



PROYECTO 1.

INTRODUCCIÓN A PYTHON

09.14.2021

Miguel Angel Santamaria Vilchis
Grupo 1

1. Índice

1. Índice	1
2. Introducción	3
3. Definición del código	3
3.1 Validaciones	3
Fig 3.1.0 Validaciones	4
3.2 Login	4
Fig 3.2.0 Login	4
Fig 3.2.1 Login en ejecución	4
3.3 Menu	4
Fig 3.3.0 Menu	5
Fig 3.3.1 Menu y validación de entero en ejecución	5
3.4 Tabla Maestra	6
Fig 3.4.0 Tabla Maestra	7
3.5 Reportes de productos	7
Fig 3.5.0 Reportes Generales	9
Fig 3.5.1 Ejemplo de método de reporte individual	9
3.6 Reporte de ventas	9
Fig 3.6.0 Reporte de ventas	11
4. Solución al problema	12
4.1 Productos más vendidos y productos rezagados	12
Fig 4.1.0 Top 15 productos más vendidos	12
Fig 4.1.1 Top 5 productos menos vendidos	12
Fig 4.1.2 Top 20 productos más buscados	13
Fig 4.1.3 Top 20 productos menos buscados	13
4.2 Productos por reseña en el servicio	14
Fig 4.2.0 Top 10 productos con mejor puntuación	14
Fig 4.2.1 Top 10 productos con peor puntuación	14
Fig 4.2.2 Top 10 productos con más devoluciones y peor puntuación	14
4.3 Reporte de ventas	15
Fig 4.3.0 Ingresos anuales, promedio mensual y ventas por mes ordenadas de mayor a menor	15
5. Conclusión	16



2. Introducción

La programación en Python es una herramienta tecnológica que facilita el análisis, manejo y clasificación de datos, y que es amigable con el usuario por ser un lenguaje de alto nivel. Por este motivo es de relevancia para la Ciencia de Datos que las comunidades aprendan a usarlo, ya que su facilidad de uso y sintaxis simple facilitan la adaptación para personas que no tienen experiencia en ingeniería.

Para este proyecto se planteó el escenario de la empresa LifeStore, quienes requieren de un sistema de para lidiar con su acumulación de inventario. La tienda ha identificado patrones en las búsquedas de productos y rotación de productos que pueden ser útiles para determinar que productos no conviene tener en el inventario.

3. Definición del código

El código se compone de las siguientes secciones:

3.1 Validaciones

Son métodos auxiliares que permiten asegurar de que la entrada puede convertirse en el tipo de variable que es necesitada en alguna parte del código. Reciben como argumento el mensaje con el que se va a solicitar el dato de entrada y regresan el dato en el tipo de variable solicitado. Inician un ciclo infinito que sólo se puede terminar hasta que se ingresa una entrada válida para la conversión.

```
def validarString(txt):  
    return input(txt)  
  
def validarInt(txt):  
    while True:  
        n = input(txt)  
        if n.isnumeric():  
            return int(n)  
        else:  
            print("No es entero. Intenta de nuevo")  
  
def validarFloat(txt):  
    while True:  
        n = input(txt)  
        if n.isdecimal():
```

```

        return float(n)
    else:
        print("No es flotante. Intenta de nuevo")

```

Fig 3.1.0 Validaciones

3.2 Login

Método de login que no permite acceder al Menu hasta que se ingresen las credenciales correctas. Entra en un ciclo infinito que sólo se puede terminar cuando se ingresan el usuario y la contraseña correctas al mismo tiempo. Si al menos uno de los dos datos es incorrecto, despliega un mensaje de que las credenciales son inválidas. En el caso contrario, manda llamar al método Menu.

```

def login():
    print("Bienvenid@ al Sistema")
    while True:
        user=validarString("Ingresa el usuario: ")
        password=validarString("Ingresa la contraseña: ")

        if user!="usuario-administrador" or password!="Emtech2021#": #
            print("Usuario o contraseña invalidos")
            print()
        else:
            menu()
            break

```

Fig 3.2.0 Login

```

Bienvenid@ al Sistema
Ingresa el usuario: user
Ingresa la contraseña: pass
Usuario o contraseña invalidos

Ingresa el usuario: usuario-administrador
Ingresa la contraseña: Emtech2021#

Bienvenid@ al menú

```

Fig 3.2.1 Login en ejecución

3.3 Menu

El método menu imprime mensajes con las acciones que se pueden realizar en el programa que son 1) Productos más vendidos y productos rezagados, 2) Productos por

reseña en el servicio, y 3) Reporte de ventas; además de la opción 0) Salir del sistema, que permite terminar con la ejecución del código.

