



Reporte 2020

Jessica Fernanda Rodríguez Mondragón

6 de Septiembre 2020

jesrmond@gmail.com

Git: <https://github.com/Jesrmond/Proyecto01EMTECH.git>

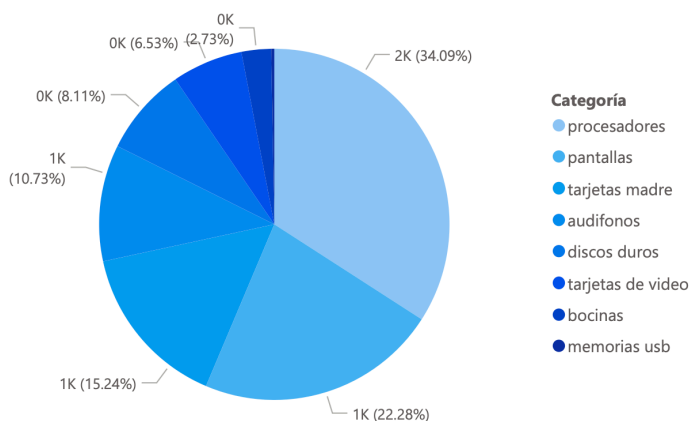
LifeStore es una tienda virtual que maneja una amplia gama de artículos, recientemente, la Gerencia de ventas, se percató que la empresa tiene una importante acumulación de inventario. Asimismo, se ha identificado una reducción en las búsquedas de un grupo importante de productos, lo que ha redundado en una disminución sustancial de sus ventas del último trimestre.

Productos

LifeStore cuenta con un total de 96 productos únicos, distribuidos en 8 categorías, en el siguiente cuadro se muestran los productos que se encuentran en inventario, teniendo un total de 5,341 productos.

Categoría	Stock
procesadores	1821
pantallas	1190
tarjetas madre	814
audifonos	573
discos duros	433
tarjetas de video	349
bocinas	146
memorias usb	15
Total	5341

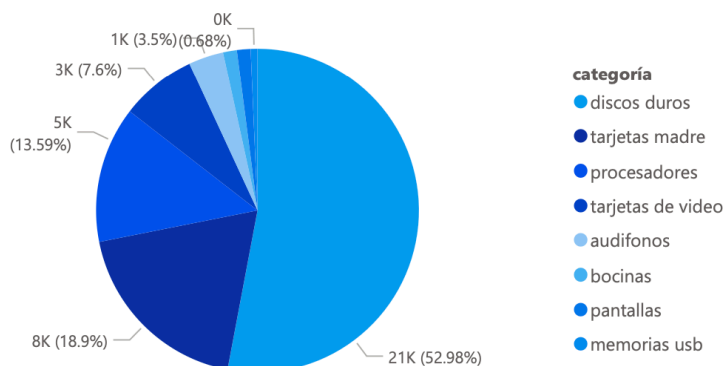
Productos en inventario por categoría



Como podemos notar, contamos con un mayor numero de procesadores en inventario, que abarcan el 34% de nuestros productos totales, a continuación, realizaremos un análisis sobre las ventas por categoría.

categoría	Ventas totales
discos duros	21291
tarjetas madre	7595
procesadores	5460
tarjetas de video	3055
audifonos	1405
bocinas	555
pantallas	551
memorias usb	274
Total	40186

