

Análisis de ventas para LifeStore

Proyecto 1, Introducción a Python, EMTECH

Tabla de contenido

- [Introducción: Problema de negocio](#)
- [Datos](#)
- [Metodología](#)
- [Análisis](#)
- [Solución al problema](#)
- [Conclusión](#)
- [Anexo - Descripción del código](#)

Introducción: Problema de negocio

LifeStore es una tienda virtual que maneja una amplia gama de artículos, principalmente en el sector de artículos para computadoras. La Gerencia de Ventas se percató que la empresa tiene una acumulación de inventario, también notó que ha tenido reducciones de ventas en determinados artículos.

Nuestro trabajo será realizar un análisis de los productos, tratando de identificar aspectos importantes, para poder ofrecerle una solución a LifeStore

Datos

La información que se nos provee es la siguiente:

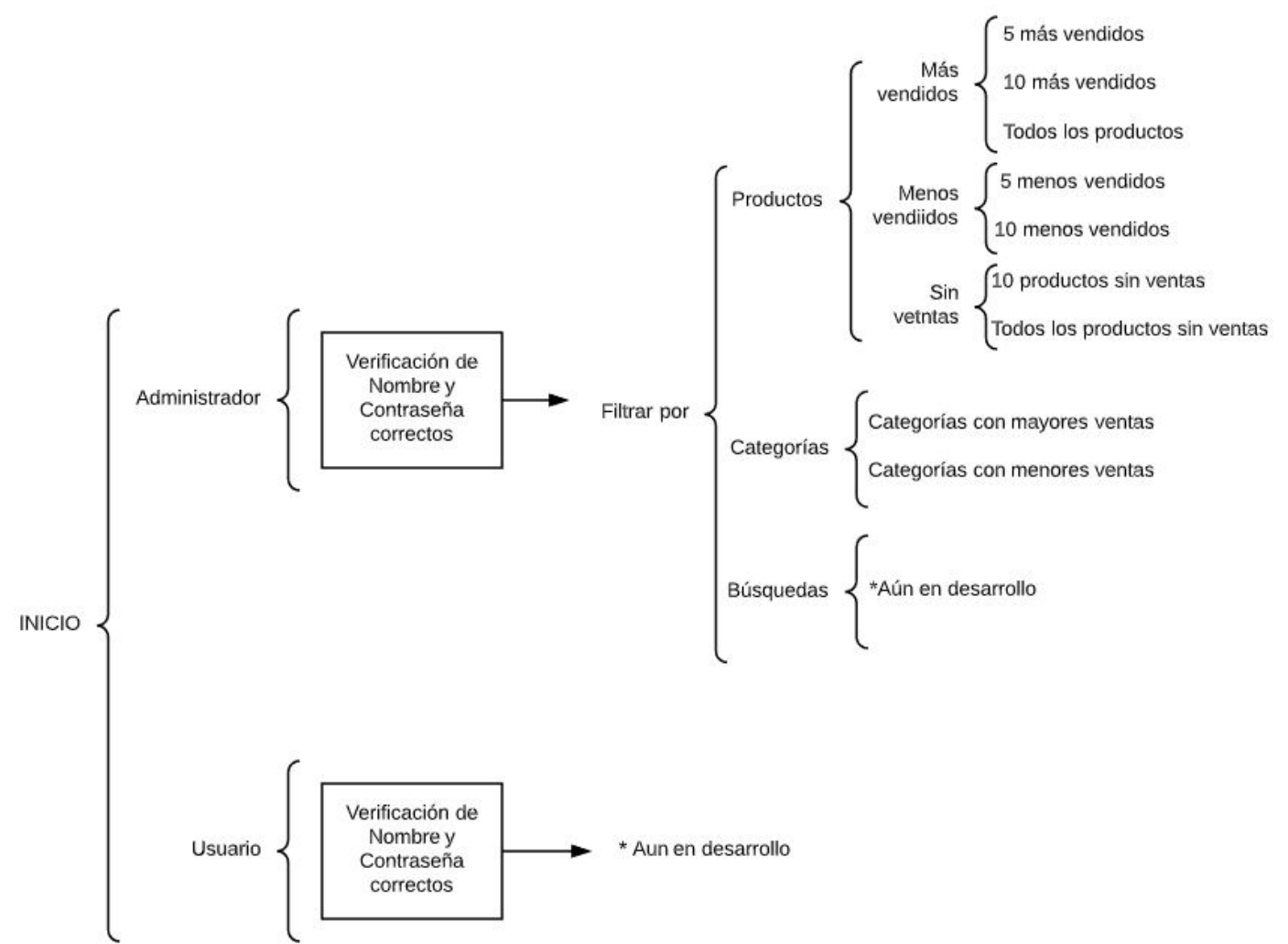
lifestore-products , esta lista contiene 92 registros, cada uno contiene información sobre el identificador de cada producto, nombre, precio, categoría y stock.

lifestore-sales , esta lista contiene 283 registros, cada uno contiene información sobre el identificador de cada venta, identificador del producto, puntaje, fecha y reembolso.

lifestore-searches , esta lista contiene 1033 registros, cada uno con información sobre el identificador de la búsqueda y el identificador del producto.

Metodología

La forma en cómo le vamos a permitir al Usuario o Administrador realizar alguna consulta se muestra en el siguiente diagrama.



