en las pantallas los productos 62, 63, 64, 65, 68, 69, 70, 71, 72 y 73 no generan ingresos, solo dos productos generan ingresos dentro de esta categoría. En sexto lugar, en las bocinas solo un producto genera ingresos: el producto 74. En séptimo lugar, en los audífonos solo los productos 84, 85, 89 y 94 generan ingresos. Finalmente, en octavo lugar, en los USB el producto 60 se vende mientras que el 61 no genera ingresos.

## ANÁLISIS MENSUAL

También analizaremos los ingresos por mes del establecimiento. **El ingreso promedio mensual es: \$84,464**

<b>Mes</b>	<b>Ingresos</b>
Enero	\$124,705
Febrero	\$110,139
Marzo	\$164,729
Abril	\$193,295
Mayo	\$96,135
Junio	\$36,949
Julio	\$26,949
Agosto	\$3,077
Septiembre	\$4,199
<b>Total</b>	<b>\$760,177</b>

Vemos que hay tendencia decreciente en los ingresos que genera LifeStore pero, como mencionamos anteriormente, con los datos actuales no podemos asegurar que no es un comportamiento normal de sus ventas en cada año.

Si añadimos las devoluciones al análisis, podemos observar que solo hay 9 devoluciones y el único producto que ha sido devuelto más de una vez es el 31. Las devoluciones tienen un valor total de \$22,261.

## ANÁLISIS DE RESEÑA DE SERVICIO

20 productos mejor valorados:

Producto	Valoración Promedio
85	5
84	5
67	5
66	5
60	5
52	5
50	5
49	5
40	5
28	5
25	5
22	5
21	5
11	5
8	5
7	5
6	5
1	5
53	4.86
3	4.80

20 productos peor valorados (solo contando los que sí tienen alguna valoración):

<b>Producto</b>	<b>Valoración</b>
17	1
45	1
31	1.8
46	2
89	3
10	4
13	4
94	4
29	4.1
2	4.2
18	4.4
4	4.4
33	4.5
74	4.5
47	4.5
42	4.5
44	4.6
48	4.6
51	4.6
5	4.7

Es importante resaltar que la mayoría de los productos no tienen suficientes evaluaciones para saber la opinión general del público, quizá fueron eventos aislados. De todas formas, si es un producto con pocas evaluaciones también es un producto que se vende poco.



## ANÁLISIS INVENTARIO

Los 20 productos con más existencias en el inventario son:

Producto	Inventario
3	987
67	411
54	300
4	295
41	286
68	239
92	232
66	188
2	182
26	180
63	146
93	139
5	130
31	120
7	114
39	98
69	94
84	83
64	71
37	60

El producto 26 tiene cero ventas en el año y aun hay 180 en el inventario, el producto 37 tiene 60 existencias en el inventario y cero ventas en el año, el producto 39 tiene 98 existencias en el inventario y cero ventas en el año, y el producto 41 tiene 286 existencias en el inventario y ha generado cero ventas en el año.

Los 20 productos con menos existencias en el inventario son:

<b>Producto</b>	<b>Inventario</b>
12	0
21	0
22	0
42	0
44	0
51	0
13	1
17	1
40	1
53	1
74	1
77	1
90	1
11	2
24	2
34	2
78	2
95	2
96	2
28	3

Los productos 42 y 12 ya no tienen existencias en el inventario y, también, se encuentran en el top 10 de productos más vendidos.

## RECOMENDACIONES

Como vimos, LifeStore tiene gran variedad de productos en las categorías de pantallas y audífonos, aunque pocos productos son los que generan las ventas en cada una de ellas. Asimismo, los productos 85 y 67 (son pantallas y audífonos) son buscados repetidamente por los consumidores pero no generan ventas. Más aun, el producto 85 tiene 35 existencias en el inventario y el producto 67 tiene 411 existencias. Por lo tanto, la primera recomendación sería rematar las existencias de estos dos productos (85 y 67) y, posteriormente, dejar de venderlos.

Después, el análisis por categoría nos muestra que la tienda puede prescindir del producto 9 ya que es el único de los procesadores que no presenta ingresos en todo el año y, además, tiene 35 existencias en el inventario. La segunda recomendación sería rematar las existencias del producto 9 y no volver a adquirir este producto.

Las tarjetas de video 14, 15, 16, 19, 20, 23, 24, 26 y 27 no generan ingresos. Aunque podrían prescindir de la mayoría de estos productos debido a la gran variedad que tiene LifeStore, hay uno que tiene una razón extra para dejar de venderlo en la tienda; el producto 26, además de no generar ingresos, tiene 180 existencias en el inventario. La tercera recomendación sería rematar estas unidades y no adquirir más del producto 26.

En cuanto a las tarjetas madre los productos 30, 32, 34, 35, 36, 37, 38, 39, 41 y 43 no generan ingresos. El producto 41 tiene 286 existencias, el 39 tiene 98 existencias y el 37 tiene 60 existencias. Por lo tanto, la cuarta recomendación sería rematar los productos 41, 39 y 37.

En la categoría de discos duros los productos 53, 55, 56, 58 y 59 no generan ingresos y son la mitad de los productos dentro de esta categoría. Por esta razón, LifeStore podría prescindir de algunos de ellos.

En las pantallas, los productos 62, 63, 64, 65, 68, 69, 70, 71, 72 y 73 no generan ingresos. En esta categoría podemos observar que el producto 63 es el que tiene más búsquedas de estos últimos. Por eso el producto 63 debe ser discontinuado en la tienda ya que la gente sí

lo busca pero prefiere no comprarlo. También puede prescindir de algunos otros ya que tienen una gran cantidad de productos en esta categoría.

En el caso de los USB no es recomendable quitar el producto que no se vende porque LifeStore solo tiene dos USB distintos a la venta. No sería conveniente que el consumidor no tenga la posibilidad de elegir.