Ventas totales por categoría



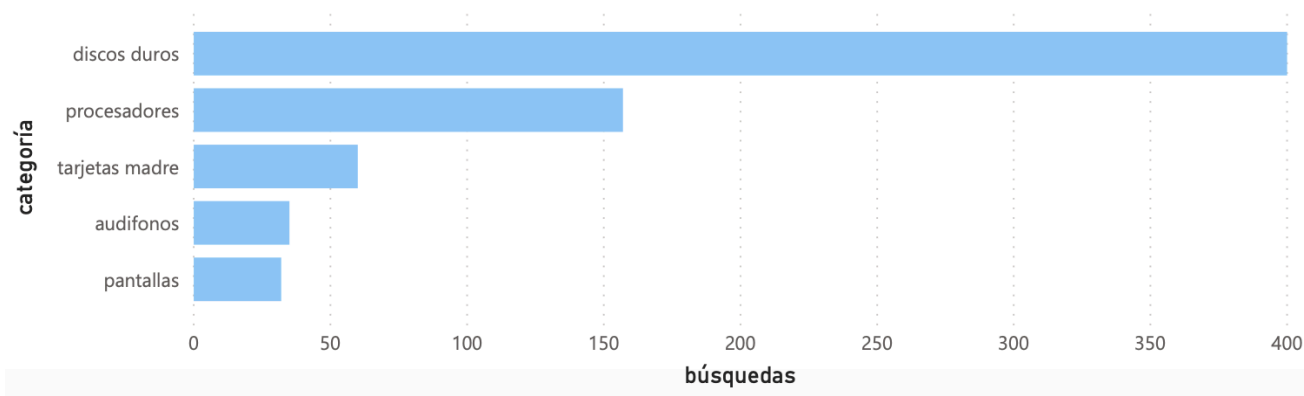
Al ver comportamiento de las ventas totales, nos percatamos que los productos más vendidos pertenecen a la categoría de discos duros con un total de 21, 291 ventas, es decir, el 52.98% del total de ventas, mientras que para los procesadores, que son los que más se tienen en inventario, las ventas totales pertenecen al 13.59%.

Por otra parte, a nivel producto, tenemos que el producto mas vendido, con un total de 50 ventas es el producto: SSD Kingston A400, 120GB, SATA III, 2.5", 7mm, que pertenece a la categoría de discos duros, el segundo mas vendido es el producto: Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth, con 42 ventas, que pertenece a la categoría de procesadores, mientras que el tercer mas vendido es el producto: Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake) que se vendió 20 veces.

A continuación se muestra el Top 10 de los productos más buscados por los clientes, y las categorías a las que pertenecen.

Producto	búsquedas
SSD Kingston A400, 120GB, SATA III, 2.5", 7mm	263
SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm	107
Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	60
Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth	55
Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire	41
Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul	35
TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro	32
Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación - Coffee Lake)	31
Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)	30
SSD XPG SX8200 Pro, 256GB, PCI Express, M.2	30

Top 10 de búsquedas por categoría



Por otro lado, se cuenta con el numero de calificación que los clientes asignan al producto, el cual puede estar entre el 0 para una calificación mala, hasta el 5, con una calificación buena, por lo que a continuación se muestran los 5 productos que obtuvieron las mejores calificaciones/reseñas del cliente.

El producto: Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache obtuvo una calificación de: 5

El producto: Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth obtuvo una calificación de: 5

El producto: Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire obtuvo una calificación de: 5

El producto: Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake) obtuvo una calificación de: 5

El producto: Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake) obtuvo una calificación de: 5

Y los productos que obtuvieron una calificación mas baja otorgada por los clientes:

El producto: Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0 obtuvo una calificación de: 1

El producto: Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD obtuvo una calificación de: 1

El producto: Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel obtuvo una calificación de: 1