Análisis

Ejecutando el código, filtrando y realizando las distintas consultas que consideramos importantes, obtuvimos los siguientes resultados.

Filtrando por productos

Vemos que algunos de los productos con **mayor cantidad de ventas** son:

"SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm"
 "Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth"
 "Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)"
 "Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD"
 "SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm"

*Es decir, discos **SSD** de entre 120 GB y 256 GB, así como procesadores **AMD Ryzen 5** y **Core i3 de 9na generación**.*

Analizando los productos con **menores ventas**:

"TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Widescreen, Negro"
 "TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro"
 "Logitech Audífonos Gamer G332, Alámbrico, 2 Metros, 3.5mm, Negro/Rojo"
 "Cougar Audífonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro."
 "HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro, Inalámbrico, USB, 3.5mm, Negro"

*Notamos que las **televisiones** y **audífonos** son productos con pocas ventas*

Pero incluso hay productos que **no han tenido ventas**:

'Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)'
 'Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB 64-bit GDDR3, PCI Express 2.0'
 'Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0'
 'Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming, 6GB 192-bit GDDR6, PCI Express 3.0'
 'Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile, 4GB 128-bit GDDR5, PCI Express 3.0 x16'
 'Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC, 8 GB 256 bit GDDR6, PCI Express x16 3.0'
 'Tarjeta de Video MSI Radeon X1550, 128MB 64 bit GDDR2, PCI Express x16']
 'Tarjeta de Video PNY NVIDIA GeForce RTX 2080, 8GB 256-bit GDDR6, PCI Express 3.0\xa0'
 'Tarjeta de Video VisionTek AMD Radeon HD 5450, 1GB DDR3, PCI Express x16 2.1'
 'Tarjeta de Video VisionTek AMD Radeon HD5450, 2GB GDDR3, PCI Express x16'

*Podemos afirmar que los procesadores **Core i3 de 8va generación** son productos que la gente ya no está comprando, y sobre todo notamos que hay **tarjetas de video NVIDIA**, no hay tenido ventas*

Filtrando por categoría

*Al realizar este filtro, nos damos cuenta de que las categorías con mayores ventas son los **procesadores** con 104 ventas, seguido de los **discos duros** con 94 ventas, sin embargo las **tarjetas madre** hay tenido 49 ventas.*

```
'procesadores', 104 ventas  
'discos duros', 94 ventas  
'tarjetas madre', 49 ventas
```

Por otro lado las categorías como **audífonos**, **bocinas**, **pantallas** y **memorias usb** son categorías que tienen registrada la menor cantidad de ventas.

```
'audifonos', 5 ventas  
'bocinas', 2 ventas  
'pantallas', 2 ventas  
'memorias usb', 1 venta
```

Solución al problema

Tener en cuenta que las categorías de *audífonos*, *bocinas*, *pantallas* y *memorias usb* son categorías con pocas ventas, por lo tanto, lo que le recomendamos a LifeStore es **no adquirir más stock** de productos relacionados con éstas categorías.

También tomar en cuenta **no adquirir más stock de procesadores Core i3 de 8va generación, ni tarjetas de video NVIDIA**, ya que son productos que no se venden.

Sin embargo, le recomendamos seguir invirtiendo en adquirir stock de **discos duros SSD de 120 GB y 240GB**, así como procesadores **AMD Ryzen y Core i3 de 9na generación**, ya que son bien aceptados por los consumidores.

Conclusión

Respecto a LifeStore

Si pudiéramos tener disponibles registros de nuestros compradores, como su edad, su género, su ocupación, su grado de estudios, podríamos identificar patrones, para determinar con más detalle el grupo de clientes que tenemos, una vez identificando algunas de las características de nuestros compradores, podríamos agregar stock que aún no tiene la tienda, pero que en base el perfil de nuestros compradores, seguro les podrían interesar.

Respecto al proyecto general

Considero que fue un gran reto, ya que usualmente estamos acostumbrados a utilizar librerías para ahorrarnos trabajo al momento de realizar nuestro código, sin embargo como en este proyecto no se nos permitió utilizar librerías ni algunas funciones complejas, tuvimos que pensar cómo realizar el conteo de una lista, cómo realizar el ordenamiento, factores, variables, banderas, contadores que tuvimos que tener en cuenta para obtener el resultado deseado. Esto nos da una perspectiva de lo que pasa detrás de cada función, inclusive detrás de cada *sort()*, detrás de cada *count()*, y demás funciones.

Anexo - Descripción del código

(Este es exactamente el mismo código que esta en repl.it, unicamente que aquí está desglosado tratando de explicar cada parte del código)

Importamos las lista que se nos proporcionan en el archivo lifestore_file

```
In [6]: from lifestore_file import lifestore_products
        from lifestore_file import lifestore_sales
        from lifestore_file import lifestore_searches
```

Registro que contiene los datos de los usuarios y administradores registrados con sus contraseñas respectivamente

```
In [ ]: usuarios_pass = [['Juan', '123'], ['Karla', 'sol'], ['Enrique', 'lapis']]
        administradores_pass = [['Oscar', 'siete'], ['Kassandra', 'libreta']]
```

