



EMTECH

Emerging Technologies Institute

Violeta Guadalupe Valdivia Zaragoza
Autora

Índice

I.	Introduccion.....	Pág. 3
II.	Definición del código	Pág. 4
III.	Solución al problema	Pág. 34
IV.	Conclusiones	Pág. 35

Introducción

La empresa LifeStore, es una empresa que vende una infinidad de artículos en su tienda virtual. En el último inventario que realizaron, se percataron de que tienen una gran cantidad de productos almacenados los cuales, se han ido acumulando al paso de los días. Además, observaron una reducción en sus ventas en el bimestre que acaba de terminar, esto ha generado grandes pérdidas para la empresa ya que no solo tienen una acumulación de productos, sino que, además, las búsquedas de los productos más importantes han disminuido drásticamente.

Es por ello que han acudido a nosotros para solicitar un análisis de la rotación de productos en donde piden identificar los siguientes elementos:

1. Productos más vendidos y productos rezagados a partir del análisis de las categorías con menores ventas y categorías con menores búsquedas.
2. Productos por reseña en el servicio a partir del análisis de categorías con mayores ventas y categorías con mayores búsquedas.
3. Sugerir una estrategia de productos a retirar del mercado, así como sugerencia de cómo reducir la acumulación de inventario considerando los datos de ingresos y ventas mensuales.

Definición del código.

1. Listas

En el encabezado del código se mandarán llamar las listas que proporcione el cliente con los datos de sus productos, información sobre las ventas y búsquedas realizadas por parte de los usuarios. Esto nos permitirá poder realizar un análisis detallado de la información a partir de los criterios establecidos por el cliente.

Ilustración 1 Código para mandar llamar las listas

```
4  #Mandar llamar a las listas
5  from lifestore_file import lifestore_products
6  from lifestore_file import lifestore_sales
7  from lifestore_file import lifestore_searches
```

2. Declaración de variables

En la parte inferior siguiendo la continuidad de donde se declararon las listas, se declararán las variables, que se utilizarán en todo el código. Las cuales almacenaran datos y se usaran en su respectivo apartado dentro del código.

Variables para el funcionamiento del Login

Para el correcto funcionamiento del Login se declararán 3 variables las cuales se describen a continuación:

- **Nombre_usuario = "admin"**

Almacenara el nombre de usuario el cual es "admin".

- **Contraseña = "123"**

Almacenara la contraseña para ingresar al sistema y su valor es "123".

- **Numero_intentos:**

Es un contador el cual se ira autoincrementando según se requiera. Y llevara el control de los intentos realizados por el usuario para ingresar al sistema.

Ilustración 2 Variables utilizadas para el funcionamiento del Login

```
# Declaración de variables que almacenarán información para el funcionamiento del Login
# Variable que almacenará el nombre del usuario
nombre_usuario = "admin"

# Variable que almacenará la clave de la contraseña
contrasena = "123"

# Variable que almacenará el número de intentos permitidos para ingresar al sistema
numero_intentos = 1
```

Variables utilizadas para mostrar los productos más vendidos

A continuación, se describen las variables que se utilizarán para el funcionamiento del apartado de los productos más vendidos:

- **Lista_ventas_tot= []**

Es una lista vacía la cual guardará los datos totales de las ventas.

- **Lista_ventas_ord= []**

Esta lista se encontrará vacía al momento de la declaración e irá guardando los productos según la coincidencia que se tenga y los pondrá de manera ordenada.

- **Índice_ventas_total=0**

Es un contador que llevará el registro de las veces que se repite un producto dentro de la lista_ventas_total.

- **Contador_productos_ord = 0**

Es un contador que llevará un registro de las veces que se mostrarán los productos más vendidos.

Ilustración 3 Variables utilizados en productos más vendidos

```
#Declaración de variables que se utilizarán para mostrar los productos más vendidos
# Lista que almacenará los datos totales de las ventas
lista_ventas_tot = []

# Lista que ordenará los datos que se vayan ingresando
lista_ventas_ord = []

# Contador
indice_ventas_tot = 0

#variable que llevará el conteo de las veces que se mostrarán los productos más vendidos
contador_productos_ord=0
```

Variables utilizadas para mostrar los productos con más búsquedas

Para este apartado se declararon 4 variables las cuales se describen a continuación:

- **Lista_búsquedas_tot = []**
Lista vacía que realizara la búsqueda total de productos.
- **Lista_búsqueda_ord = []**
Lista vacía que ordenara los datos que se vayan iterando y coincidan con la condición que se le imponga.
- **Índice_búsqueda_tot=0**
Es un contador que llevara el registro de las veces que se repite un producto dentro de la lista_búsquedas_total.
- **Cont_búsquedas_ord =0**
Variable que llevara el conteo de las veces que se mostraran los productos con más búsquedas.