Posteriormente, inicia un ciclo infinito que pedirá que se ingrese un entero de la acción que se quiera realizar y del que solo se puede salir al ingresar la opción 0. Si se ingresa un entero que no esté dentro de las opciones desplegará el mensaje de opción no válida. Cada opción válida llama a distintos métodos que despliegan la información solicitada.

```
def menu():
    print()
    print("Bienvenid@ al menú")
    print("1) Productos más vendidos y productos rezagados")
    print("2) Productos por reseña en el servicio")
    print("3) Reporte de ventas")
    print("0) Salir del sistema")

    while True:
        op=validarInt("Introduzca una opcion: ")
        if op == 0:
            print("Logout")
            break
        if op == 1:
            data=crearTablaMaestra()
            productosPorVentas(data)
        if op == 2:
            data=crearTablaMaestra()
            productosPorReseña(data)
        if op == 3:
            crearBalance()
        else:
            print("Opción no válida")
    crearTablaMaestra()
```

Fig 3.3.0 Menu

```
Bienvenid@ al menú
1) Productos más vendidos y productos rezagados
2) Productos por reseña en el servicio
3) Reporte de ventas
0) Salir del sistema
Introduzca una opcion: a
No es entero. Intenta de nuevo
Introduzca una opcion: 5
Opción no válida
Introduzca una opcion: 0
Logout
```

Fig 3.3.1 Menu y validación de entero en ejecución

3.4 Tabla Maestra

El método crearTablaMaestra() crea una lista de listas que contiene información ordenada de las listas lifestore_searches, lifestore_sales y lifestore_products. Esto nos permite usar la función sort que ya viene implementada en Python y filtrar la información tomando como criterio diferentes columnas.

Para la creación de la Tabla Maestra, primero se itera por la lista lifestore_products para asegurar que hay un registro de cada producto y se obtiene los datos de id, precio y nombre; después se itera la lista lifestore_sales para obtener los datos de número de ventas, puntuación y devoluciones y los guarda de manera acumulada; posteriormente se itera la tabla lifestore_searches para obtener los datos de búsquedas, que también se guardan de manera acumulada; finalmente se itera sobre la Tabla Maestra para realizar el cálculo de ingresos totales y de la puntuación promedio por productos.

El método regresa la lista con los datos de la Tabla Maestra.

```
def crearTablaMaestra():
    print("Reporte de Ventas")
    """ ESTRUCTURA DE DATA
    [0]=id producto
    [1]=precios -> cambia a ingresos por producto
    [2]=ventas
    [3]=puntuacion acumulada -> cambia a puntuacion promedio
    [4]=devoluciones
    [5]=busquedas
    [6]=nombre

    """
    # Llena data[] al iterar las tablas
    data=[]

    for row in lifestore_products:
        # Inicializa la tabla y obtiene id, precio y nombre
        data.append([row[0],row[2],0,0,0,0,row[1]]) # Ya estaban
ordenados los id
```

```

    for row in lifestore_sales:
        # Obtiene ventas, puntuacion y devoluciones
        i=row[1]-1                # Hace el ajuste del indice pgra
que inicie en 0
        data[i][2]+=1            # Acumulador por cada repeticion
del id
        data[i][3]+=row[2]        # Suma la puntuacion cada que
aparece el producto
        data[i][4]-=row[4]        # Guardadas en negativo para
facilitar el sort

    for row in lifestore_searches:
        i=row[1]-1                # Hace el ajuste del indice para
que inicie en 0
        data[i][5]+=1            # Acumulador por cada repeticion
del id

    for row in data:
        # Obtener puntuación promedio
        if row[2]!=0:
            row[3]=row[3]/row[2]    # puntuacion acumulada/ventas
        else:
            row[3]=0                # En caso de division con 0

        # Ingresos por producto

        row[1]=(row[2]+row[4])*row[1] # [ventas+(-devoluciones)]*precio

    return data

```

Fig 3.4.0 Tabla Maestra

3.5 Reportes de productos

En la sección de reportes existen dos grupos de funciones: a) las que organizan los productos por número de ventas y búsquedas, y b) las que organizan los productos por puntuación promedio y devoluciones. Ambos métodos reciben como parámetro la Tabla Maestra.