En esta parte pedimos que se indique si es usuario o administrador, y que ingrese sus credenciales, nombre y contraseña, en caso de que sí estén registrados, les mostramos un mensaje **"Acceso Permitido"**

```

In [ ]: print('Bienvenido a Lifestore:')
        user_or_admin = 0
        es_usuario = 0
        es_admin = 0
        salir = 0

        # Oscanwhile (user_or_admin != '1' or user_or_admin != '2') and (es_usuario == es_admin):
        while salir == 0:
            user_or_admin = input('Seleccione: 1 si es Usuario, 2 si es Administrador ')
            if user_or_admin == '1':
                print('\n    Usted intenta acceder como Usuario')
                es_usuario = 1
                es_admin = 0
                salir = 1
            elif user_or_admin == '2':
                print('\n    Usted intenta acceder como Administrador')
                es_usuario = 0
                es_admin = 1
                salir = 1
            else:
                print('        ¡ Selección no válida intente de nuevo !')
                salir = 0

        if es_usuario == 1:
            usuario_registrado = 0
            while usuario_registrado == 0:
                user_id = input('Ingrese su nombre de Usuario: ')
                user_pass = input('Ingrse su contraseña de Usuario: ')
                for i in usuarios_pass:
                    if user_id == i[0] and user_pass == i[1]:
                        usuario_registrado = 1
                        administrador_registrado = 0
                        print('\n    Acceso Permitido como Usuario')
                        print(' \n    ¡ Opciones para consulta de usuario aún no disponibles, \n
estamos trabajando en ello !')
                        consulta_de_administrador = []
                        break
                    if usuario_registrado == 0:
                        print(' \n        ¡ Credenciales no válidas, ingrese nuevamente sus datos !')
            else:
                administrador_registrado = 0
                contador_para_ubicar_nombre = []
                while administrador_registrado == 0:
                    admin_id = input('Ingrese su nombre de Administrador: ')
                    admin_pass = input('Ingrese su contraseña de Administrador: ')
                    for i in administradores_pass:
                        contador_para_ubicar_nombre.append(1)
                        if admin_id == i[0] and admin_pass == i[1]:
                            administrador_registrado = 1
                            usuario_registrado = 0
                            print('\n    Acceso Permitido como Administrador')

                            break
                    if administrador_registrado == 0:
                        print(' \n        ¡ Credenciales no válidas, ingrese nuevamente sus datos !')
                        contador_para_ubicar_nombre = [] # Con este contador sabemos en qué posició
de la lista se encuentra, y por ende sabemos quien esta ingresando.

        if administrador_registrado == 1:
            print('    Hola', administradores_pass[len(contador_para_ubicar_nombre)-1][0], 'bienven
ido.\n')

```

Hasta aqui ya accedió el usuario o el administrador, lo que se hace ahora es preguntar por la consulta (es decir FILTRAR)

```

In [ ]: preguntar_consulta = 1 # Variable para entrar en un WHILE para pedir una consulta
if administrador_registrado == 1: # Esta variables solo se pone a 1 cuando el nombre y c
ontraseña introducidas son correctas
    while preguntar_consulta == 1: #Esta variable nos indica si ya se especificó una con
sulta
        # Preguntamos si se quiere filtrar por PRODUCTOS, o CATEGORÍAS, o BÚSQUEDAS
        consulta_de_administrador = input('Filtrar búsqueda por:\n    1 - Productos \n
2 - Categorías \n    3 - Búsquedas \n')

        # -----Condición para filtrar por Productos--
        -----

        # Aquí pedimos indicar si quiere filtrar por los más vendidos, Los menos vendido
s o sin ventas

        if consulta_de_administrador == '1':
            salir = 0
            while salir == 0: #Cuando se indique una opción permitida ponemos esta variab
le a 1 para dejar de preguntar.
                filtro_por_producto = input('Seleccionó filtrar por productos, indique:\n
1 - Productos más vendidos\n    2 - Productos menos vendidos\n    3 - Productos sin vent
as\n')
                if filtro_por_producto == '1': #Bandera para mostrar Los productos más vendi
dos

                    print('Seleccionó filtrar por los productos más vendidos')
                    productos_a_mostrar = input('    Indique el número de productos que dese
a visualizar. \n a - Visualizar los 5 productos más vendidos. \n b - Visualizar los 10 p
roductos más vendidos. \n c - Visualizar todos los productos \n')
                    if productos_a_mostrar == 'a' or productos_a_mostrar == 'b' or productos
_a_mostrar == 'c':
                        preguntar_consulta = 0 #AL poner esta variable a cero, se saldrá del c
iclo while
                        salir = 1
                    else:
                        print(' ¡ Opción no válida, vuelva a intentar !')
                elif filtro_por_producto == '2': #Bandera para mostrar Los productos menos v
endidos

                    print('Seleccionó filtrar por los productos menos vendidos')
                    productos_a_mostrar = input('    Indique el número de productos que dese
a visualizar. \n a - Visualizar los 5 productos menos vendidos. \n b - Visualizar los 10
productos menos vendidos\n')
                    if productos_a_mostrar == 'a' or productos_a_mostrar == 'b':
                        preguntar_consulta = 0 #AL poner esta variable a cero, se saldrá del c
iclo while
                        salir = 1
                    else:
                        print(' ¡ Opción no válida, vuelva a intentar !')
                elif filtro_por_producto == '3': #Bandera para mostrar Los productos sin ven
tas

                    print('Seleccionó filtrar por los productos sin ventas')
                    productos_a_mostrar = input('    Indique el número de productos que dese
a visualizar. \n a - Visualizar solo 10 productos sin ventas. \n b - Visualizar todos lo
s productos sin ventas\n')
                    if productos_a_mostrar == 'a' or productos_a_mostrar == 'b':
                        preguntar_consulta = 0 #AL poner esta variable a cero, se saldrá del c
iclo while
                        salir = 1
                    else:
                        print(' ¡ Opción no válida, vuelva a intentar !')
            else:
                print(' ! Selección no válida, intente nuevamente ¡')
            salir = 0