Ilustración 4 Variables utilizadas para los productos más buscados.

```
#Declaración de variables que se utilizarán para mostrar los productos con mas búsquedas
# Lista que almacenara los datos totales de las búsquedas
lista_búsquedas_tot = [ ]

# lista que ordenara los datos que se vayan ingresando
lista_búsquedas_ord = [ ]

# contador
indice_búsquedas_tot = 0

#variable que llevará el conteo de las veces que se mostraran los productos mas buscados
cont_búsquedas_ord = 0
```

Variables utilizadas para mostrar los productos menos vendidos

- **lista_sale_total = []**
Almacenará los datos totales de los productos con menos ventas.
- **lista_sales_clasif = []**
Lista que ordenara los datos que se vayan ingresando.
- **index = 0**

Contador inicializado en cero el cual realizara el conteo de las veces que se repite un producto.

- `index_sales_total = 0`

Contador el total de productos con menos ventas.

- `contador_búsquedas_ord=1`

variable que llevará el conteo de las veces que se mostrará en pantalla los productos menos comprados.

Ilustración 5 Variables para productos con menos ventas

```
#Declaración de variables que contienen los productos con menos ventas
# Lista que almacenara los datos totales de los productos con menor venta
lista_sale_total = []

# Lista que ordenara los datos que se vayan ingresando
lista_sales_clasif = []

#contador
index = 0

#contador de ventas totales
index_sales_total = 0

#variable que llevará el conteo de las veces que se mostrarán los productos menos comprados
contador_búsquedas_ord=1
```

Variables utilizadas para mostrar los productos con menos búsquedas

- `lista_menorBusq_tot = []`

Lista que almacenara los datos totales de las búsquedas.

- `lista_menorBus_ord = []`

Lista que ordenara los datos que se vayan ingresando.

- `indice_menorBusq_tot = 0`

contador que llevara la cuenta de las veces que se repite un producto con menor búsqueda.

- `cont_menorBusqu_ord = 1`

Variable que llevará el conteo de las veces que se mostrarán los productos menos buscados

Ilustración 6 Variables utilizadas para determinar los productos con menor búsquedas

```
# Lista que almacenara los datos totales de las búsquedas
lista_menorBusq_tot = []

# Lista que ordenara los datos que se vayan ingresando
lista_menorBus_ord = []

# contador
indice_menorBusq_tot = 0

# variable que llevará el conteo de las veces que se mostrarán los productos menos buscados
cont_menorBusqu_ord = 1
```

Variables utilizadas para mostrar los productos con menos búsquedas

- lista_resenas_tot = []

Lista que almacenara los datos totales de las ventas.

- lista_resenas_ord = []

Lista que ordenara los datos que se vayan ingresando.

- indice_resenas_tot = 0

Realizara el conteo de las veces que un producto tiene una reseña alta.

- contador_puntaje_ord=1

Contador de las veces que se mostrará en pantalla la lista con los productos de mejor calificación

Ilustración 7 Variables utilizadas para determinar los productos con mejor calificación

```
#Declaración de variables que se utilizarán para identificar los productos con mejor calificación
# Declaración de variables
# Lista que almacenara los datos totales de las ventas
lista_resenas_tot = []

# Lista que ordenara los datos que se vayan ingresando
lista_resenas_ord = []

# contador
indice_resenas_tot = 0

#contador
contador_puntaje_ord=1
```


3. Ingresar al Sistema

El primer paso una vez que inicia la ejecución del programa es que el usuario ingrese al sistema, para ello se le mostrara un mensaje de bienvenida y otro donde se le advierte que solo tiene 3 intentos para ingresar. Cabe mencionar que después de los 3 intentos fallidos el programa se detendrá. A continuación, se describe cada una de las líneas de código utilizadas:

- **Print("-----Sistema LifeStore-----")**

Mostrará en pantalla un mensaje.

- **Print ("Solo tienes 3 intentos para ingresar \n")**

Mostrará un mensaje para el usuario indicándole que solo tiene 3 oportunidades para ingresar al sistema \n indica un salto de línea.

- **While numero_intento<=3:**

Mediante un ciclo While se pondrá la condición: "mientras numero_intento sea menor o igual a 3" se ejecutará lo que está dentro del bloque.

- **Intento_contrana = input ("Ingresa el nombre de Usuario:")**

Con esta línea de código, se pedirá que se ingrese el nombre de usuario y este dato se guardara dentro de la variable intento_contrasena.

- **intento_contrasena = input ("Ingresa la Contraseña: ")**

Se pedirá al usuario que ingrese un valor para lo contraseña y este último se guardará dentro de la variable intento_contrasena.

- **if intento_usuario == nombre_usuario and intento_contrasena == contrasena:**

Se pondrá una condición if la cual se interpreta de la siguiente forma: "Si intento_usuario es igual al valor de nombre_usuario y intento_contrasena es igual valor de contraseña" Si se cumple entonces se puede ejecutar el siguiente bloque de información que está dentro del if.

- **print("Datos correctos" \n)**

Se mostrará al usuario un mensaje indicándole que los datos ingresados son correctos.

- `print("-----Bienvenido al Sistema de Administración----- ")`

Aparecerá en pantalla un mensaje de bienvenida.

- **Break**

Sentencia para salirse del ciclo.

- **else:**

Si la condición `if (intento_usuario == nombre_usuario and intento_contraseña == contraseña)` no se cumple entonces se ejecutará el `else`.