En otro rubro, los productos 42 y 12 están en el top 10 de productos más vendidos y ya no tienen existencias. Entonces, es recomendable que LifeStore adquiera más de estos productos.

En cuanto a los meses, el ingreso promedio excede por una gran cantidad al ingreso de la mayoría de los meses ya que durante los primeros meses las ventas son bastante altas. Si es una tendencia constante todos los años, LifeStore debería implementar una política de descuentos durante estos meses para que sean capaces de vender un poco más durante Junio, Julio y Agosto.

Finalmente, es recomendable que la tienda deje de vender los productos con menos evaluaciones menores a 3/5. Estos productos son 45, 31, 17 y 46. Estos productos podrían estar ocasionando que los clientes se lleven una mala impresión de LifeStore. De hecho, el producto 31 es el producto que más veces ha sido devuelto.

## APÉNDICE:

Tabla de 20 productos más vendidos:

<b>Producto</b>	<b>Ventas</b>
54	50
3	42
5	20
42	18
57	15
29	14
4	13
2	13
47	11
48	9
12	9
7	7
44	6
31	6
18	5
8	4
51	3
49	3
11	3
6	3

Tabla de 20 productos menos vendidos: esta tabla será omitida ya que hay 54 productos con cero ventas.

Tabla de 20 productos más buscados:

<b>Producto</b>	<b>Búsquedas</b>
54	263
57	107
29	60
3	55
4	41
85	35
67	32
7	31
47	30
5	30
48	27
44	25
2	24
42	23
8	20
66	15
21	15
12	15
51	11
18	11

CÓDIGO:

##### INICIO DE SESIÓN

#La primera entrada es el usuario y la segunda su contraseña

```
administradores=[[ 'Ricardo', '1234'], [ 'Juan', '2345'], [ 'Sofia','3456'],  
[ 'Lorena','4567']]
```

```
#Usamos input para que ingrese un usuario y una contraseña  
usuario_entrada=input('Ingrese un usuario: ')  
contrasena_entrada=input('Contraseña: ')  
###Hacemos variables para controlar el número de intentos y también una  
dicotómica para identificar al administrador
```

```
admin=0  
intentos=0
```

```
##Utilizamos iteraciones  
for administrador in administradores:  
    if usuario_entrada==administrador[0] and  
contrasena_entrada==administrador[1]:  
        print("Bienvenido, administrador")  
        admin=1  
        if admin==1:  
            break  
    else:  
        while intentos<8 and admin==0:  
            print("Intentelo de nuevo")  
            usuario_entrada=input("Ingrese un usuario: ")  
            contrasena_entrada=input('Contraseña: ')  
            for administrador in administradores:  
                if usuario_entrada==administrador[0] and  
contrasena_entrada==administrador[1]:  
                    print("Bienvenido, administrador")  
                    admin=1  
                    if admin==1:  
                        break  
            else:  
                intentos+=1
```

```
if intentos==8:  
    print("Lo sentimos, no tienes acceso a esta información")
```

```
#Si eres administrador te preguntará sobre qué quieres saber. Primer 3  
opciones: ranking de búsquedas, ventas y valoraciones
```

```

#####Productos más buscados
#####Para acceder a a las entradas usamos [1]

#####Asignamos los números de búsqueda y el id de producto
numero_búsqueda= list(range(0,1033))
numero_producto= list(range(1,101))

búsquedas=[]
repeticiones=[]

####Usamos un for para quedarnos solo con las búsquedas, sin el id de
búsqueda
for numero in numero_búsqueda:
    búsquedas.append(lifestore_searches[numero][1])

#print(búsquedas)

#print(búsquedas.count(1))

#####Ponemos las veces que se repite cada producto
for n in numero_producto:
    repeticiones.append(búsquedas.count(n))

#####Agregaremos el id del producto
repeticiones_búsqueda_con_id=[]

for n in range(0,96):
    repeticiones_búsqueda_con_id.append([n+1, repeticiones[n]])

#####Ordenamos de menos búsqueda a más búsqueda
repeticiones_búsqueda_con_id_ordenadas=[]

```



```

#print(repeticiones_busqueda_con_id)
while repeticiones_busqueda_con_id:
    minimo=repeticiones_busqueda_con_id[0][1]
    lista_actual= repeticiones_busqueda_con_id[0]
    for repeticion in repeticiones_busqueda_con_id:
        if repeticion[1]<minimo:
            minimo=repeticion[1]
            lista_actual=repeticion
    repeticiones_busqueda_con_id_ordenadas.append(lista_actual)
    repeticiones_busqueda_con_id.remove(lista_actual)

repeticiones_busqueda_con_id_ordenadas_prime=[]
#print(repeticiones_busqueda_con_id_ordenadas)

for n in range (0,96):

    repeticiones_busqueda_con_id_ordenadas_prime.append([repeticiones_busqueda_c
on_id_ordenadas[n][1], repeticiones_busqueda_con_id_ordenadas[n][0]])

    repeticiones_busqueda_con_id_ordenadas_prime1=sorted(repeticiones_busqueda_c
on_id_ordenadas_prime, reverse=True)

    #print(repeticiones_busqueda_con_id_ordenadas_prime1)
    repeticiones_busqueda_con_id_ordenadas_prime2=[]
    for n in range (0,96):

        repeticiones_busqueda_con_id_ordenadas_prime2.append([repeticiones_busqueda_
con_id_ordenadas_prime1[n][1],
        repeticiones_busqueda_con_id_ordenadas_prime1[n][0]])

    #print(repeticiones_busqueda_con_id_ordenadas_prime2)

    #repeticiones_busqueda_con_id_ordenadas_prime2. Esta la usaremos para
    mostrar al administrador en el código

#####Usamos un while para quedarnos con las 50 más buscadas. EL producto 54
es el más buscado con 263 búsquedas
pausa=0
while pausa<48:

    repeticiones_busqueda_con_id_ordenadas.remove(repeticiones_busqueda_con_id_o
rdenadas[0])
    pausa+=1