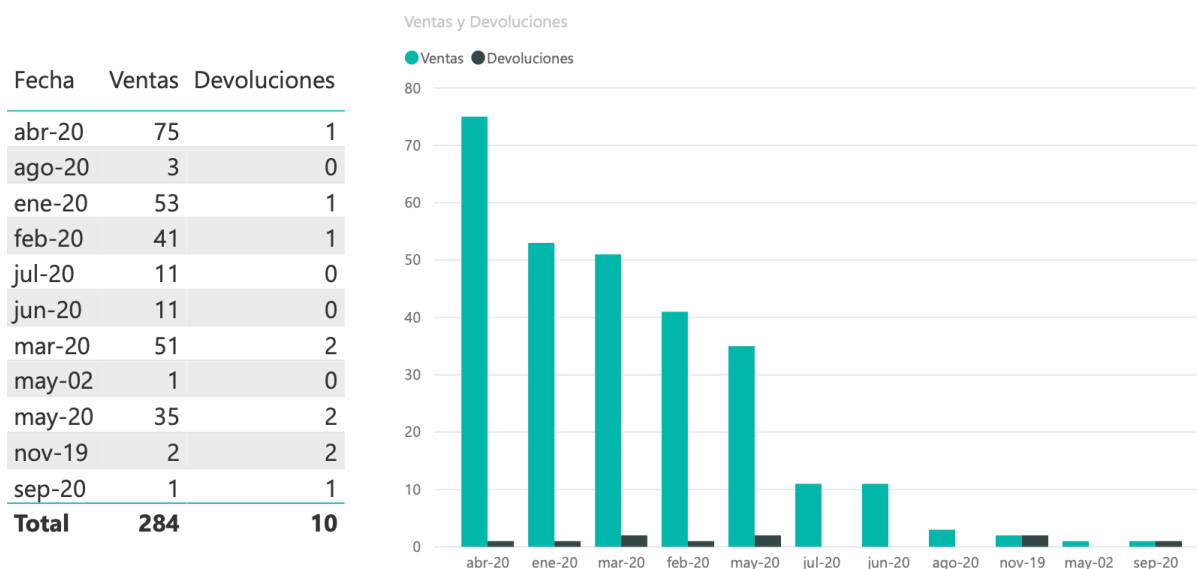
El producto: Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel obtuvo una calificación de: 2

El producto: Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth obtuvo una calificación de: 3

Ventas

LifeStore, tiene registradas ventas desde mayo del 2002, sin embargo, la única venta que se realizó ese año fue devuelta, para el año del 2019, se registró una venta y también fue devuelta, por lo que para el análisis solo consideraremos a partir del año 2020.

A continuación se muestra el histórico de las ventas mensuales, y las devoluciones.

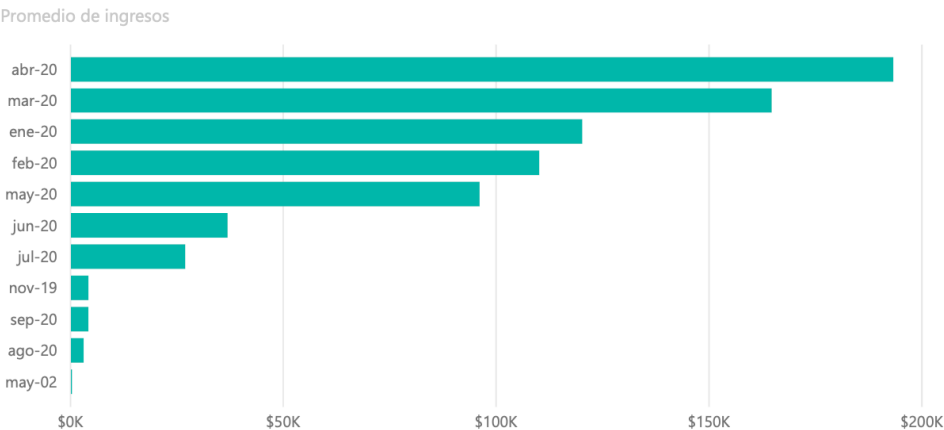


Por lo tanto para 2020 tenemos que la mayor cantidad de ventas se dieron en abril, con 74 ventas considerando devoluciones, además, el total de ventas registradas en LifeStore es de 274 ventas totales anuales correspondientes a 2020.

Ingresos

Los ingresos totales anuales para 2020 corresponden a \$755,709 MXN, siendo abril con el mayor numero de ingresos y agosto con el mes mas bajo.

Fecha	Ingreso
abr-20	\$193,295
mar-20	\$164,729
ene-20	\$120,237
feb-20	\$110,139
may-20	\$96,135
jun-20	\$36,949
jul-20	\$26,949
sep-20	\$4,199
ago-20	\$3,077



Código

Dentro de LifeStore, el personal de la tienda que tiene acceso a la información de inventario son los siguientes:

Posición:	Usuario :	Password:
CEO	CEO	ceolfs
Gerente de Operaciones	Operaciones	oplfs
Gerente de Ventas	ventas	venlfs
Gerente de Marketing	marketing	mktlfs
Gerente de Finanzas	finanzas	finlfs

Por lo que solo estos usuarios tienen permitido acceder a las bases de datos e información de LifeStore.