Y se mostrará en pantalla un mensaje para el usuario diciéndole que “El usuario o contraseña no son correctos”.

- `Numero_intento = numero_intentos + 1`

Al no cumplirse la condición del `if` se comenzará con el conteo de intentos realizados es por ello que la variable `numero_intento` se le suma 1.

- **If numero_intentos==4:**

Si la variable `numero_intentos` tiene como valor 4 entonces el usuario habrá excedido las 3 oportunidades que tiene para ingresar al sistema y entonces se le mostrará el siguiente mensaje: “Has excedido el número de intentos” y lo sacará del sistema.

Ilustración 8 Código usado para validar el Login

```
# Mensaje de bienvenida
print("-----Sistema LifeStore----- ")

# Mensaje para el usuario
print("Solo tienes 3 intentos para ingresar \n ")

# Mientras número_intentos sea igual o menor a 3 se ejecutará el bloque dentro del while
while numero_intentos <= 3:

    # Pide los datos sobre el nombre de usuario
    intento_usuario = input("Ingresa el nombre de Usuario: ")

    # Pide los datos sobre la contraseña
    intento_contraseña = input("Ingresa la Contraseña: ")

    # Si el nombre de usuario ingresado es igual al usuario y la contraseña ingresada es igual a la contraseña
    if intento_usuario == nombre_usuario and intento_contraseña == contraseña:

        # Muestra el mensaje y hace un salto de línea
        print("Datos Correctos \n")

        # Mensaje para el usuario
        print("-----Bienvenido al Sistema de Administración----- ")

        # se sale del ciclo
        break
```

```

else:
    # Mensaje de error
    print("Usuario o Contraseña incorrectos \n")

    # Actualizar la variable sumandole 1 al valor
    numero_intentos = numero_intentos + 1

# Si se cumplen los intentos permitidos entonces...
if numero_intentos == 4:
    # Mensaje para el usuario
    print("Has excedido el número de intentos ")

```

Ilustración 9 Datos mostrados en consola, si son correctos se muestra el menú principal

```

Ingresa el nombre de Usuario: admin
Ingresa la Contraseña: 123
Datos Correctos

-----Bienvenido al Sistema de Administración-----
-----Menú Principal-----

Elige un número de las siguientes opciones:

1 Productos más vendidos y productos rezagados.
2 Productos por reseña en el servicio.
3 Total de ingresos y ventas promedio.

Seleccionaste la opción: |

```

Ilustración 10 Usuario o contraseña incorrectos

```

-----Sistema LifeStore-----
Solo tienes 3 intentos para ingresar

Ingresa el nombre de Usuario: soporte
Ingresa la Contraseña: 123
Usuario o Contraseña incorrectos

Ingresa el nombre de Usuario: soporte
Ingresa la Contraseña: 123
Usuario o Contraseña incorrectos

Ingresa el nombre de Usuario: soporte
Ingresa la Contraseña: 123
Usuario o Contraseña incorrectos

Has excedido el número de intentos

```

4. Menú principal y Sub-menús

Una vez que usuario ingrese al sistema se le mostrará un menú principal, en donde se le pedirá elegir un numero de las opciones siguientes:

1. Productos más vendidos y productos rezagados.
2. Productos por reseña de servicio.
3. Total, de ingresos y ventas.

Ilustración 11 Menú Principal ejecutado en la consola

```
-----Bienvenido al Sistema de Administración-----  
-----Menú Principal-----  
  
Elige un número de las siguientes opciones:  
  
1 Productos más vendidos y productos rezagados.  
2 Productos por reseña en el servicio.  
3 Total de ingresos y ventas promedio.  
  
Seleccionaste la opción: |
```

Cuando se elige una opción se desplegara un sub-menú para cada categoría principal

OPCIÓN 1: Productos más vendidos y productos rezagados

1. Listado de los productos con mayores ventas.
2. Listado de los productos con mayores búsquedas.
3. Listado de productos con menores ventas.
4. Productos con menos búsquedas.

Ilustración 12 Sub-menú de Productos más vendidos y productos rezagados

```
-----Productos más vendidos y productos rezagados-----  
Elige un número de las siguientes opciones:  
  
1 Listado de los productos con mayores ventas.  
2 Listado de los productos con mayores búsquedas.  
3 Listado de productos con menores ventas.  
4 Productos con menos búsquedas.  
  
La opción seleccionada es: |
```

OPCIÓN 2: Productos por reseña de servicio.

1. Listado de productos con mejores reseñas.
2. Listado de productos con peores reseñas.

Ilustración 13 Sub-menú Productos por reseña en el servicio

```
-----Listado de Productos por reseña en el servicio-----
Elige un número de las siguientes opciones:

Listado de productos con mejores reseñas:
Listado de productos con peores reseñas:

La opción seleccionada es: |
```

OPCIÓN 3

1. Total, de ingresos.
2. Ventas promedio mensual.
3. Ventas promedio anual.
4. Meses con más ventas al año.

Ilustración 14 Sub-menú para Total de ingresos y ventas promedio

```
Seleccionaste la opción: 3
-----Total de ingresos y ventas promedio-----

1 Total de ingresos
2 Ventas promedio mensual
3 Ventas promedio anual
4 Meses con más ventas al año

La opción seleccionada es: 3
```

Para mostrar el menú principal en pantalla se utilizaron las siguientes líneas de código las cuales

- `print("-----Menú Principal-----\n")`
- `print("Elige un número de las siguientes opciones:\n")`
- `print("1 Productos más vendidos y productos rezagados.")`
- `print("2 Productos por reseña en el servicio.")`
- `print("3 Total de ingresos y ventas promedio.\n")`

Permiten mostrar las opciones del menú mediante un mensaje en pantalla.