```

```
#####Los 50 productos más buscados, en orden descendente,
son: 54, 57, 29, 3, 4, 85, 67, 7, 47, 5, 48, 44, 2, 42, 8, 66, 21, 12, 51,
18, 84, 49, 40, 31, 25, 6, 1, 89, 50, 94, 74, 52, 28, 26, 22, 11, 73, 63,
46, 15, 95, 39, 17, 91, 76, 56, 13, 93, 80, y 70
```

```
#####Los 46 productos con menos búsquedas son: 1, 16, 19,
20, 23, 24, 30, 32, 33, 34, 36, 37, 38, 41, 43, 53, 55, 58, 60, 61, 62, 64,
65, 68, 69, 71, 72, 75, 77, 78, 79, 81, 82, 83, 86, 87, 88, 90, 92 y 96
```

```
#####Productos más vendidos
```

```
numero_venta= range(0,283)
numero_producto=range(1,97)
```

```
ventas=[]
repeticiones=[]
```

```
for numero in numero_venta:
    ventas.append(lifestore_sales[numero][1])
```

```
#print(ventas)
```

```
for n in range(1, 97):
    repeticiones.append(ventas.count(n))
```

```
#print(repeticiones)
```

```
repeticiones_con_id=[]
```

```
for n in range(0,96):
    repeticiones_con_id.append([n+1, repeticiones[n]])
```

```
#####Ordenamos de menos vendido a más
repeticiones_con_id_ordenadas=[]
```

```
while repeticiones_con_id:
    minimo=repeticiones_con_id[0][1]
    lista_actual= repeticiones_con_id[0]
    for repeticion in repeticiones_con_id:
        if repeticion[1]<minimo:
```

```

        minimo=repeticion[1]
        lista_actual=repeticion
        repeticiones_con_id_ordenadas.append(lista_actual)
        repeticiones_con_id.remove(lista_actual)

repeticiones_con_id_ordenadas_prime=[]

for n in range(0,96):

repeticiones_con_id_ordenadas_prime.append([repeticiones_con_id_ordenadas[n]
[1], repeticiones_con_id_ordenadas[n][0]])

repeticiones_con_id_ordenadas_prime1=sorted(repeticiones_con_id_ordenadas_prime, reverse=True)

repeticiones_con_id_ordenadas_prime2=[]
for n in range(0,96):

repeticiones_con_id_ordenadas_prime2.append([repeticiones_con_id_ordenadas_prime1[n][1], repeticiones_con_id_ordenadas_prime1[n][0]])

#print(repeticiones_con_id_ordenadas_prime2)

#pausa=0
#while pausa<54:
# repeticiones_con_id_ordenadas.remove#(repeticiones_con_id_ordenadas[0])
# pausa+=1

#print(repeticiones_con_id_ordenadas)

#####Los 42 productos más vendidos, en orden descendente, son: 54, 3, 5,
42, 57, 29, 4, 2, 47, 48, 12, 7, 44, 31, 18, 8, 51, 49, 11, 6, 85, 74, 52,
33, 25, 21, 1, 94, 89, 84, 67, 66, 60, 50, 46, 45, 40, 28, 22, 17, 13, 10

#####Los otros 54 productos tienen 0 ventas

#####Mejores y peores reseñas:

numero_valoracion= range(0,283)
numero_producto=range(1,97)

valoraciones=[]
repeticiones=[]

for numero in numero_valoracion:

```

```

valoraciones.append([lifestore_sales[numero][1],lifestore_sales[numero][2]])

#print(valoraciones)
#####HAcemos la suma total de sus valoraciones
valoracion_total=[]
valoracion_total_producto=[]

#print(valoraciones.count([1,5]))
for i in range (1,97):
    for n in range(1,6):
        valoracion_total_producto=valoraciones.count([i,n])*n
        valoracion_total.append([i,valoracion_total_producto])

suma=[]
for n in range (0,480, 5):

suma.append(valoracion_total[n][1]+valoracion_total[n+1][1]+valoracion_total
[n+2][1]+valoracion_total[n+3][1]+valoracion_total[n+4][1])

#print(valoracion_total)
#print(suma)


for n in range(1, 97):
    repeticiones.append(ventas.count(n))

repeticiones_con_id=[]

for n in range(0,96):
    repeticiones_con_id.append([n+1, repeticiones[n]])

###Pocos productos tienen suficientes valoraciones como para saber qué opina
el público en general al adquirirla. Productos: 2, 3, 4, 29, 42. Casi
ninguno llega a 12 observaciones

```

```

promedio_valoracion=[]

for n in range(0,96):
    if repeticiones[n]!=0:
        promedio_valoracion.append(suma[n]/repeticiones[n])
    else:
        promedio_valoracion.append(0)

valoracion_promedio_conid=[]

for n in range (1,97):
    valoracion_promedio_conid.append([n, promedio_valoracion[n-1]])

#####Ordenamos de menos valorado a más valorado
valoraciones_con_id_ordenadas=[]

while valoracion_promedio_conid:
    minimo=valoracion_promedio_conid[0][1]
    lista_actual= valoracion_promedio_conid[0]
    for valoracion in valoracion_promedio_conid:
        if valoracion[1]<minimo:
            minimo=valoracion[1]
            lista_actual=valoracion
    valoraciones_con_id_ordenadas.append(lista_actual)
    valoracion_promedio_conid.remove(lista_actual)

pausa=0
while pausa<54:
    valoraciones_con_id_ordenadas.remove(valoraciones_con_id_ordenadas[0])
    pausa+=1

valoraciones_con_id_ordenadas_prime=[]
for n in range(0,42):

    valoraciones_con_id_ordenadas_prime.append([valoraciones_con_id_ordenadas[n]
[1], valoraciones_con_id_ordenadas[n][0]])