El método `productosPorVentas(data)` muestra los reportes de los productos:

- por número de ventas descendente,
- por número de ventas ascendente,
- por número de búsquedas ascendente,
- por número de búsquedas descendente.

Y el método `productosPorReseña(data)` muestra los reportes de los productos:

- por puntuación promedio y devoluciones ascendentes,
- por puntuación promedio y devoluciones descendentes,
- por devoluciones y puntuación promedio descendentes (cambia porque se usa las devoluciones como criterio principal).

Para cada reporte se creó una función única que recibe como parámetro la Tabla Maestra y realiza las siguientes acciones:

1. Usa la función `sort()` para ordenar los datos (ya sea de manera ascendente o descendente) usando como criterio alguna columna de la Tabla Maestra
2. Despliega en la consola los datos hasta el límite especificado por el problema y en un formato para que sea comprendido por el usuario.

```
def productosPorVentas(data):

    print("-----")
    print("-----PRODUCTOS POR VENTAS-----")
    print("-----")

    # Ordenar por ventas descendente
    ventasAsc(data)
    print()

    # Ordenar por ventas ascendente
    ventasDes(data)
    print()

    # Ordenar por búsquedas ascendente
    busquedasAsc(data)
    print()

    # Ordenar por búsquedas descendente
    busquedasDes(data)
    print()

def productosPorReseña(data):

    print("-----")
    print("-----PRODUCTOS POR RESEÑA-----")
```

```

print("-----")
# Ordenar por puntuacion promedio y devoluciones ascendente
puntuacionAsc(data)
print()

# Ordenar por puntuacion promedio y devoluciones descendente
puntuacionDes(data)
print()

# Ordenar por devoluciones y puntuacion promedio descendente
devolucionesDes(data)
print()

```

Fig 3.5.0 Reportes Generales

```

def devolucionesDes(data):
    data.sort(key=lambda row: (row[4],row[3]),reverse=False)

    print("Productos con más devoluciones y peor puntuacion")
    print("Rank  Puntuacion  Devoluciones  ID  Nombre")
    # Hasta 10 productos
    for i in range(0,10):
        pun=str(data[i][3])[:5]
        dev=str(data[i][4] *-1)
        id=str(data[i][0])
        nom=data[i][6]
        i=str(i+1)
        txt=i+(6-len(i))*" "+pun+(12-len(pun))*" "+dev+(14-len(dev))*"
"+id+(4-len(id))*" "+nom
        print(txt)

```

Fig 3.5.1 Ejemplo de método de reporte individual

3.6 Reporte de ventas

El método crearBalance() crea una lista de ingresos que contiene el id del producto, la fecha compra y el precio por producto. Para crear dicha lista se itera por la lista de lifestore_sales para obtener los datos de id del producto y fecha de compra, y usa el resultado del id para buscar su precio en la tabla de lifestore_sales.

Similar al caso de la Tabla Maestra, la lista de ingresos sirve para tener los datos de manera ordenada y se pueda usar la función sort para obtener las ventas por mes. Se mandan llamar dos funciones: totalAnual() y totalMes().

La función totalAnual() itera sobre la lista de ingresos y en una variable acumuladora suma la columna con los precios de los productos para obtener el total de ingresos anual y el promedio mensual.