```



```

# -----Condición para filtrar por Categorías
-----
elif consulta_de_administrador == '2':
    salir = 0
    while salir == 0:
        filtro_por_categoria = input('Selecciónó filtrar por categorías, indique:\n
1 - Categorías con mayores ventas \n    2 - Categorías con menores ventas\n')
        if filtro_por_categoria == '1': #Bandera para mostrar las categorías más ven
didas
            print('Selecciónó filtrar por las categorías más vendidas\n')
            preguntar_consulta = 0 #Al poner esta variable a cero, se saldrá del cic
Lo while
            salir = 1
        elif filtro_por_categoria == '2': #Bandera para mostrar los categorías menos
vendas
            print('Selecciónó filtrar por las categorías menos vendidas\n')
            preguntar_consulta = 0 #Al poner esta variable a cero, se saldrá del cic
Lo while
            salir = 1
        else:
            print('    ¡ Selección no válida, intente nuevamente !')
            salir = 0

#----- Condición para filtrar por Búsquedas
-----
elif consulta_de_administrador == '3':
    print(' ¡ Aún no disponible, estamos trabajando en esta parte... !')
    preguntar_consulta = 0 #Al poner esta variable a cero, se saldrá del ciclo w
hile

    else:
        # Si no se seleccionó una opción descrita, ponemos esta variable a 1, para q
ue el ciclo while siga en curso
        # y sigamos preguntando hasta que se seleccione una opción válida

        print('    ¡ Opción para filtrar no válida, vuelva a intentar !')
        preguntar_consulta = 1 #Al poner esta variable a cero, se saldrá del ciclo w
hile

```

Cuando el usuario o administrado indique que quiere ver los productos más vendidos.

Lo que hacemos primero es crear una lista con los productos ordenados de mayor a menor

```

In [ ]: # Cuando 'consulta_de_administrador' sea igual a '1', significa que se van a mostrar Los
        productos más vendidos.

        if consulta_de_administrador == '1':
            #Obtenemos solo Los ID de cada producto vendido
            solo_ID=[] #Vector para guardar Los ID's de cada producto vendido
            for id in lifestore_sales:
                solo_ID.append(id[1]) #Vector que contiene Los ID's de Los todos Los productos v
            endidos
            #solo_ID

            #Contamos el numero de ventas para cada id, empezando por el id=1, id=2, id=3 ...
            contador = 0
            num_ventas=[]
            for comparador in range(1, max(solo_ID)+1):
                for id in solo_ID: # <-- aqui va La lista con solo ID's
                    if id == comparador:
                        contador = contador + 1
                #print(contador)
                num_ventas.append(contador)
                contador=0
            #num_ventas

            # Generamos un vector con ID's ascendentes, estos corresponden al numero de ID de Lo
            s productos vendidos.
            id_numero_ventas=[]
            for a in range(1,len(num_ventas)+1):
                id_numero_ventas.append(a)
            #id_numero_ventas #Esta lista llega hasta el 97, porque el número de ID_product más
            grande es el

            # Juntamos Las Listas de Los ID's de procutos vendidos con su correspondiente número
            de ventas
            for a in range(len(num_ventas)):
                id_numero_ventas.insert(a,[id_numero_ventas[a],num_ventas[a]])
                del id_numero_ventas[a+1]
            #print(id_numero_ventas)
            #print('\n Es decir; el ID_product=1 tuvo',id_numero_ventas[0][1],'ventas, el ID_pro
            duct= 2 tuvo',id_numero_ventas[1][1],'ventas, el ID_product=3 tuvo',id_numero_ventas[2]
            [1],'ventas ...')
            # Con esto lo que logramos es tener una lista de listas que contienen el ID y su cor
            respondiente número de ventas.

            #
            # AHORA ES MOMENTO DE ORDENAR LAS VENTAS DE MAYOR A MENOR.

            id_numero_ventas_para_ordenar = id_numero_ventas[:]
            a = [] #Variables que nos sirvan para ir vaciando en ellas datos en un determinado
            orden
            b = [] #Variables que nos sirvan para ir vaciando en ellas datos en un determinado
            orden

            for i in id_numero_ventas_para_ordenar: # Obtenemos el producto más vendido aún sin
            su ID
                a.append(i[1])
                veces_mas_vendido = max(a) # = 8

                i = 0

            while len(id_numero_ventas_para_ordenar) >= 0:
                #print('valor primero de i: ',i)

```

```

if id_numero_ventas_para_ordenar[i][1] == veces_mas_vendido:
    b.append(id_numero_ventas_para_ordenar[i])
    id_numero_ventas_para_ordenar.remove(id_numero_ventas_para_ordenar[i])
    if len(id_numero_ventas_para_ordenar) == 0:
        break
    #print('b: ',b)
    #print('ID numero ventas',id_numero_ventas)

    #Verificamos si hay otro id de producto que se haya vendido las mismas veces
    a = []
    for j in id_numero_ventas_para_ordenar: # Obtenemos el más vendido aún sin s
u ID
        a.append(j[1])
        veces_mas_vendido = max(a) # = 8

    #print('Veces más vendido segunda vuelta: ',veces_mas_vendido)
    i = -1
    #print('valor segundo de i: ',i)
    #print('-----')
    i = i + 1

id_product_y_num_ventas = b
# Esta es la lista final que contiene el ID de cada producto y su correspondiente n
úmero de ventas [ID, Veces vendido]
#