```

```

valoraciones_con_id_ordenadas_prime1=sorted(valoraciones_con_id_ordenadas_prime, reverse=True)

valoraciones_con_id_ordenadas_prime2=[]
for n in range(0,42):

valoraciones_con_id_ordenadas_prime2.append([valoraciones_con_id_ordenadas_prime1[n][1], valoraciones_con_id_ordenadas_prime1[n][0]])

#print(valoraciones_con_id_ordenadas_prime2)

####Los 42 productos con mejores valoraciones son: 85, 84, 67, 66, 60, 52, 50, 49, 40, 28, 25, 22, 21, 11, 8, 7, 6, 1 (estos tienen valoración de 5.0), 57, 3, 12, 54, 5, 51, 48, 44, 42, 47, 74, 33, 4, 18, 2, 29, 94, 13, 10, (AQUI EMPIEZAN LOS REALMENTE MAL EVALUADOS <=3), 89, 46, 31, 45, 17 (45 y 17 tienen 1, lo mínimo). El resto no tiene ninguna valoración. se obtienen con print(valoraciones_con_id_ordenadas)

#####Recordar: lifestore-products = [id_product, name, price, category, stock]

####Nos quedaremos con id, categoría y price

lista_para_revenue=[]

for n in range(0,96):
    lista_para_revenue.append([lifestore_products[n][0], lifestore_products[n][2], lifestore_products[n][3]])
###¿Cuántas y qué categorías existen?
lista_para_revenue_1=[]
for n in range(0,96):
    if lifestore_products[n][3]=='procesadores':
        lista_para_revenue_1.append([1, lifestore_products[n][0], lifestore_products[n][2]])
    elif lifestore_products[n][3]=='tarjetas de video':
        lista_para_revenue_1.append([2, lifestore_products[n][0], lifestore_products[n][2]])
    elif lifestore_products[n][3]=='tarjetas madre':
        lista_para_revenue_1.append([3, lifestore_products[n][0], lifestore_products[n][2]])
    elif lifestore_products[n][3]=='discos duros':
        lista_para_revenue_1.append([4, lifestore_products[n][0], lifestore_products[n][2]])
    elif lifestore_products[n][3]=='memorias usb':

```

```

        lista_para_revenue_1.append([5, lifestore_products[n][0],
lifestore_products[n][2]])
        elif lifestore_products[n][3]=='pantallas':
            lista_para_revenue_1.append([6, lifestore_products[n][0],
lifestore_products[n][2]])
        elif lifestore_products[n][3]=='bocinas':
            lista_para_revenue_1.append([7, lifestore_products[n][0],
lifestore_products[n][2]])
        elif lifestore_products[n][3]=='audifonos':
            lista_para_revenue_1.append([8, lifestore_products[n][0],
lifestore_products[n][2]])

```

####El orden de lista\_para\_revenue\_1 queda: categoría, id, precio. Las categorías son: 1=procesadores, 2=tarjetas de video, 3=tarjetas madre, 4=discos duros, 5=memorias usb, 6=pantallas, 7=bocinas, 8=audifonos

#Hagamos sub listas por categoria

```

procesadores=[]
tarjetas_de_video=[]
tarjetas_madre=[]
discos_duros=[]
memorias_usb=[]
pantallas=[]
bocinas=[]
audifonos=[]

```

```

for n in range(0,96):
    if lifestore_products[n][3]=='procesadores':
        procesadores.append([lifestore_products[n][0],
lifestore_products[n][2]])
    elif lifestore_products[n][3]=='tarjetas de video':
        tarjetas_de_video.append([lifestore_products[n][0],
lifestore_products[n][2]])
    elif lifestore_products[n][3]=='tarjetas madre':
        tarjetas_madre.append([lifestore_products[n][0],
lifestore_products[n][2]])
    elif lifestore_products[n][3]=='discos duros':
        discos_duros.append([lifestore_products[n][0],
lifestore_products[n][2]])
    elif lifestore_products[n][3]=='memorias usb':
        memorias_usb.append([lifestore_products[n][0],
lifestore_products[n][2]])

```

```

elif lifestore_products[n][3]=='pantallas':
    pantallas.append([lifestore_products[n][0], lifestore_products[n][2]])
elif lifestore_products[n][3]=='bocinas':
    bocinas.append([lifestore_products[n][0], lifestore_products[n][2]])
elif lifestore_products[n][3]=='audifonos':
    audifonos.append([lifestore_products[n][0], lifestore_products[n][2]])

#Saquemos el revenue vía procesadores: (usemos las repeticiones de venta de
cada producto, repeticiones_con_id). La primera entrada de las categorías es
el id y la segunda es el precio. Hay nueve productos en la categoría
procesadores
len(procesadores)

revenue_por_procesador=[]

for n in range (0,9):
    revenue_por_procesador.append([procesadores[n][0],
procesadores[n][1]*repeticiones_con_id[n][1]])

#print(revenue_por_procesador). 1: 6038, 2:54717, 3: 129738, 4: 28717, 5:
35580, 6: 35427, 7: 59913, 8:21596, 9:0

revenue_total_procesador=
revenue_por_procesador[0][1]+revenue_por_procesador[1][1]+revenue_por_procesador[2][1]+revenue_por_procesador[3][1]+revenue_por_procesador[4][1]+revenue_por_procesador[5][1]+revenue_por_procesador[6][1]+revenue_por_procesador[7][1]+revenue_por_procesador[8][1]

#print(revenue_total_procesador)=371726

###Ahora tarjetas de video. len(tarjetas_de_video)=19
revenue_por_tarjeta_v=[]

for n in range (0,19):
    revenue_por_tarjeta_v.append([tarjetas_de_video[n][0],
tarjetas_de_video[n][1]*repeticiones_con_id[n+9][1]])

#print(revenue_por_tarjeta_v). 10: 889, 11:22197, 12:59571, 13: 3989, 14:0,
15:0, 16:0, 17:4199, 18:10995, 19:0, 20:0, 21:10318, 22:3429, 23:0, 24:0,
25:11058, 26:0, 27:0, 28:9579

revenue_total_tarjeta_v=0

for n in range(0,19):