Se cuenta con información sobre los productos, ventas y búsquedas de la tienda LifeStore, distribuidos en las siguientes listas:

`lifestore-searches = [id_search, id product]`

`lifestore-sales = [id_sale, id_product, score (1 to 5), date, refund (1 for true or 0 for false)]`

`lifestore-products = [id_product, name, price, category, stock]`

Con la ayuda de python, se generó un análisis basado en la petición de la gerencia de ventas, obteniendo los resultados anteriores.

Consideraciones importantes:

Se debe iniciar sesión con los usuarios y contraseñas definidos en el punto anterior, en caso de no ser así, se marcará como inválido o incorrecto.

El código le da al usuario la oportunidad de elegir que es lo que quiere consultar mediante un menú de opciones, el cual es el siguiente:

```
Bienvenido a lifestore

¿Qué deseas consultar?

1. Productos con mayores ventas
2. Productos con mayores búsquedas
3. Productos con menores ventas por categoría
4. Productos con menores búsquedas por categoría
5. Productos con mejores reseñas
6. Productos con peores reseñas
7. Ventas totales mensuales
8. Ingresos totales mensuales
9. Meses con mayores ventas
10. Salir

Elige una opción: > 
```

Una vez que elija la opción el usuario, se mostrará la información solicitada, y en el punto 3 se mostrará un submenu a elegir por categoría.

Dentro del código, se definió una función para ordenar dependiendo de la posición de la lista nuestros datos, la cual es la siguiente:

```
def ordenar(valor):

    return valor[2]
```

La función anterior, aplicada a nuestra lista con la función `sorted(lista, key=ordenar)`, nos regresara la lista ordenada dependiendo el valor de la posición que indique la función `ordenar`.

Para poder realizar las visualizaciones, se exportaron los resultados arrojados por el código de python, aquí adjunto, como formato csv y las visualizaciones se crearon con la ayuda de Power BI, sin embargo el análisis anterior fue hecho con el código de python.

A continuación se encuentra el código trabajado para este proyecto, sin embargo también se anexa el código en formato .py para una mejor visualización y ejecución.