- `opcion_menu_principal = input("Seleccionaste la opción: ")`

Permite al usuario ingresar un valor.

5. Validación del menú y sub-menús

Un aspecto importante de mencionar dentro del menú y sub-menús es la validación, esta se realizó con la finalidad de que los usuarios no ingresaran valores incorrectos. Para ello se utilizó un ciclo while cabe mencionar que para cada una de las opciones presentadas en el menú principal se siguió el mismo procedimiento, a continuación, se describe de manera detallada cada una de las líneas de código utilizadas para la validación de la primera opción del menú principal.

- While validar_menu_prin !=1:

Mientras que validar_menu_prin sea diferente o igual a uno se puede ejecutar el bloque de código dentro del ciclo while.

- if opcion_menu_principal == "1":

Si la opción ingresada cuando se mostró el menú principal es la 1 entonces se cumple la condición.

- print("-----Productos más vendidos y productos rezagados-----")
- print("Elige un número de las siguientes opciones: \n")
- print(" 1 Listado de los productos con mayores ventas.")
- print(" 2 Listado de los productos con mayores búsquedas.")
- print(" 3 Listado de productos con menores ventas.")
- print(" 4 Productos con menos búsquedas.\n")

Al ser correcta la condición se le mostrará al usuario un segundo sub-menú el cual le permitirá visualizar diferentes listas de análisis.

- subMenu_produc_vent = input("La opción seleccionada es: ")

Se mostrará al usuario un mensaje en donde se le pedirá ingresar un valor acorde a la opción del sub-menú que desea consultar.

- validar_menu_prin = 1

Se validará el dato ingresado en el menú principal y la variable validar_menu_prin se pondrá en verdadero. En pocas palabras significa que el dato ingresado es correcto.

Pudiera existir la posibilidad de que por cualquier motivo el usuario se puede equivocar al momento de ingresar el valor para seleccionar una opción ya sea del menú o sub-menú para ello se le enviara un mensaje en donde se le hará saber que la opción ingresada es invalida y posteriormente se le volverá a mostrar el menú con las opciones para que nuevamente ingrese un valor.

Ilustración 15 Código utilizado para validar el menú y sub-menús

```
# Mensaje para el usuario
print("-----Menú Principal-----\n")

# Mensaje para el usuario donde se mostrara el menu principal
print("Elige un número de las siguientes opciones:\n")
print("1 Productos más vendidos y productos rezagados.")
print("2 Productos por reseña en el servicio.")
print("3 Total de ingresos y ventas promedio.\n")

# Pide al usuario ingresar la opción a elegir del menú
opcion_menu_principal = input("Seleccionaste la opción: ")

# Validar que los datos ingresados son verdaderos del menu principal
while validar_menu_prin != 1:

    #si la opción seleccionada del menú es 1 entonces...
    if opcion_menu_principal == "1":

        # Mensaje para el usuario
        print(
            "-----Productos más vendidos y productos rezagados-----"
        )

        # Se mostrará un nuevo sub-menú al usuario
        print("Elige un número de las siguientes opciones: \n")
        print(" 1 Listado de los productos con mayores ventas.")
        print(" 2 Listado de los productos con mayores búsquedas.")
        print(" 3 Listado de productos con menores ventas.")
        print(" 4 Productos con menos búsquedas.\n")

        #El usuario ingresará una opción
        subMenu_produc_vent = input("La opción seleccionada es: ")

        #La bandera se pondra en verdadero
        validar_menu_prin = 1

        #Validar los datos de sub-menu Productos más vendidos y productos rezagados
        while validar_submenu_venta != 1:

            #Si el valor ingresado por el usuario es 1
            if subMenu_produc_vent == "1":
```

```

#Filtrado de datos para mostrar los productos con mas ventas
print("Poner todo el cofigo para productos con mas ventas ")

#La bandera se pondra en verdadero
validar_submenu_venta = 1

#Si la opción seleccionada es 2 entonces...
elif subMenu_produc_venta == "2":

    #Filtrado de datos para mostrar los productos con mas búsquedas
    print("Poner todo el codigo para productos con mas busquedas ")

    #La bandera se pondra en verdadero
    validar_submenu_venta = 1

#Si la opción seleccionada es 3 entonces...
elif subMenu_produc_venta == "3":

    #Filtrado de datos para mostrar los productos con menos ventas
    print("Poner todo el cofigo para productos con menos ventas ")

    #La bandera se pondra en verdadero
    validar_submenu_venta = 1

#Si la opción seleccionada es 4 entonces...
elif subMenu_produc_venta == "4":

    #Filtrado de datos para mostrar los productos con menos búsquedas
    print(
        "Poner todo el cofigo para productos con menos busquedas ")

    #La bandera se pondra en verdadero
    validar_submenu_venta = 1

#Si la opción ingresada es incorrecta entonces...
else:

    #Mensaje para el usuario
    print("Opción incorrecta: \n")