```



```

    revenue_total_tarjeta_v+=
revenue_total_tarjeta_v+revenue_por_tarjeta_v[n][1]

#REVENUE POT TARJETAS DE VIDEO
#print(revenue_total_tarjeta_v)=136224

##Ahora tarjetas madre. print(len(tarjetas_madre))=18

revenue_por_tarjeta_m=[]

for n in range (0,18):
    revenue_por_tarjeta_m.append([tarjetas_madre[n][0],
tarjetas_madre[n][1]*repeticiones_con_id[n+28][1]])

#print(revenue_por_tarjeta_m). 29=34986, 30=0, 31=13374, 32=0, 33=8538,
34=0, 35=0, 36=0, 37=0, 38=0, 39=0, 40=17439, 41=0, 42=32022, 43=0,
44=16554, 45=2869, 46=1539. Podríamos reducir la variedad de tarjetas madre,
muchos ceros

revenue_total_tarjeta_m=0

for n in range(0,18):
    revenue_total_tarjeta_m+=revenue_por_tarjeta_m[n][1]

#print(revenue_total_tarjeta_m)=127321. REVENUE POR TARJETAS MADRE

#Ahora discos duros .print(len(discos_duros))=13
revenue_por_discos=[]

for n in range (0,13):
    revenue_por_discos.append([discos_duros[n][0],
discos_duros[n][1]*repeticiones_con_id[n+46][1]])

#print(revenue_por_discos). 47=13299, 48=23031, 49=9417, 50=2949, 51=7197,
52=11318, 53=0, 54=12950, 55=0, 56=0, 57=13335, 58=0, 59=0

revenue_total_discos=0

for n in range(0,13):
    revenue_total_discos+= revenue_total_discos+revenue_por_discos[n][1]

#print(revenue_total_discos)=93496. REVENUE TOTAL DE DISCOS DUROS

#Ahora usb. print(len(memorias_usb))=2

```

```

revenue_por_usb=[]

for n in range (0,2):
    revenue_por_usb.append([memorias_usb[n][0],
memorias_usb[n][1]*repeticiones_con_id[n+59][1]])

#print(revenue_por_usb). 60=2519, 61=0

revenue_total_usb=0

for n in range(0,2):
    revenue_total_usb+= revenue_total_usb+revenue_por_usb[n][1]

#print(revenue_total_usb)=2519. REVENUE TOTAL POR usb

#Ahora pantallas. print(len(pantallas))=12

revenue_por_pantalla=[]

for n in range (0,12):
    revenue_por_pantalla.append([pantallas[n][0],
pantallas[n][1]*repeticiones_con_id[n+61][1]])

#print(revenue_por_pantalla). 62=0, 63=0, 64=0, 65=0, 66=0, 66=8049,
67=3229, 68=0, 69=0, 70=0, 71=0, 72=0, 73=0. Otra vez hay muchos productos
con ceor revenue generado.

revenue_total_pantallas=0

for n in range(0,12):
    revenue_total_pantallas+=
revenue_total_pantallas+revenue_por_pantalla[n][1]

#print(revenue_total_pantallas)=11278. REVENUE TOTAL POR pantallas

#Ahora bocinas. print(len(bocinas))=10

revenue_por_bocina=[]

for n in range (0,10):
    revenue_por_bocina.append([bocinas[n][0],
bocinas[n][1]*repeticiones_con_id[n+73][1]])

```

```

#print(revenue_por_bocina). 74:8478, 75=0, 76=0, 77=0, 77=0, 78=0, 79=0,
80=0, 81=0, 82=0, 83=0. Otra vez tenemos muchos ceros

revenue_total_bocinas=0

for n in range(0,10):
    revenue_total_bocinas+= revenue_total_bocinas+revenue_por_bocina[n][1]

#print(revenue_total_bocinas)=8478. Solo se vende una bocina, la 74. REVENUE
TOTAL BOCINAS

#Ahora audifonos. print(len(audifonos))=13

revenue_por_audifono=[]

for n in range (0,13):
    revenue_por_audifono.append([audifonos[n][0],
audifonos[n][1]*repeticiones_con_id[n+83][1]])

#print(revenue_por_audifono). 84=1089, 85=4318, 86=0, 87=0, 88=0, 89=859,
90=0, 91=0, 92=0, 93=0, 94=2869, 95=0, 96=0. También hay demasiados ceros

revenue_total_audifonos=0

for n in range(0,13):
    revenue_total_audifonos+=
revenue_total_audifonos+revenue_por_audifono[n][1]

#print(revenue_total_audifonos)=9135

###¿Cuál es el revenue total de la tienda?

revenue_total_tienda=
revenue_total_procesador+revenue_total_tarjeta_v+revenue_total_tarjeta_m+rev
enue_total_discos+revenue_total_usb+revenue_total_pantallas+revenue_total_bo
cinas+revenue_total_audifonos

#print(revenue_total_tienda)=770619. REVENUE TOTAL DE LA TIENDA

####Ventas por mes

lifestore_sales_fecha=[]
#####Reordenamos para que quede primero el año y sort funcione
correctamente
for n in range (0,283):