```

```

# Hasta aquí lo que tenemos es una lista de los productos más vendidos ordenados de
mayor número de
# ventas a menor número de ventas, la lista contiene el ID de producto y el número d
e ventas respectivamente.
# A los administradores les vamos a mostrar esta lista junto con el nombre del produ
cto y su categoría

id_prod_and_num_vent = id_product_y_num_ventas[:] #Lo copiamos en una nueva variable
para poder realizar consultas posteriores
id_prod_and_name = lifestore_products[:] #Lo copiamos en una nueva variable para pod
er realizar consultas posteriores

# AHORA VAMOS A REALIZAR LA UNION DE LAS LISTA [ID, VECES VENDIDO] CON [NOMBRE DEL P
RODUCTO]
# Unimos la lista que contiene el id del producto, el número de ventas lista con su
correspondiente
# de la lista que contiene los nombres

while len(id_prod_and_name) != (len(lifestore_products)-len(b)):
    match = 0

    # Estos contadores nos sirven para identificar en qué posición de la lista se en
cuentran los datos que nos interesan.

    contador1 = -1
    contador2 = -1
    for id_prod1 in id_prod_and_num_vent:
        contador1 = contador1 + 1
        contador2 = -1
        for id_prod2 in id_prod_and_name:
            contador2 = contador2 + 1
            #print('Id To merge',id_prod2[0])
            #print('Position2: ',contador2)
            if id_prod1[0] == id_prod2[0]:
                match = 1
                #print('Match y break')

```

```
                break
            if match == 1:
                id_prod_and_num_vent[contador1].append(id_prod_and_name[contador2][1])
                id_prod_and_name.remove(id_prod_and_name[contador2])
            break

    # 'id_prod_and_num_vent' <-- Esta lista contiene TODOS os [product_id, cantidad de v
    entas, nombre del producto]
    # Esa Lista no se muestra, ya que se le preguntará al usuario cuántos elementos des
    ea consultar.
```

Hasta aquí lo que tenemos es una lista que contiene TODOS los `id_productos` con su cantidad de ventas, y su nombre del producto. Esta información está en la variable **'id_prod_and_num_vent'**

Una vez que ya tenemos esta lista con **[id_product, cantidad de ventas, nombre del producto]**, lo que sigue es mostrar el número de elementos que el usuario o administrador nos indique. Pueden ser 5, 10, o todos.

La variable **'filtro_por_producto'** nos permite saber si se quieren mostrar los más vendidos, los menos vendidos o los que no han tenido ventas.

En esta parte del código también pedimos que el administrador indique si quiere ver solo los 5 más vendidos, o solo 10, o todos. La misma idea es para los productos menos vendidos y también para los productos sin ventas.

```

In [ ]: #_____ filtro_por_producto == '1' -> IMPRIMIR PRODUCTOS MÁS VENDIDOS
        if filtro_por_producto == '1':
            id_prod_and_num_vent_and_names=[] # Lista que modificaremos de tamaño dependiendo de la consulta del administrador
            if productos_a_mostrar == 'a': # La opción 'a' muestra 5 productos más vendidos
                print('\n Mostrando los 5 productos más vendidos: [ID del producto, Número de ventas, Nombre del producto] ...\n')
                id_prod_and_num_vent_and_names = id_prod_and_num_vent[:5] # Para mostrar 5 los productos
                for p in id_prod_and_num_vent_and_names:
                    print(p)
            elif productos_a_mostrar == 'b': # La opción 'b' muestra 10 productos más vendidos
                print('\n Mostrando los 10 productos más vendidos: [ID del producto, Número de ventas, Nombre del producto] ...\n')
                id_prod_and_num_vent_and_names = id_prod_and_num_vent[:10] #Para mostrar 10 productos
                for p in id_prod_and_num_vent_and_names:
                    print(p)
            elif productos_a_mostrar == 'c': # La opción 'c' muestra todos los productos
                print('\n Mostrando todos los productos: [ID del producto, Número de ventas, Nombre del producto] ...\n')
                id_prod_and_num_vent_and_names = id_prod_and_num_vent[:] #Para mostrar todos los productos
                for p in id_prod_and_num_vent_and_names:
                    print(p)
            else:
                print('Vamos a arreglar esta parte para que vuelva solicitar una entrada')
        #_____ filtro_por_producto == '2' -> IMPRIMIR PRODUCTOS MENOS VENDIDOS
        if filtro_por_producto == '2':
            id_prod_and_num_vent_and_names=[] # Lista que modificaremos de tamaño dependiendo de la consulta del administrador
            if productos_a_mostrar == 'a': # La opción 'a' muestra 5 productos menos vendidos
                print('\n Mostrando los 5 productos menos vendidos: [ID del producto, Número de ventas, Nombre del producto] ...\n')
                prod_sin_ventas = []
                for sublista in id_prod_and_num_vent: ## IMPORTANTE: Usaremos esta forma para imprimir los resultados en cada consulta.
                    if sublista[1] == 0:
                        prod_sin_ventas.append(sublista)
                prod_menos_vendidos = []
                prod_menos_vendidos = id_prod_and_num_vent[:len(id_prod_and_num_vent)-len(prod_sin_ventas)] #imprime hasta el elemento 42
                for menos_vendidos in prod_menos_vendidos[len(prod_menos_vendidos)-5:]: ## Aquí se especifica el limite de los productos a mostrar
                    print(menos_vendidos)
            elif productos_a_mostrar == 'b': # La opción 'b' muestra 10 productos menos vendidos
                print('\n Mostrando los 10 productos menos vendidos: [ID del producto, Número de ventas, Nombre del producto] ...\n')
                prod_sin_ventas = []
                for sublista in id_prod_and_num_vent: ## IMPORTANTE: Usaremos esta forma para imprimir los resultados en cada consulta.
                    if sublista[1] == 0:
                        prod_sin_ventas.append(sublista)
                prod_menos_vendidos = []
                prod_menos_vendidos = id_prod_and_num_vent[:len(id_prod_and_num_vent)-len(prod_sin_ventas)] #imprime hasta el elemento 42
                for menos_vendidos in prod_menos_vendidos[len(prod_menos_vendidos)-10:]: ## Aquí se especifica el limite de los productos a mostrar