Anexo

```
from lifestore_file import lifestore_products
from lifestore_file import lifestore_sales
from lifestore_file import lifestore_searches

admin = [['CEO', 'ceolfs'], ['Operaciones', 'oplfs'], ['ventas', 'venlfs'],
          ['marketing', 'mktlfs'], ['finanzas', 'finlfs']]

usuario = input('Introduce tu usuario: ')
paswd = input('Ingresa tu contraseña: ')

adm = 0

for i in admin:
    if i[0] == usuario and i[1] == paswd:
        adm = 1

if adm == 1:
    print('\nBienvenido a lifestore\n')
    salir = False
    opcion = 0

    while not salir:
        print('\n¿Qué deseas consultar?\n')
```

```
print('1. Productos con mayores ventas')
print('2. Productos con mayores búsquedas')
print("3. Productos con menores ventas por categoría")
print("4. Productos con menores búsquedas por categoría")
print("5. Productos con mejores reseñas")
print('6. Productos con peores reseñas')
print('7. Ventas totales mensuales')
print('8. Ingresos totales mensuales')
print('9. Meses con mayores ventas')
print('10. Salir')

opcion = input('\nElige una opción: ')

if opcion == '1':
    print("Opcion 1: \nProductos con mayores ventas\n")
    print(
        'A continuación se encuentran los 50 productos con mayores ventas'
    )
    c = 0
    ventas = []
    for p in lifestore_products:
        for v in lifestore_sales:
            if p[0] == v[1]:
```

```
        c += 1

    form = [p[0], p[1], c]
    ventas.append(form)

    c = 0
```

```
def ordenar(valor):
    return valor[2]
```

```
ventas2 = sorted(ventas, key=ordenar, reverse=True)

for t in ventas2[:50]:
    print('El producto: ', t[1], 'se vendió: ', t[2], 'veces \n')
```

```
elif opcion == '2':
    print('\nLos 100 productos con mayores búsquedas son: \n')

    t = 0
    busqueda = []
    for p in lifestore_products:
        for s in lifestore_searches:
            if p[0] == s[1]:
                t += 1

        formato = [p[0], p[1], t]
        busqueda.append(formato)

    t = 0
```

```
def ordenar(valor):
```

```
    return valor[2]
```

```
for t in sorted(busqueda, key=ordenar, reverse=True):
```

```
    print('El producto: ', t[1], 'con ID: ', t[0], 'se buscó: ',  
          t[2], 'veces \n')
```

```
elif opcion == '3':
```

```
    exit = False
```

```
    suboption = 0
```

```
    while not exit:
```

```
        print('\n¿Qué categoría deseas consultar?\n')
```

```
        print('a. Procesadores')
```

```
        print('b. Tarjetas de video')
```

```
        print('c. Tarjetas madre')
```

```
        print('d. Discos duros')
```

```
        print('e. Memorias USB')
```

```
        print('f. Pantallas')
```

```
        print('g. Bocinas')
```

```
        print('h. Aufifonos')
```

```
        print('i. Todas las categorías')
```

```
        print('s. Regresar al menú principal')
```

```
    suboption = input('\nElige una opción: ')
```

```
if suboption == 'a':
    print('\nVentas de procesadores\n')
    cat = 0
    categoria = []

    for p in lifestore_products:
        for s in lifestore_sales:
            if p[0] == s[1]:
                cat += 1
            formato = [p[3], p[1], cat]
            categoria.append(formato)
            cat = 0

    a = [y for y in categoria if y[0] == 'procesadores']

    def ordenar(valor):
        return valor[2]

    for t in sorted(a, key=ordenar, reverse=False):
        print('CATEGORIA:', t[0], ';El producto: ', t[1],
              'se vendió: ', t[2], 'veces \n')
elif suboption == 'b':
    print('\n Ventas de Tarjetas de video \n')
```

```
cat = 0
```

```
categoria = []
```

```
for p in lifestore_products:
```

```
    for s in lifestore_sales:
```

```
        if p[0] == s[1]:
```

```
            cat += 1
```

```
    formato = [p[3], p[1], cat]
```

```
    categoria.append(formato)
```

```
    cat = 0
```

```
b = [y for y in categoria if y[0] == 'tarjetas de video']
```

```
def ordenar(valor):
```

```
    return valor[2]
```

```
for t in sorted(b, key=ordenar, reverse=False):
```

```
    print('CATEGORIA:', t[0], ';El producto: ', t[1],
```

```
        'se vendió: ', t[2], 'veces \n')