```

```

# Mensaje para el usuario
print(
    "-----Productos más vendidos y productos rezagados-----"
)

# Se mostrará un nuevamente el sub-menú al usuario
print("Elige un número de las siguientes opciones: \n")
print(" 1 Listado de los productos con mayores ventas.")
print(" 2 Listado de los productos con mayores búsquedas.")
print(" 3 Listado de productos con menores ventas.")
print(" 4 Productos con menos búsquedas.\n")

#El usuario ingresara nuevamente una opción
subMenu_produc_venta = input("La opción seleccionada es: ")

#Si la opción seleccionada en el menú principal es 2 entonces
elif opcion_menu_principal == "2":

    #Mensaje para el usuario
    print(
        "-----Listado de Productos por reseña en el servicio-----"
    )

    # Se mostrará un nuevo sub-menú al usuario
    print("Elige un número de las siguientes opciones: \n")
    print("Listado de productos con mejores reseñas: ")
    print("Listado de productos con peores reseñas:\n ")

    #El usuario ingresara una opción
    subMenu_produc_resena = input("La opción seleccionada es: ")

    #La bandera se pondra en verdadero
    validar_menu_prin = 1

    #validar las opciones del sub-menu de los Productos por reseña en el servicio
    while validar_submenu_resena != 1:

        #Si la opción seleccionada en el sub-menu es 1
        if subMenu_produc_resena == "1":

            #Filtrado de datos para mostrar los productos con mejores reseñas
            print("Poner todo el cofigo para productos con mas reseñas ")

```



```

        #La bandera se pondrá en verdadero
        validar_submenu_resena = 1

#Si la opción seleccionada en el sub-menu es 2
elif subMenu_produc_resena == "2":

    #Filtrado de datos para mostrar los productos con peores reseñas
    print("Poner todo el código para productos con mas reseñas ")

    #La bandera se pondrá en verdadero
    validar_submenu_resena = 1

#Si la opción es incorrecta entonces...
else:

    #Mensaje para el usuario
    print("Opción incorrecta: \n")

    #Mensaje para el usuario
    print(
        "-----Listado de Productos por reseña en el servicio-----"
    )

    # Se mostrará nuevamente el sub-menú al usuario
    print("Elige un número de las siguientes opciones: \n")
    print("Listado de productos con mejores reseñas: ")
    print("Listado de productos con peores reseñas:\n ")

    #El usuario ingresara nuevamente una opción
    subMenu_produc_resena = input("La opción seleccionada es: ")

#Si la opción seleccionada en el menú principal es 3 entonces
elif opcion_menu_principal == "3":

    #Mensaje para el usuario
    print(
        "-----Total de ingresos y ventas promedio----- \n"
    )
    print("1 Total de ingresos")
    print("2 Ventas promedio mensual")
    print("3 Ventas promedio anual")
    print("4 Meses con más ventas al año \n")

    #El usuario ingresara un valor

```

```

subMenu_produc_ingresos = input("La opción seleccionada es: ")

```

```

#La bandera se pondra en verdadero
validar_menu_prin = 1

```

```

while validar_submenu_ingresos != 1:

```

```

    #Si la opción seleccionada en el sub-menu es 1
    if subMenu_produc_ingresos == "1":

        #mensaje para el usuario
        print("LO SIENTO NO LOGRE TERMINAR ESTA PARTE :( ")

        #La bandera se pondra en verdadero
        validar_submenu_ingresos = 1

```

```

    #Si la opción seleccionada en el sub-menu es 2
    elif subMenu_produc_ingresos == "2":

        #mensaje para el usuario
        print("LO SIENTO NO LOGRE TERMINAR ESTA PARTE :( ")

        #La bandera se pondra en verdadero
        validar_submenu_ingresos = 1

```

```

    #Si la opción seleccionada en el sub-menu es 3
    elif subMenu_produc_ingresos == "3":

        #mensaje para el usuario
        print("LO SIENTO NO LOGRE TERMINAR ESTA PARTE :( ")

        #La bandera se pondra en verdadero
        validar_submenu_ingresos = 1

```

```

    #Si la opción seleccionada en el sub-menu es 4
    elif subMenu_produc_ingresos == "4":

        #mensaje para el usuario
        print("LO SIENTO NO LOGRE TERMINAR ESTA PARTE :( ")