```

```

        print(menos_vendidos)

    else:
        print('Vamos a arreglar esta parte para que vuelva solicitar una entrada')

#_____ filtro_por_producto == '3' -> IMPRIMIR PRODUCTOS SIN
VENTAS _____
    elif filtro_por_producto == '3':
        if productos_a_mostrar == 'a':
            print('\n Mostrando 10 productos sin ventas: [ID del producto, Número de ven
tas, Nombre del producto] ...\n')
            prod_sin_ventas = []
            for sublista in id_prod_and_num_vent:
                if sublista[1] == 0:
                    prod_sin_ventas.append(sublista)
            for para_mostrar in prod_sin_ventas[:10]:
                print(para_mostrar)
        elif productos_a_mostrar == 'b':
            print('\n Mostrando los productos sin ventas: [ID del producto, Número de ve
ntas, Nombre del producto] ...\n')
            prod_sin_ventas = []
            for sublista in id_prod_and_num_vent:  ## IMPORTANTE: Usaremos esta forma pa
ra imprimir los resultados en cada consulta.
                if sublista[1] == 0:
                    prod_sin_ventas.append(sublista)
            print('Hay ', len(prod_sin_ventas), 'productos que no han tenido ventas\n')
            for para_mostrar in prod_sin_ventas:
                print(para_mostrar)
    else:
        print('Vamos a arreglar esta parte par que vuleva a solitar una entrada')

```

La variable '**consulta_de_administrador**' nos indica que se necesita realizar una consulta por CATEGORIAS.

El sguiente código ya se realizó arriba, pero cuando se entra en la seleccion de realizar **consulta por CATEGORÍAS** la variable '**id_prod_and_num_vent**' es necesaria, por lo tanto la generamos.

```

In [ ]: if consulta_de_administrador == '2': #Se indico realizar consulta por CATEGORIAS

    #Obtenemos solo los ID de cada producto vendido
    solo_ID=[] #Vector para guardar los ID's de cada producto vendido
    for id in lifestore_sales:
        solo_ID.append(id[1]) #Vector que contiene los ID's de los todos los productos vendidos
    #solo_ID

    #Contamos el numero de ventas para cada id, empezando por el id=1, id=2, id=3 ...
    contador = 0
    num_ventas=[]
    for comparador in range(1, max(solo_ID)+1):
        for id in solo_ID: # <-- aqui va la lista con solo ID's
            if id == comparador:
                contador = contador + 1
            #print(contador)
            num_ventas.append(contador)
            contador=0
        #num_ventas

    # Generamos un vector con ID's ascendentes, estos corresponden al numero de ID de los productos vendidos.
    id_numero_ventas=[]
    for a in range(1,len(num_ventas)+1):
        id_numero_ventas.append(a)
    #id_numero_ventas #Esta lista llega hasta el 97, porque el número de ID_product más grande es el

    # Juntamos las listas de los ID's de productos vendidos con su correspondiente número de ventas
    for a in range(len(num_ventas)):
        id_numero_ventas.insert(a,[id_numero_ventas[a],num_ventas[a]])
        del id_numero_ventas[a+1]
    #print(id_numero_ventas)

    #print('\n Es decir; el ID_product=1 tuvo',id_numero_ventas[0][1],'ventas, el ID_product= 2 tuvo',id_numero_ventas[1][1],'ventas, el ID_product=3 tuvo',id_numero_ventas[2][1],'ventas ...')

    # Con esto lo que logramos es tener una lista de listas que contienen el ID y su correspondiente número de ventas.
    # Ahora es momento de ordenar las ventas de mayor a menor

    id_numero_ventas_para_ordenar = id_numero_ventas[:]
    a = []
    b = []
    for i in id_numero_ventas_para_ordenar: # Obtenemos el más vendido aún sin su ID
        a.append(i[1])
    veces_mas_vendido = max(a) # = 8
    #print('Veces más vendido: ',veces_mas_vendido)

    #for i in range(len(id_numero_ventas)):
    i = 0

    while len(id_numero_ventas_para_ordenar) >= 0:
        #print('valor primero de i: ',i)
        if id_numero_ventas_para_ordenar[i][1] == veces_mas_vendido:
            b.append(id_numero_ventas_para_ordenar[i])
            id_numero_ventas_para_ordenar.remove(id_numero_ventas_para_ordenar[i])
            if len(id_numero_ventas_para_ordenar) == 0:
                break
            #print('b: ',b)