```

```

    lifestore_sales_fecha.append([lifestore_sales[n][3][6] +
lifestore_sales[n][3][7] + lifestore_sales[n][3][8] +
lifestore_sales[n][3][9] + lifestore_sales[n][3][2] +
lifestore_sales[n][3][3] + lifestore_sales[n][3][4] +
lifestore_sales[n][3][2] + lifestore_sales[n][3][0] +
lifestore_sales[n][3][1], lifestore_sales[n][1]])

fecha_ordenada=sorted(lifestore_sales_fecha)

#print(fecha_ordenada)

fecha_ordenada_idventa=[]
for n in range(0, 283):
    fecha_ordenada_idventa.append([fecha_ordenada[n][0], fecha_ordenada[n][1],
n+1])

#Ventas en enero. Son de la primera observacion a la 55

venta_enero=[]

for n in range(0,55):
    venta_enero.append(fecha_ordenada_idventa[n][1])

#Hay que ordenarlo print(venta_enero).

venta_enero_ordenada= sorted(venta_enero)

#print(venta_enero_ordenada)
productos_enero=[]
for n in range (1,97):
    productos_enero.append([n, venta_enero_ordenada.count(n),
lifestore_products[n-1][2]])

#print(productos_enero). Su estructura es: id producto, cantidad vendida en
enero, precio del productos_enero

revenue_producto_enero=[]

for n in range (0, 96):
    revenue_producto_enero.append([n+1,
productos_enero[n][2]*productos_enero[n][1]])

```

```
#print(revenue_producto_enero). El podructo 3 sobresale de los demás. El 2, 8 y 29 también se venden bastante
```

```
revenue_total_enero=0
```

```
for n in range(0, 96):  
    revenue_total_enero+= revenue_producto_enero[n][1]
```

```
#print(revenue_producto_enero)  
#print(revenue_total_enero)=124705. REVENUE TOTAL ENERO
```

```
#Hagamos febrero. print(fecha_ordenada_idventa). Llega hasta la 96
```

```
venta_febrero=[]
```

```
for n in range(0,41):  
    venta_febrero.append(fecha_ordenada_idventa[n+55][1])
```

```
#Hay que ordenarlo print(venta_febrero).
```

```
venta_febrero_ordenada= sorted(venta_febrero)
```

```
#print(venta_febrero_ordenada)  
productos_febrero=[]  
for n in range (1,97):  
    productos_febrero.append([n, venta_febrero_ordenada.count(n),  
lifestore_products[n-1][2]])
```

```
#print(productos_febrero). Su estructura es: id producto, cantidad vendida  
en enero, precio del productos_febrero
```

```
revenue_producto_febrero=[]
```

```
for n in range (0, 96):  
    revenue_producto_febrero.append([n+1,  
productos_febrero[n][2]*productos_febrero[n][1]])
```

```
#print(revenue_producto_febrero). El podructo 11 sobresale de los demás. El 3, 74 y 7 también se venden bastante
```

```
revenue_total_febrero=0
```

```
for n in range(0, 96):  
    revenue_total_febrero+= revenue_producto_febrero[n][1]
```

```

#print((revenue_total_febrero))=110139. REVENUE TOTAL FEBRERO

#Marzo print(fecha_ordenada_idventa). Llega hasta la 147

venta_marzo=[]

for n in range(0,51):
    venta_marzo.append(fecha_ordenada_idventa[n+96][1])

#Hay que ordenarlo print(venta_marzo).

venta_marzo_ordenada= sorted(venta_marzo)

#print(venta_marzo_ordenada)
productos_marzo=[]
for n in range (1,97):
    productos_marzo.append([n, venta_marzo_ordenada.count(n),
lifestore_products[n-1][2]])

#print(productos_marzo). Su estructura es: id producto, cantidad vendida en
marzo, precio del productos_marzo

revenue_producto_marzo=[]

for n in range (0, 96):
    revenue_producto_marzo.append([n+1,
productos_marzo[n][2]*productos_marzo[n][1]])

#print(revenue_producto_marzo). El podructo 11 sobresale de los demás. El 3,
74 y 7 también se venden bastante

revenue_total_marzo=0

for n in range(0, 96):
    revenue_total_marzo+= revenue_producto_marzo[n][1]

#print(revenue_producto_marzo). 2 3 Y 38 SOBRESALEN
#print(revenue_total_marzo)=164729. REVENUE TOTAL DE MARZO

#Abril print(fecha_ordenada_idventa). Llega hasta la 222

venta_abril=[]

for n in range(0,75):
    venta_abril.append(fecha_ordenada_idventa[n+147][1])

```

```

#Hay que ordenarlo print(venta_abril).

venta_abril_ordenada= sorted(venta_abril)

#print(venta_abril_ordenada)
productos_abril=[]
for n in range (1,97):
    productos_abril.append([n, venta_abril_ordenada.count(n),
lifestore_products[n-1][2]])

#print(productos_abril). Su estructura es: id producto, cantidad vendida en
marzo, precio del productos_marzo

revenue_producto_abril=[]

for n in range (0, 96):
    revenue_producto_abril.append([n+1,
productos_abril[n][2]*productos_abril[n][1]])

#print(revenue_producto_abril). El podructo 11 sobresale de los demás. El 3,
74 y 7 también se venden bastante

revenue_total_abril=0

for n in range(0, 96):
    revenue_total_abril+= revenue_producto_abril[n][1]

#print(revenue_producto_abril). 3, 4 Y 29
#print(revenue_total_abril)=193295. REVENUE TOTAL EN ABRIL

#Mayo llega hasta la 257. print(fecha_ordenada_idventa)

venta_mayo=[]

for n in range(0,35):
    venta_mayo.append(fecha_ordenada_idventa[n+222][1])

#Hay que ordenarlo print(venta_mayo).

venta_mayo_ordenada= sorted(venta_mayo)

#print(venta_mayo_ordenada)
productos_mayo=[]
for n in range (1,97):