```

```

        #La bandera se pondra en verdadero
        validar_submenu_ingresos = 1

    else:

        #Mensaje para el usuario
        print("Opción incorrecta: \n")

        #Mensaje para el usuario
        print(
            "-----Total de ingresos y ventas promedio----- \n"
        )
        print("1 Total de ingresos")
        print("2 Ventas promedio mensual")
        print("3 Ventas promedio anual")
        print("4 Meses con más ventas al año \n")

        #El usuario ingresara un valor
        submenu_produc_ingresos = input("La opción seleccionada es: ")

# Si la opción seleccionada en el menú principal es incorrecta entonces...
else:
    #mensaje para el usuario
    print("Opción incorrecta: \n")

    # Mensaje para el usuario donde se mostrara nuevamente el menu principal
    print("Elige un número de las siguientes opciones:\n")
    print("1 Productos más vendidos y productos rezagados.")
    print("2 Productos por reseña en el servicio.")
    print("3 Total de ingresos y ventas promedio.\n")

    # Pide al usuario ingresar la opción a elegir del menú nuevamente
    opcion_menu_principal = input("Seleccionaste la opción: ")

```

6. Filtrado de datos

Para realizar este apartado de filtrado de datos de cada una de las opciones de los sub-menús se utilizó el bucle for para hacer iteraciones a la lista, un aspecto importante de mencionar es que se crearon listas “temporales” que guardarían los datos que se extrajeron de las iteraciones a continuación se muestran imágenes del código con los pasos que se siguieron en donde cada línea de código esta comentareada para un mejor entendimiento.

7. Filtrado de datos para los productos con mayores ventas

```

# Filtrado de Sales
# Se iniciara la iteración en la lista lifestore_sales:
for linea_lista in lifestore_sales:

    # Si se encuentra un producto se pondra en falso
    producto_encontrado = False

    # la variable productos se iguala al valor que valla teniendo la variable
    producto = linea_lista[1]

    # Se hará una iteración en lista_ventas_tot
    for linea_ventas_tot in lista_ventas_tot:

        # Si producto es igual al valor de linea_ventas_tot en la posición cero entonces...
        if producto == linea_ventas_tot[0]:
            # Se autoincrementara linea_ventas_tot
            linea_ventas_tot[1] = linea_ventas_tot[1] + 1

            # producto se pondra en verdadero
            producto_encontrado = True

    # Si producto_encontrado == igual a falso entonces...
    if producto_encontrado == False:
        # Se agregará el nuevo producto a lista_ventas_tot
        lista_ventas_tot.append([producto, 1])

# Se hará una iteración en listas_ventas_tot
for linea_ventas_tot in lista_ventas_tot:

```

```

# Producto se igualara al valor de linea_ventas_tot en la posición zero
producto = linea_ventas_tot[0]

# Cantidad se igualara al valor de linea_ventas_tot en la posición 1
cantidad = linea_ventas_tot[1]

# Si el indice_ventas_tot es igual a cero entonces...
if indice_ventas_tot == 0:

    # Se insertara a listas_ventas_ord el producto y cantidad
    lista_ventas_ord.insert(0, [producto, cantidad])

# Si la condición no se cumple entonces....
else:

    # encontrado se pondra en falso
    encontrado = False

    # indice se reiniciara a cero
    indice = 0

    # Se hará una iteración en lista_ventas_ord
    for linea_ventas_ord in lista_ventas_ord:

        # Si cantidad es mayor o igual a linea_ventas_ord en la posición 1 entonces...
        if cantidad >= linea_ventas_ord[1]:
            # Se insertaran datos a la lista_búsquedas_ord
            lista_ventas_ord.insert(indice, [producto, cantidad])

        # Encontrado será verdadero
        encontrado = True

        # Terminará el bucle
        break

    # indice se autoincrementara de 1 en 1
    indice = indice + 1

    # Si encontrado es igual a falso entonces...
    if encontrado == False:
        # Se agregarán los campos producto y cantidad a listas_ventas_ord
        lista_ventas_ord.append([producto, cantidad])

# la variable indice_ventas_total se autoincrementara de 1 en 1
indice_ventas_tot = indice_ventas_tot + 1

# Mensaje para el usuario
print("-----Lista ordenada de los productos con mas ventas -----\\n \\n ")

# Se hará una iteración en la la lista lista_ventas_ord
for linea_ventas_ord in lista_ventas_ord:

    # Se hará una iteración en la lifestore_products:
    for linea_productos in lifestore_products:

        # Si los datos de la lista linea_ventas_ord en el indice [0] son iguales que
        # los datos de la lista linea_productos en el indice[0]:
        if linea_ventas_ord[0] == linea_productos[0]:
            # Se mostrara la siguiente información al usuario
            print(contador_productos_ord, "-", "ID:", linea_productos[0])

```

```

        print("Nombre del Producto: ", linea_productos[i])

        print("Obtuvo: ", linea_ventas_ord[i], "venta(s)\n")

# Si contador_productos_ord es mayor a 50 entonces...
if contador_productos_ord >= 50:
    # Se detendra el conteo.
    break

# Se ira sumando uno ala variable contador_productos_ord cada que se ejecute
contador_productos_ord = contador_productos_ord + 1

# La bandera se pondra en verdadero
validar_submenu_venta = 1