```

```
elif suboption == 'c':
```

```
    print('\n Ventas de Tarjetas madre \n')
```

```
    cat = 0
```

```
    categoria = []
```

```
for p in lifestore_products:
    for s in lifestore_sales:
        if p[0] == s[1]:
            cat += 1
        formato = [p[3], p[1], cat]
        categoria.append(formato)
        cat = 0
c = [y for y in categoria if y[0] == 'tarjetas madre']

def ordenar(valor):
    return valor[2]

for t in sorted(c, key=ordenar, reverse=False):
    print('CATEGORIA:', t[0], ';El producto: ', t[1],
          'se vendió: ', t[2], 'veces \n')

elif suboption == 'd':

    print('\n Ventas de Discos duros \n')
    cat = 0
    categoria = []

    for p in lifestore_products:
```

```
for s in lifestore_sales:
    if p[0] == s[1]:
        cat += 1
    formato = [p[3], p[1], cat]
    categoria.append(formato)
    cat = 0
d = [y for y in categoria if y[0] == 'discos duros']
```

```
def ordenar(valor):
    return valor[2]
```

```
for t in sorted(d, key=ordenar, reverse=False):
    print('CATEGORIA:', t[0], ';El producto: ', t[1],
          'se vendió: ', t[2], 'veces \n')
```

```
elif suboption == 'e':
    print('\n Ventas de Memorias USB \n')
    cat = 0
    categoria = []
```

```
for p in lifestore_products:
    for s in lifestore_sales:
        if p[0] == s[1]:
            cat += 1
```

```
formato = [p[3], p[1], cat]
categoria.append(formato)
cat = 0
e = [y for y in categoria if y[0] == 'memorias usb']
```

```
def ordenar(valor):
    return valor[2]

for t in sorted(e, key=ordenar, reverse=False):
    print('CATEGORIA:', t[0], ';El producto: ', t[1],
          'se vendió: ', t[2], 'veces \n')
```

```
elif suboption == 'f':
    print('\n Ventas de Pantallas \n')
    cat = 0
    categoria = []
```

```
for p in lifestore_products:
    for s in lifestore_sales:
        if p[0] == s[1]:
            cat += 1
        formato = [p[3], p[1], cat]
        categoria.append(formato)
    cat = 0
```

```
f = [y for y in categoria if y[0] == 'pantallas']
```

```
def ordenar(valor):
```

```
    return valor[2]
```

```
for t in sorted(f, key=ordenar, reverse=False):
```

```
    print('CATEGORIA:', t[0], ';El producto: ', t[1],
```

```
        'se vendió: ', t[2], 'veces \n')
```

```
elif suboption == 'g':
```

```
    print('\n Ventas de Bocinas \n')
```

```
    cat = 0
```

```
    categoria = []
```

```
for p in lifestore_products:
```

```
    for s in lifestore_sales:
```

```
        if p[0] == s[1]:
```

```
            cat += 1
```

```
    formato = [p[3], p[1], cat]
```

```
    categoria.append(formato)
```

```
    cat = 0
```

```
g = [y for y in categoria if y[0] == 'bocinas']
```

```
def ordenar(valor):
```

```
return valor[2]
```

```
for t in sorted(g, key=ordenar, reverse=False):  
    print('CATEGORIA:', t[0], ';El producto: ', t[1],  
          'se vendió: ', t[2], 'veces \n')
```

```
elif suboption == 'h':  
    print('\n Ventas de Audífonos \n')  
    cat = 0  
    categoria = []
```

```
for p in lifestore_products:  
    for s in lifestore_sales:  
        if p[0] == s[1]:  
            cat += 1  
        formato = [p[3], p[1], cat]  
        categoria.append(formato)  
    cat = 0  
h = [y for y in categoria if y[0] == 'audifonos']  
for t in sorted(h, key=ordenar, reverse=False):  
    print('CATEGORIA:', t[0], ';El producto: ', t[1],  
          'se vendió: ', t[2], 'veces \n')
```

```
elif suboption == 'i':
```

```
print(
    '\n Listado de los 50 productos con menores ventas por categoría: \n'
)
cat = 0
categoria = []

for p in lifestore_products:
    for s in lifestore_sales:
        if p[0] == s[1]:
            cat += 1
        formato = [p[3], p[1], cat]
        categoria.append(formato)
        cat = 0

def ordenar2(valor):
    return valor[2]

categoria2 = sorted(categoria, key=ordenar2, reverse=False)
for t in categoria2[:50]:
    print('CATEGORIA:', t[0], ';El producto: ', t[1],
        'se vendió: ', t[2], 'veces \n')

elif suboption == 's':
```

```
        break

    else:

        print('Opción incorrecta')

elif opcion == '4':