La función `totalMes()` primero usa la función `sort()` para ordenar las ventas por mes de menor a mayor. Posteriormente se crea un arreglo con los nombres de los meses y después se itera sobre la lista de ingresos, donde se usa una variable que acumula la suma del precio del producto siempre y cuando el mes sea el mismo del anterior, de lo contrario se guarda el acumulado en el arreglo de meses y se reinicia la variable con la suma. Cuando termina el ciclo, se usa la función `sort()` en el arreglo con los meses para obtener los meses con más ventas en el año.

```
def crearBalance():
    print("-----")
    print("-----REPORTE DE VENTAS-----")
    print("-----")

    """ ESTRUCTURA DE DATA
    [0]=id producto
    [1]=fecha
    [2]=precio

    """

    ingresos=[]
    for row in lifestore_sales:
        if row[4]==0:          # Considera que en las devoluciones no hay
                                ingresos
            id=row[1]
            price=lifestore_products[id-1][2]
            ingresos.append([id, row[3], price])

    """Reportes"""

    # Obtiene los ingresos totales y obtiene el promedio mensual
    totalAnual(ingresos)

    # Obtiene los ingresos por mes y los ordena de mayor a menor
    totalMes(ingresos)

def totalAnual(ingresos):
    sum=0
    for row in ingresos:
        sum+=row[2]

    txt="Los ingresos anuales totales son: $" +str(sum)
    print(txt)
    print()
    txt="Las ventas promedio por mes son: $" +str(int(sum/12))
    print(txt)
    print()

def totalMes(ingresos):
```

```

    ingresos.sort(key=lambda row: row[1][3:5])    # Los organiza por mes

    """ ESTRUCTURA DE DATA
    [0]=mes
    [1]=ingresos

    """

meses=[["ENE",0],["FEB",0],["MAR",0],["ABR",0],["MAY",0],["JUN",0],["JUL",
0],["AGO",0],["SEP",0],["OCT",0],["NOV",0],["DIC",0]]

    actual=1
    for row in ingresos:
        if int(row[1][3:5])==actual:
            meses[actual-1][1]+=row[2]
        else:
            actual+=1
            meses[actual-1][1]+=row[2]

    print("Meses con mas ventas")
    print("Rank Mes Ventas")

    meses.sort(key=lambda row: row[1], reverse=True)
    i=1
    for row in meses:
        rank=str(i)
        mes=row[0]
        ventas=str(row[1])
        txt=(2-len(rank))*"0"+rank+"      "+mes+"      "+ "$"+ventas

        print(txt)
        i+=1
    print()

```

Fig 3.6.0 Reporte de ventas

4. Solución al problema

De acuerdo con las solicitudes de la Gerencia de Ventas, el programa regresa los siguientes reportes.

Para acceder al repositorio código está en el siguiente link:

<https://github.com/ElMike1712/ST21-GP1-Parte-1-Fundamentos-de-Programaci-n-con-Python.git>

4.1 Productos más vendidos y productos rezagados

Productos mas vendidos			
Rank	Ventas	ID	Nombre
1	50	54	SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm
2	42	3	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth
3	20	5	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)
4	18	42	Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
5	15	57	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm
6	14	29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
7	13	2	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth
8	13	4	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador
9	11	47	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2
10	9	12	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0
11	9	48	SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2
12	7	7	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)
13	6	31	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
14	6	44	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AMD B450, 64GB DDR4 para AMD
15	5	18	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030, 2GB 64-bit GDDR5, PCI Express x16 3.0

Fig 4.1.0 Top 15 productos más vendidos

Productos menos vendidos			
Rank	Ventas	ID	Nombre
1	0	9	Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)
2	0	14	Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB 64-bit GDDR3, PCI Express 2.0
3	0	15	Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0
4	0	16	Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming, 6GB 192-bit GDDR6, PCI Express 3.0
5	0	19	Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile, 4GB 128-bit GDDR5, PCI Express 3.0 x16

Fig 4.1.1 Top 5 productos menos vendidos

Productos mas buscados			
Rank	Busquedas	ID	Nombre
1	263	54	SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm
2	107	57	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm
3	60	29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
4	55	3	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth
5	41	4	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire
6	35	85	Logitech Audifonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul
7	32	67	TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro
8	31	7	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)
9	30	47	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2
10	30	5	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)
11	27	48	SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2
12	25	44	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AMD B450, 64GB DDR4 para AMD
13	24	2	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth
14	23	42	Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
15	20	8	Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake)
16	15	66	TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Widescreen, Negro
17	15	21	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC, 8GB 128-bit GDDR6, PCI Express 4.0
18	15	12	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0
19	11	51	SSD Kingston UV500, 480GB, SATA III, mSATA
20	11	18	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030, 2GB 64-bit GDDR5, PCI Express x16 3.0