```

Ilustración 16 Resultado del análisis

```

-----Lista ordenada de los productos con mas ventas -----

1 - ID: 54
Nombre del Producto:  SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm
Obtuvo:  50 venta(s)

2 - ID: 3
Nombre del Producto:  Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth
Obtuvo:  42 venta(s)

3 - ID: 5
Nombre del Producto:  Procesador Intel Core i3-9100F, 5-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)
Obtuvo:  28 venta(s)

4 - ID: 42
Nombre del Producto:  Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
Obtuvo:  18 venta(s)

5 - ID: 57
Nombre del Producto:  SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm
Obtuvo:  15 venta(s)

6 - ID: 29
Nombre del Producto:  Tarjeta Madre ASUS micro ATX TUF B450H-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
Obtuvo:  14 venta(s)

```

8. Filtrado de datos para los artículos con mayor búsqueda

```
# Se iniciara la iteración en la lista lifestore_sales:
for linea_lista_busq in lifestore_searches:

    # Si se encuentra un artículo se pondra en falso
    articulo_ubicado = False

    # La variable articulo se iguala al valor que valla teniendo la variable linea_lista_busq
    articulo = linea_lista_busq[1]

    # Se hará una iteración en lista_busquedas_tot
    for linea_busquedas_tot in lista_busquedas_tot:

        # Si el articulo es igual al valor de linea_busquedas_tot en la posición cero entonces...
        if articulo == linea_busquedas_tot[0]:
            # Se autoincrementara linea_busquedas_tot
            linea_busquedas_tot[1] = linea_busquedas_tot[1] + 1

            # articulo se pondra en verdadero
            articulo_ubicado = True

    # Si producto_encontrado es igual a falso entonces...
    if articulo_ubicado == False:
        # Se agregara el nuevo articulo a lista_busquedas_tot
        lista_busquedas_tot.append([articulo, 1])

# Se hará una iteración de lista_busquedas_tot
for linea_busquedas_tot in lista_busquedas_tot:

    # articulo se igualará al valor de linea_busquedas_tot en la posición cero
    articulo = linea_busquedas_tot[0]

    # proporcion se igualara a linea_busquedas_tot en la posición 1
    proporcion = linea_busquedas_tot[1]

    # Si indice_busquedas_tot es igual a cero entonces...
    if indice_busquedas_tot == 0:

        # Se insertara a lista_busquedas_ord los valores de articulo y proporcion
        lista_busquedas_ord.insert(0, [articulo, proporcion])

    # si la condición no se cumple entonces...
    else:

        # ubicado se pondra en falso
        ubicado = False

        # indice_busq se reiniciara a cero
        indice_busq = 0

        # Se hará una iteración a lista_busquedas_ord
        for linea_busquedas_ord in lista_busquedas_ord:

            # if proporcion es mayor o igual que linea_busquedas_ord en la posición 1 entonces...
            if proporcion >= linea_busquedas_ord[1]:
                # Se insertaran datos ala lista_busquedas_ord.
                lista_busquedas_ord.insert(indice_busq, [articulo, proporcion])

            # ubicado será verdadero
            ubicado = True
```

```

        # Terminará el bucle
        break

    # indice_busq se autoincrementará de 1 en 1
    indice_busq = indice_busq + 1

    # Si ubicado es igual a falso entonces...
    if ubicado == False:
        # Se agregarán los campos artículo y proporción a listas_busquedas_ord
        lista_busquedas_ord.append([articulo, proporción])

    # La variable indice_busquedas_tot se autoincrementará de 1 en 1
    indice_busquedas_tot = indice_busquedas_tot + 1

# La bandera se pondrá en verdadero
validar_submenu_venta = 1

# Mensaje para el usuario
print("-----Lista ordenada de los productos con mas búsquedas -----\\n \\n ")

# Se hará una iteración en la lista lista_ventas_ord
for linea_busquedas_ord in lista_busquedas_ord:

    # Se hará una iteración en la lifestore_products:
    for linea_producto_busq in lifestore_products:

        # Si los datos de la lista linea_ventas_ord en el índice [0] son iguales que
        # los datos de la lista linea_productos en el índice[0]:
        if linea_busquedas_ord[0] == linea_producto_busq[0]:

```

```

        # Se mostrará la información al usuario
        print(cont_busquedas_ord, "ID:", linea_producto_busq[0])

        print("Nombre del Producto: ", linea_producto_busq[1])

        print("Obtuvo: ", linea_busquedas_ord[1], "busqueda(s)\\n")

    # Si contador_productos_ord es mayor a 50 entonces...
    if cont_busquedas_ord >= 100:
        # Se detendrá el conteo
        break

    cont_busquedas_ord = cont_busquedas_ord + 1

# Si la opción seleccionada es 3 entonces...
elif submenu_produc_vent == "3":

    # Se hará una iteración
    for linea_lista_sale in lifestore_sales:

        # la variable articulo Ubicado estará en falso
        articulo Ubicado = False

        # la variable articulo se iguala al valor que vaya teniendo la variable:
        # linea_lista_sale en índice 1
        articulo = linea_lista_sale[1]