    print(

        '\n Los 100 productos con menores búsquedas por categoría son los siguientes:

\n'

    )

    bus = 0

    categoria_busqueda = []

    for p in lifestore_products:

        for s in lifestore_searches:

            if p[0] == s[1]:

                bus += 1

            formato = [p[3], p[1], bus]

            categoria_busqueda.append(formato)

            bus = 0

def ordenar(valor):

    return valor[2]

for t in sorted(categoria_busqueda, key=ordenar, reverse=False):
```

```
print('CATEGORIA:', t[0], ';El producto: ', t[1], 'tuvo: ',  
      t[2], 'búsquedas \n')
```

```
elif opcion == '5':
```

```
    print(  
        '\n Los 20 productos con mejores reseñas, son los siguientes: \n'  
    )
```

```
    reseña = []
```

```
    for p in lifestore_products:
```

```
        for s in lifestore_sales:
```

```
            if p[0] == s[1]:
```

```
                formato = [p[1], s[2]]
```

```
            reseña.append(formato)
```

```
def res(valor):
```

```
    return valor[1]
```

```
mejores_reseñas = sorted(reseña, key=res, reverse=True)
```

```
for t in mejores_reseñas[:20]:
```

```
    print('El producto: ', t[0], 'obtuvo una calificación de: ',  
          t[1])
```

```
elif opcion == '6':
```

```
print(
    '\n Los 20 productos con las reseñas mas bajas, son los siguientes: \n'
)

reseña_mala = []
for p in lifestore_products:
    for s in lifestore_sales:
        if p[0] == s[1]:
            formato = [p[1], s[2]]
            reseña_mala.append(formato)

def res(valor):
    return valor[1]

malas_reseñas = sorted(reseña_mala, key=res, reverse=False)
for t in malas_reseñas[:20]:
    print('El producto: ', t[0], 'obtuvo una calificación de: ',
          t[1])
elif opcion == '7':
    print('\n Ventas totales mensuales:\n')
    dev = [(x[3][3:], x[4]) for x in lifestore_sales if x[4] == 1]
    dates = [x[3] for x in lifestore_sales]
    months = [x[3:] for x in dates]
    ventas_por_mes = list(set(months))
```

```
aux = []

for a in ventas_por_mes:
    aux.append((a, months.count(a), [x[0] for x in dev].count(a)))

def mon(valor):
    return valor[0]

for t in sorted(aux, key=mon):
    print('Fecha: ', t[0], 'ventas: ', t[1], 'devoluciones:', t[2])

elif opcion == '8':
    print('\nIngresos totales mensuales: \n')
    dev = [(x[3][3:], x[4]) for x in lifestore_sales
            if x[4] == 1] #devolución

    dates = [x[3] for x in lifestore_sales]
    months = [x[3:] for x in dates]
    months_ = list(set(months))
    aux = []
    for a in months_:
        aux.append((a, months.count(a), [x[0] for x in dev].count(a)))
    aux2 = [[x[3][3:], x[1], x[4]] for x in lifestore_sales]
    id_price = dict(
        zip([x[0] for x in lifestore_products],
```

```
        [x[2] for x in lifestore_products]))

id_name = dict(
    zip([x[0] for x in lifestore_products],
        [x[1] for x in lifestore_products]))

prices = [[x[0], x[1], x[2], id_price[x[1]]] for x in aux2]

aux3 = []
aux4 = []

for m in months_:
    sells = 0
    for p in prices:
        if p[0] == m:
            sells += p[3]
    aux4.append([m, sells])

def ingresos(valor):
    return valor[1]

for t in sorted(aux4, key=ingresos, reverse=True):
    print('\n Fecha:', t[0], 'ingresos totales: ', t[1])

elif opcion == '9':
```

```
print('\n Meses con mayores ventas: \n ')

dev = [(x[3][3:], x[4]) for x in lifestore_sales if x[4] == 1]

dates = [x[3] for x in lifestore_sales]

months = [x[3:] for x in dates]

ventas_por_mes = list(set(months))

aux = []

for a in ventas_por_mes:

    aux.append((a, months.count(a), [x[0] for x in dev].count(a)))

def mon2(valor):

    return valor[1]

aux2 = sorted(aux, key=mon2, reverse=True)

mayores_ventas = [x for x in aux2 if x[1] >= 40]

for t in mayores_ventas:

    print('Mes: ', t[0], 'ventas', t[1])

elif opcion == '10':

    break

else:

    print(

        'Opción incorrecta...\ningresa una nueva opción para continuar'

    )

print("Sesión cerrada")
```

```
else:
```

```
    print('Usuario o contraseña invalidos, por favor intente de nuevo')
```