Fig 4.1.2 Top 20 productos más buscados

Productos menos buscados			
Rank	Busquedas	ID	Nombre
1	0	14	Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB 64-bit GDDR3, PCI Express 2.0
2	0	16	Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming, 6GB 192-bit GDDR6, PCI Express 3.0
3	0	19	Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile, 4GB 128-bit GDDR5, PCI Express 3.0 x16
4	0	20	Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC, 8 GB 256 bit GDDR6, PCI Express x16 3.0
5	0	23	Tarjeta de Video MSI Radeon X1550, 128MB 64 bit GDDR2, PCI Express x16
6	0	24	Tarjeta de Video PNY NVIDIA GeForce RTX 2080, 8GB 256-bit GDDR6, PCI Express 3.0
7	0	30	Tarjeta Madre AORUS ATX Z390 ELITE, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel
8	0	32	Tarjeta Madre ASRock Z390 Phantom Gaming 4, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel
9	0	34	Tarjeta Madre ASUS ATX ROG STRIX B550-F GAMING WI-FI, S-AM4, AMD B550, HDMI, max. 128GB DDR4 para AMD
10	0	36	Tarjeta Madre Gigabyte micro ATX Z490M GAMING X (rev. 1.0), Intel Z490, HDMI, 128GB DDR4 para Intel
11	0	37	Tarjeta Madre ASRock ATX Z490 STEEL LEGEND, S-1200, Intel Z490, HDMI, 128GB DDR4 para Intel
12	0	38	Tarjeta Madre Gigabyte Micro ATX H310M DS2 2.0, S-1151, Intel H310, 32GB DDR4 para Intel
13	0	41	Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM, S-1151, Intel H370, HDMI, 64GB DDR4 para Intel
14	0	43	Tarjeta Madre ASUS ATX ROG STRIX Z390-E GAMING, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel
15	0	53	SSD Addlink Technology S70, 512GB, PCI Express 3.0, M.2
16	0	55	SSD para Servidor Supermicro SSD-DM128-SMCMVN1, 128GB, SATA III, mSATA, 6Gbit/s
17	0	58	SSD para Servidor Lenovo Thinksystem S4510, 480GB, SATA III, 2.5'', 7mm
18	0	61	Kit Memoria RAM Corsair Vengeance LPX DDR4, 2400MHz, 32GB, Non-ECC, CL16
19	0	62	Makena Smart TV LED 32S2 32'', HD, Widescreen, Gris
20	0	64	Samsung TV LED LH43QMREBGCXGO 43, 4K Ultra HD, Widescreen, Negro

Fig 4.1.3 Top 20 productos menos buscados

4.2 Productos por reseña en el servicio

Productos con mejor puntuación				
Rank	Puntuación	Devoluciones	ID	Nombre
1	5.0	0	1	Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache
2	5.0	0	6	Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake)
3	5.0	0	7	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)
4	5.0	0	8	Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake)
5	5.0	0	11	Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Express 3.0
6	5.0	0	21	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC, 8GB 128-bit GDDR6, PCI Express 4.0
7	5.0	0	22	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC, 4GB 128-bit GDDR5, PCI Express x16 3.0
8	5.0	0	25	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming, 8GB 128-bit GDDR6, PCI Express 4.0
9	5.0	0	28	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti, 6GB 192-bit GDDR6, PCI Express x16 3.0
10	5.0	0	40	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare, S-STRX4, AMD TRX40, 256GB DDR4 para AMD