```

```

    productos_mayo.append([n, venta_mayo_ordenada.count(n),
lifestore_products[n-1][2]])

#print(productos_mayo). Su estructura es: id producto, cantidad vendida en
mayo, precio del productos_mayo

revenue_producto_mayo=[]

for n in range (0, 96):
    revenue_producto_mayo.append([n+1,
productos_mayo[n][2]*productos_mayo[n][1]])

#print(revenue_producto_mayo). El podructo 11 sobresale de los demás. El 3,
74 y 7 también se venden bastante

revenue_total_mayo=0

for n in range(0, 96):
    revenue_total_mayo+= revenue_producto_mayo[n][1]

#print(revenue_producto_mayo). 2, 3 y 6
#print(revenue_total_mayo)=96135. REVENUE TOTAL MAYO

#Junio.print(fecha_ordenada_idventa) llega hasta la 268

venta_junio=[]

for n in range(0,11):
    venta_junio.append(fecha_ordenada_idventa[n+257][1])

#Hay que ordenarlo print(venta_junio).

venta_junio_ordenada= sorted(venta_junio)

#print(venta_mayo_ordenada)
productos_junio=[]
for n in range (1,97):
    productos_junio.append([n, venta_junio_ordenada.count(n),
lifestore_products[n-1][2]])

#print(productos_junio). Su estructura es: id producto, cantidad vendida en
junio, precio del productos_junio

revenue_producto_junio=[]

```



```

for n in range (0, 96):
    revenue_producto_junio.append([n+1,
    productos_junio[n][2]*productos_junio[n][1]])

#print(revenue_producto_junio). El podructo 11 sobresale de los demás. El 3,
74 y 7 también se venden bastante

revenue_total_junio=0

for n in range(0, 96):
    revenue_total_junio+= revenue_producto_junio[n][1]

#print(revenue_total_junio)=36959. REVENUE TOTAL JUNIO
###print(revenue_producto_junio). 11, 2 y 3

#Julio print(venta_ordenada_idventa). Llega hasta la 279

venta_julio=[]

for n in range(0,11):
    venta_julio.append(venta_ordenada_idventa[n+268][1])

#Hay que ordenarlo print(venta_julio).

venta_julio_ordenada= sorted(venta_julio)

#print(venta_julio_ordenada)
productos_julio=[]
for n in range (1,97):
    productos_julio.append([n, venta_julio_ordenada.count(n),
    lifestore_products[n-1][2]])

#print(productos_julio). Su estructura es: id producto, cantidad vendida en
julio, precio del productos_julio

revenue_producto_julio=[]

for n in range (0, 96):
    revenue_producto_julio.append([n+1,
    productos_julio[n][2]*productos_julio[n][1]])

#print(revenue_producto_junio). El podructo 11 sobresale de los demás. El 3,
74 y 7 también se venden bastante

revenue_total_julio=0

```

```

for n in range(0, 96):
    revenue_total_julio+= revenue_producto_julio[n][1]

#print(revenue_producto_julio). 1 y 3
#print(revenue_total_julio)=26949. REVENUE TOTAL JULIO

#Agosto. hasta la 282

venta_agosto=[]

for n in range(0,3):
    venta_agosto.append(fecha_ordenada_idventa[n+279][1])

#Hay que ordenarlo print(venta_agosto).

venta_agosto_ordenada= sorted(venta_agosto)

#print(venta_agosto_ordenada)
productos_agosto=[]
for n in range (1,97):
    productos_agosto.append([n, venta_agosto_ordenada.count(n),
lifestore_products[n-1][2]])

#print(productos_agosto). Su estructura es: id producto, cantidad vendida en
agosto, precio del productos_agosto

revenue_producto_agosto=[]

for n in range (0, 96):
    revenue_producto_agosto.append([n+1,
productos_agosto[n][2]*productos_agosto[n][1]])

revenue_total_agosto=0

for n in range(0, 96):
    revenue_total_agosto+= revenue_producto_agosto[n][1]

#print(fecha_ordenada_idventa)
#print(revenue_producto_agosto). Solo se vendieron 48 y 54
#print(revenue_total_agosto)=3077

```

```

#Septiembre. #Solo han vendido un producto 17

revenue_total_septiembre=4199

revenue_anual=revenue_total_enero+revenue_total_febrero+revenue_total_marzo+
revenue_total_abril+revenue_total_mayo+revenue_total_junio+revenue_total_jul
io+revenue_total_agosto+revenue_total_septiembre

#print(revenue_anual)
#print(revenue_total_tienda) Hacen match: 760177

rep=0
for n in range(0,96):
    rep=rep+repeticiones_con_id[n][1]

rev=0

for n in range(0,96):
    rev+= repeticiones_con_id[n][1]*lifestore_products[n][2]

###Para sacar las ventas promedio mensuales dividiremos el total de ventas
entre 8 ya que solo tenemos observaciones de enero a agosto y una de
septiembre. Restamos el dato de septiembre

ventas_promedio_mensuales=(revenue_anual-revenue_total_septiembre)/12

#print(ventas_promedio_mensuales)=62998.16

##Devoluciones
devoluciones=[]
for n in range(0,283):
    if lifestore_sales[n][4]==1:
        devoluciones.append(lifestore_sales[n][1])

#print(devoluciones)

devoluciones_id=[]

for n in range(0,96):
    devoluciones_id.append([n+1, devoluciones.count(n+1)])

#print(devoluciones_id). Solo hay 9 devoluciones y el 31 es el unico que han
devuelto más de una vez
valor_de_devoluciones=0

```

```

for n in range(0,96):
    valor_de_devoluciones+= devoluciones_id[n][1]*lifestore_products[n][2]
#print(valor_de_devoluciones)=22261

####Productos en inventarios con id

inventario_porid=[]
for n in range(0,96):
    inventario_porid.append([lifestore_products[n][4], n+1])

#print(inventario_porid)

#####Ordenamos por productos con más stock a menos stock

inventario_porstock=sorted(inventario_porid, reverse=True)

#print(inventario_porstock)

inventario_prime=[]
for n in range(0,96):
    inventario_prime.append([inventario_porstock[n][1],
inventario_porstock[n][0]])

inventario_porid_prime=[]
for n in range (0,96):
    inventario_porid_prime.append([inventario_porid[n][1],
inventario_porid[n][0]])

#print(inventario_porstock)
#Los 10 productos con más stock son: 3,67,54,4,41,68,92,66,2,26
#Los 11 productos con menos stock son: 12, 21, 22, 42, 44, 51, 13, 17,
40,53,74
if admin==1:
    ver_mas=input("Elige una opción: Busquedas/Ventas/Valoraciones/Inventario:
")
    if ver_mas=="Inventario":
        inv=input("Elige: producto/rango: ")
        if inv=="producto":
            producto=int(input("Elige un producto (número): "))
            print(lifestore_products[producto-1][4])
        elif inv=="rango":
            eleccion=input("Elige: mas stock/menos stock: ")
            if eleccion=="mas stock":