```

```

# print('ID numero ventas', id_numero_ventas)

# Verificamos si hay otro id de producto que se haya vendido las mismas veces
a = []
for j in id_numero_ventas_para_ordenar: # Obtenemos el más vendido aún sin s
    a.append(j[1])
    veces_mas_vendido = max(a) # = 8

# print('Veces más vendido segunda vuelta: ', veces_mas_vendido)
i = -1
# print('valor segundo de i: ', i)
# print('-----')
i = i + 1
## Hasta aqui funciona perfecto para ordenar de mayor a menor [ID, Veces vendid
o]
id_product_y_num_ventas = b
# print('Product ID y número de veces vendido ordenados de mayor a menor: ', id_product_y_num_ventas) # Esta es la lista final que contiene
# el ID de cada producto y su correspondiente número de ventas

# Hasta aqui lo que tenemos es una lista de los productos más vendidos ordenados de mayor número de
# ventas a menor número de ventas, la lista contiene el ID de producto y el número de
# ventas respectivamente.
# A los administradores les vamos a mostrar esta lista junto con el nombre del producto y su categoría

id_prod_and_num_vent = id_product_y_num_ventas[:] # Lo copiamos en una nueva variable para poder realizar consultas posteriores
id_prod_and_name = lifestore_products[:] # Lo copiamos en una nueva variable para poder realizar consultas posteriores

# Ahora realizamos lo que correspondería a una función merge

# Merge lista y con su correspondiente de la lista que contiene los nombres
while len(id_prod_and_name) != (len(lifestore_products) - len(b)):
    match = 0
    contador1 = -1
    contador2 = -1
    for id_prod1 in id_prod_and_num_vent:
        contador1 = contador1 + 1
        # print('-----')
        # print('Id Original', id_prod1[0])
        # print('Position1: ', contador1)
        # print('-----')
        contador2 = -1
        for id_prod2 in id_prod_and_name:
            contador2 = contador2 + 1
            # print('Id To merge', id_prod2[0])
            # print('Position2: ', contador2)
            if id_prod1[0] == id_prod2[0]:
                match = 1
                # print('Match y break')
                break
        if match == 1:
            id_prod_and_num_vent[contador1].append(id_prod_and_name[contador2][1])
            id_prod_and_name.remove(id_prod_and_name[contador2])
            break
    # print(id_prod_and_num_vent) # <-- Esta lista contiene TODOS [product_id, cantidad de ventas, nombre del producto]
    # ESTO ES PORQUE PARA LAS CONSULTAS POR CATEGORIA, NECESITAMOS LA VARIABLE 'id_prod_and_num_vent'

```


El siguiente código nos va permitir obtener una lista que contenga, todas la diferentes categorías y su correspondiente número de ventas **[Categoría, Número de ventas]**, ésta información esta almacenada en la variable **categoria_ventas_final_ordenada**.

El proceso para obtener esta lista implica bastantes subprocesos, como identificar las distintas categorías, depues de tener las categorias con su número de ventas, tenemos que ordenarlas de mayor a menor para poder después solo imprimir las primeras que serían las que tienen mayor número de ventas, o las últimas que serían las que tiene el menor número de ventas

```

In [ ]: ##### PORBANDO TODO EL CODIGO COMPLETO para filtrar por categoria( Ya funciona) #####
#####
        lista_prod_ordenados = id_prod_and_num_vent[:] # /son 94 [ ID, ventas, nombre]
        lista_lifystore_prod = lifystore_products[:] # En cada match esta lista se irá reduc
iendo /son 96 [id, nombre, precio, categoria]
        lista_prod_cat = []
        contadori = -1
        contadorj = -1
        match = 0
        while len(lista_lifystore_prod) > 2:
            for i in lista_prod_ordenados:
                contadori = contadori + 1
                contadorj = -1
                for j in lista_lifystore_prod:
                    contadorj = contadorj + 1
                    #print(j)
                    if i[0] == j[0]:
                        match = 1
                        break
                if match == 1:
                    lista_prod_cat.append([lista_prod_ordenados[contadori][1],lista_lifestor
e_prod[contadorj][3]]) #Ventas / Categoria
                    lista_lifystore_prod.remove(lista_lifystore_prod[contadorj])
                    match = 0
                    contadori = -1
                    break

        #lista_prod_cat #<-- Esta variable tiene numero de ventas de un determinado product
o y su categoría.
        ## _____ HASTA AQUI TENEMOS EL NUMERO DE VENTAS DE UN DETERMINADO PRODUCTO Y SU
CATEGORIA_____

        # _____ Identificar las diferentes categorías _____
        cat_para_ir_borrando = lista_prod_cat[:] # Sobre esta vamos a iterar e ir borrando l
os repetidos para contar las categorías.
        lista_de_categorias = []
        cantidad_ventas_por_cat = []
        cat_para_ir_borrando[0][1]

        while len(cat_para_ir_borrando) > 0:
            # Contar el numero de veces que aparece 'discos duros' en la lista #aqui va el
WHILE
            categoria_actual = cat_para_ir_borrando[0][1] # discos duros
            lista_de_categorias.append(categoria_actual)
            #print('Lista de categorías: ',lista_de_categorias)

            contador_veces_que_aparece_en_lista = 0
            for i in cat_para_ir_borrando:
                if i[1] == lista_de_categorias[len(lista_de_categorias)-1]:
                    contador_veces_que_aparece_en_lista = contador_veces_que_aparece_en_list
a + 1
            #print('Contador veces que aparece en lista',contador_veces_que_aparece_en_list
a)

            suma = 0
            for s in cat_para_ir_borrando:
                if s[1] == lista_de_categorias[len(lista_de_categorias)-1]:
                    suma = suma + s[0]
            #print('Suma: ', suma)
            cantidad_ventas_por_cat.append(suma)
            #print('Cantidad de ventas por categoria: ',cantidad_ventas_por_cat)