Fig 4.2.0 Top 10 productos con mejor puntuación

Productos con peor puntuación				
Rank	Puntuación	Devoluciones	ID	Nombre
1	0	0	9	Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)
2	0	0	14	Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB 64-bit GDDR3, PCI Express 2.0
3	0	0	15	Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0
4	0	0	16	Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming, 6GB 192-bit GDDR6, PCI Express 3.0
5	0	0	19	Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile, 4GB 128-bit GDDR5, PCI Express 3.0 x16
6	0	0	20	Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC, 8 GB 256 bit GDDR6, PCI Express x16 3.0
7	0	0	23	Tarjeta de Video MSI Radeon X1550, 128MB 64 bit GDDR2, PCI Express x16
8	0	0	24	Tarjeta de Video PNY NVIDIA GeForce RTX 2080, 8GB 256-bit GDDR6, PCI Express 3.0
9	0	0	26	Tarjeta de Video VisionTek AMD Radeon HD 5450, 1GB DDR3, PCI Express x16 2.1
10	0	0	27	Tarjeta de Video VisionTek AMD Radeon HD5450, 2GB GDDR3, PCI Express x16

Fig 4.2.1 Top 10 productos con peor puntuación

Productos con más devoluciones y peor puntuación				
Rank	Puntuación	Devoluciones	ID	Nombre
1	1.833	3	31	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
2	1.0	1	17	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0
3	1.0	1	45	Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel
4	2.0	1	46	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel
5	4.142	1	29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
6	4.230	1	2	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth
7	4.72	1	54	SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm
8	0	0	9	Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)
9	0	0	14	Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB 64-bit GDDR3, PCI Express 2.0
10	0	0	15	Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0

Fig 4.2.2 Top 10 productos con más devoluciones y peor puntuación

4.3 Reporte de ventas

```
-----REPORTE DE VENTAS-----
Los ingresos anuales totales son: $737916
Las ventas promedio por mes son: $61493

Meses con mas ventas
Rank  Mes  Ventas
01    ABR  $191066
02    MAR  $162931
03    ENE  $117738
04    FEB  $107270
05    MAY  $91936
06    JUN  $36949
07    JUL  $26949
08    AGO  $3077
09    SEP  $0
10    OCT  $0
11    NOV  $0
12    DIC  $0
```

Fig 4.3.0 Ingresos anuales, promedio mensual y ventas por mes ordenadas de mayor a menor

5. Conclusión

De acuerdo con los resultados obtenidos en Productos más vendidos y productos rezagados, la mayoría de los productos más vendidos también se encuentran en los productos más buscados, por lo podrían considerarse las *Milk Cows*¹ y LifeStore debería enfocarse en tener estos productos disponibles para venta. En contraste, los productos menos vendidos suelen estar dentro de los productos menos buscados, de modo que es recomendable que LifeStore no compre más o en gran cantidad porque no hay mucha demanda de estos y estarían acumulado espacio en los almacenes.

Después, al analizar los resultados obtenidos en Productos por reseña en el servicio, se puede observar que hay varios productos con la puntuación promedio más alta (5.0) o más baja (0) y no coinciden con los productos más y menos vendidos. Esto se puede deber a que como se obtiene el promedio, la calificación de un producto con más reseñas tiende a fluctuar de los valores “perfectos” porque puede haber más variedad en lo riguroso que es la persona calificándolo, pero esto hace que después de muchas reseñas, su calificación promedio se vuelva “estable”. Y, por tanto, una propuesta para mejorar el reporte es que también se tome en consideración el número de reseñas por producto.

Ahora, si se toma en cuenta el número de devoluciones y la calificación promedio, surgen dos casos de análisis: a) Hay productos con devoluciones y calificación baja, productos que LifeStore debería cambiar de estrategia de ventas y marketing porque representan un riesgo para la imagen de la tienda; y b) Hay productos con devoluciones y calificación alta, productos en los que LifeStore debe implementar acciones para asegurarse que los productos que distribuye estén en las mejores condiciones y también proveer un servicio al cliente de calidad en el proceso de devolución para mantener su lealtad.

Finalmente, en el reporte de ventas se puede observar que en el primer cuatrimestre del año se tuvieron las mejores ventas del año con un máximo en abril. Después hay un descenso constante en el segundo cuatrimestre y en el ultimo las ventas son nulas. Para estos casos, se requiere analizar si en los años anteriores hubo ciclos similares y descartar que sea un fenómeno único del año. Sin embargo, también puede significar que los datos no están completos (ya sea porque no se han reportado aún o porque se perdieron), lo que es un problema para la solución que se obtuvo porque no sería representativa de la realidad.

¹ una fuente de ingresos fáciles de obtener; empresa rentable