```

```

        rango=int(input("Elige el rango (1 dará el producto con más stock,
20 dará los 20 con más stock): "))
        for n in range(0,rango):
            print(inventario_prime[n])
        elif eleccion=="menos stock":
            rango=int(input("Elige el rango (1 dará el producto con menos stock,
20 dará los 20 con menos stock): "))
            for n in range(0,rango):
                print(inventario_prime[95-n])
    elif ver_mas=="Busquedas":
        buscar=input("Elige: mas buscados/menos buscado: ")
        if buscar=="mas buscados":
            rango=int(input("Elige el rango (1 dará el más buscado, 20 dará los 20
más buscados): "))

            for n in range(0, rango):
                print(repeticiones_busqueda_con_id_ordenadas_prime2[n])
        elif buscar=="menos buscados":
            rango=int(input("Elige el rango (1 dará el menos buscado, 20 dará los
20 menos buscados): "))

            for n in range(0, rango):
                print(repeticiones_busqueda_con_id_ordenadas_prime2[95-n])
    elif ver_mas=="Ventas":
        vender=input("Elige una opción: mas vendidos/menos
vendidos/categorias/meses/ventas totales ")
        if vender=="ventas totales":
            print(revenue_anual)
        if vender=="mas vendidos":
            rango=int(input("Elige el rango(1 dará el más vendido, 20 dará los 20
más vendidos: "))
            for n in range(0, rango):
                print(repeticiones_con_id_ordenadas_prime2[n])
        elif vender=="menos vendidos":
            rango=int(input("Elige el rango(1 dará el menos vendido, 20 dará los
20 menos vendidos: "))
            for n in range(0, rango):
                print(repeticiones_con_id_ordenadas_prime2[95-n])
        elif vender=="categorias":
            cate=input("Elige una categoría: Procesadores/Tarjetas de
video/Tarjetas madre/Discos duros/USB/Pantallas/Bocinas/Audifonos: ")
            if cate=="Procesadores":
                print("Los ingresos por procesadores son:
"+"$"+str(revenue_total_procesador))
            elif cate=="Tarjetas de video":

```

```

        print("Los ingresos por tarjetas de video son:
"+"$"+str(revenue_total_tarjeta_v))
        elif cate=="Tarjetas madre":
            print("Los ingresos por tarjetas madre son:
"+"$"+str(revenue_total_tarjeta_m))
            elif cate=="Discos duros":
                print("Los ingresos por discos duros son:
"+"$"+str(revenue_total_discos))
                elif cate=="USB":
                    print("Los ingresos por USB son: "+"$"+str(revenue_total_usb))
                    elif cate=="Pantallas":
                        print("Los ingresos por pantallas son:
"+"$"+str(revenue_total_pantallas))
                        elif cate=="Bocinas":
                            print("Los ingresos por bocinas son:
"+"$"+str(revenue_total_bocinas))
                            elif cate=="Audifonos":
                                print("Los ingresos por bocinas son:
"+"$"+str(revenue_total_audifonos))
                                elif vender=="meses":
                                    mes=input("Elija un mes para ver los ingresos en ese periodo(empieza
la palabra con mayúscula: ")
                                    if mes=="Enero":
                                        print("Los ingresos en el mes de enero fueron:
"+"$"+str(revenue_total_enero))
                                        elif mes=="Febrero":
                                            print("Los ingresos en el mes de febrero fueron:
"+"$"+str(revenue_total_febrero))
                                            elif mes=="Marzo":
                                                print("Los ingresos en el mes de marzo fueron:
"+"$"+str(revenue_total_marzo))
                                                elif mes=="Abril":
                                                    print("Los ingresos en el mes de abril fueron:
"+"$"+str(revenue_total_abril))
                                                    elif mes=="Abril":
                                                        print("Los ingresos en el mes de abril fueron:
"+"$"+str(revenue_total_abril))
                                                        elif mes=="Mayo":
                                                            print("Los ingresos en el mes de mayo fueron:
"+"$"+str(revenue_total_mayo))
                                                            elif mes=="Junio":
                                                                print("Los ingresos en el mes de junio fueron:
"+"$"+str(revenue_total_junio))
                                                                elif mes=="Julio":

```

```

        print("Los ingresos en el mes de julio fueron:
"+"$"+str(revenue_total_julio))
        elif mes=="Agosto":
            print("Los ingresos en el mes de agosto fueron:
"+"$"+str(revenue_total_agosto))
            elif mes=="Septiembre":
                print("Los ingresos en el mes de septiembre fueron:
"+"$"+str(revenue_total_septiembre))
            else:
                print("Este dato no existe")
    if ver_mas=="Valoraciones":
        valorar=input("Elige: mas valorados/menos valorados: ")
        if valorar=="mas valorados":
            rango=int(input("Elige el rango (1 dará el más valorado, 20 dará los
20 más valorados): "))
            if rango>42:
                print("Solo hay 42 productos")

        for n in range(0, rango):
            print(valoraciones_con_id_ordenadas_prime2[n])
        elif valorar=="menos valorados":
            rango=int(input("Elige el rango (1 dará el menos valorado, 20 dará los
20 menos valorados): "))

        for n in range(0, rango):
            print(valoraciones_con_id_ordenadas_prime2[41-n])

```