```

```

    # Revisar si el segundo elemento en la lista cat_para_ir_borrando existe o no en
    la lista, en caso de que no exista, lo agregamos
    contador_2 = 0
    while contador_2 != contador_veces_que_aparece_en_lista:
        for elemento_a_chechar in cat_para_ir_borrando:
            if elemento_a_chechar[1] == lista_de_categorias[len(lista_de_categorias)-
1]:
                cat_para_ir_borrando.remove(elemento_a_chechar)
                contador_2 = contador_2 + 1
                #print(contador_2)
                break

    # Hasta que se esperaría tener una lista desordenada

#---Hasta aquí se reviso y va bien

# Uniendo las dos listas anteriores
categoria_ventas_final = []
for x in lista_de_categorias:
    categoria_ventas_final.append([x])
incremento = -1
for x in cantidad_ventas_por_cat:
    incremento = incremento +1
    categoria_ventas_final[incremento].append(x)
#for x in categoria_ventas_final: # <--- Esta es una variable que nos interesa
#    print(x)
#categoria_ventas_final | tiene categoria y total de ventas BUENO

var_prueba = categoria_ventas_final[:]
#valores_maximos_ventas_desordenados = []
categoria_ventas_final_ordenada = []

while len(var_prueba) > 0:
    valores_maximos_ventas_desordenados = []
    for z in var_prueba:
        valores_maximos_ventas_desordenados.append(z[1])
    valores_maximos_ventas_desordenados
    maximo = max(valores_maximos_ventas_desordenados)
    #print('max: ',maximo)

    for x in var_prueba:
        if x[1] == maximo:
            categoria_ventas_final_ordenada.append(x)
            var_prueba.remove(x)

```

Como la lista '**categoria_ventas_final_ordenada**' contiene [Categoria, Número de ventas], y es de tamaño 8, cuando se pida mostrar las categorías con mayores ventas, mostraremos las primeras 4, y cuando se pida mostrar las categorías con menores ventas, mostraremos las últimas 4, ya que están ordenadas de mayor a menos.

```
In [ ]: if filtro_por_categoria == '1':
        print('    Mostrando las categorías con más ventas [Categoría, Número de ventas]
        ...')
        for y in categoria_ventas_final_ordenada[:4]:
            print(y) ## <-- Esta variable es la de interés
    elif filtro_por_categoria == '2':
        print('    Mostrando las categorías con menores ventas [Categoría, Número de ven
        tas] ...')
        for y in categoria_ventas_final_ordenada[4:]:
            print(y) ## <-- Esta variable es la de interés
```

ASÍ ES COMO SE VE LA CONSOLA AL EJECUTAR EL CÓDIGO.

```
Bienvenido a Lifestore:
Seleccione: 1 si es Usuario, 2 si es Administrador 2

    Usted intenta acceder como Administrador
Ingrese su nombre de Administrador: Oscar
Ingrese su contraseña de Administrador: siete

    Acceso Permitido como Administrador
    Hola Oscar bienvenido.

Filtrar búsqueda por:
    1 - Productos
    2 - Categorías
    3 - Búsquedas
1
Seleccionó filtrar por productos, indique:
    1 - Productos más vendidos
    2 - Productos menos vendidos
    3 - Productos sin ventas
1
Seleccionó filtrar por los productos más vendidos
    Indique el número de productos que desea visualizar.
    a - Visualizar los 5 productos más vendidos.
    b - Visualizar los 10 productos más vendidos.
    c - Visualizar todos los productos
a

Mostrando los 5 productos más vendidos: [ID del producto, Número de ventas, Nombre del producto] ...

[54, 50, "SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm"]
[3, 42, 'Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth']
[5, 20, 'Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)']
[42, 18, 'Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD']
[57, 15, "SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm"]
❗
```

En este caso se indicó que un administrador accedió, filtro, y pidió visualizar los 5 productos más vendidos.

Como es Administrador, a éste le mostramos el **id_producto**, el **número de ventas** y el **nombre del producto**. En caso de que un Usuario hubiera pedido la misma consulta, **no** le mostraríamos el **id_producto**, ni el número de ventas. La misma idea aplica para las diferetes consultas, cuando sean filtradas por categorías o por búsquedas.

Sin embargo ya no me dio tiempo realizar todo eso, pero la idea para la elaboración del código sigue patrones muy similares de lo que hasta el momento se muestra en este reporte.