```

Ilustración 17 Resultado del análisis en pantalla

```
La opción seleccionada es: 2
-----Lista ordenada de los productos con mas búsquedas -----

1 ID: 54
Nombre del Producto: SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm
Obtuvo: 263 búsqueda(s)

2 ID: 57
Nombre del Producto: SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm
Obtuvo: 107 búsqueda(s)

3 ID: 29
Nombre del Producto: Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
Obtuvo: 60 búsqueda(s)

4 ID: 3
Nombre del Producto: Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth
Obtuvo: 55 búsqueda(s)

5 ID: 4
Nombre del Producto: Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire
Obtuvo: 41 búsqueda(s)

6 ID: 85
Nombre del Producto: Logitech Audifonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul
Obtuvo: 35 búsqueda(s)

7 ID: 67
Nombre del Producto: TV Monitor LED 24TLE20S-PU 24, HD, Widescreen, HDMI, Negro
Obtuvo: 32 búsqueda(s)
```

9. Filtrado de datos productos con menores ventas

```
# Se hará una iteración
for linea_lista_sale in lifestore_sales:

    # la variable articulo_ubicado estará en falso
    articulo_ubicado = False

    # la variable articulo se iguala al valor que valla teniendo la variable
    # linea_lista_sale en indice 1
    articulo = linea_lista_sale[1]

    # Se hará un nuevo recorrido en la lista pero lista_sale_total
    for linea_sale_total in lista_sale_total:

        # Si producto es igual a linea_sale_total en la posición 0 entonces
        if articulo == linea_sale_total[0]:
            # Si se encuentra un valor igual se sumará
            linea_sale_total[1] = linea_sale_total[1] + 1

            # Si se encuentra un valor igual se pondrá en verdadero
            articulo_ubicado = True

    # Si no hay un valor igual, entonces
    if articulo_ubicado == False:
        # el articulo se añadirá a la lista_sale_total encontrado
        lista_sale_total.append([articulo, 1])

# Se realizará otra iteración
for linea_sale_total in lista_sale_total:
```



```

# articulo se igualará al valor de linea_sale_total en el indice[0]
articulo = linea_sale_total[0]

# proporcion se igualará al valor de linea_sale_total en el indice[1]
proporcion = linea_sale_total[1]

# Si index_sales_total == 0 entonces...
if index_sales_total == 0:

    # Se insertarán los datos encontrados en articulo y proporcion
    lista_sales_clasif.insert(0, [articulo, proporcion])

else:

    # Si no se cumple la condición ubicado se pondrá en falso
    ubicado = False

    # El valor de index se pondrá en cero
    index = 0

    # Se hará una iteración de la lista_sales_clasif
    for linea_sale_clasif in lista_sales_clasif:

        # Si proporcion es mayor o igual a linea_sale_clasif en el indice[1]:
        if proporcion <= linea_sale_clasif[1]:
            # Se insertarán los valores de articulo y proporcion al lista_sales_clasif
            lista_sales_clasif.insert(index, [articulo, proporcion])

            # ubicado se pondra en verdadero
            ubicado = True

```

```

        # Se detendrá
        break

    # index se autoincrementará
    index = index + 1

# si ubicado=falso
if ubicado == False:
    # Se añadirá el nuevo artículo
    lista_sales_clasif.append([articulo, proporcion])

# index_sales_total se autoincrementará
index_sales_total = index_sales_total + 1

# Mensaje para el usuario
print("-----Lista ordenada de los productos con menos ventas -----\\n \\n ")

# Se hará una iteración en lista_sales_clasif
for linea_sale_clasif in lista_sales_clasif:

    # Se hará una iteración en la lifestore_products:
    for linea_product in lifestore_products:

        # Si los datos de la lista linea_sale_clasif en el índice [0] son iguales que
        # los datos de la lista linea_product en el índice[0]:
        if linea_sale_clasif[0] == linea_product[0]:
            # Se mostrará la información al usuario
            print(contador_busquedas_ord, "ID:", linea_product[0])
            print("Nombre del Producto: ", linea_product[1])

```

```

        print("Obtuvo: ", linea_sale_clasif[1], "venta(s)\\n")

    # Si contador_productos_ord es mayor a 50 entonces..
    if contador_busquedas_ord >= 50:
        # Se detendrá el conteo
        break

    # contador_busquedas_ord se autoincrementará
    contador_busquedas_ord = contador_busquedas_ord + 1

# La bandera se pondrá en verdadero
validar_submenu_venta = 1

```

Ilustración 18 Resultado del análisis en pantalla

```
La opción seleccionada es: 3
-----Lista ordenada de los productos con menos ventas -----

1 ID: 94
Nombre del Producto: HyperX Audifonos Gamer Cloud Flight para PC/PS4/PS4 Pro, Inalámbrico, USB, 3.5mm, Negro
Obtuvo: 1 venta(s)

2 ID: 89
Nombre del Producto: Cougar Audifonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro.
Obtuvo: 1 venta(s)

3 ID: 84
Nombre del Producto: Logitech Audifonos Gamer G332, Alámbrico, 2 Metros, 3.5mm, Negro/Rojo
Obtuvo: 1 venta(s)

4 ID: 67
Nombre del Producto: TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro
Obtuvo: 1 venta(s)

5 ID: 66
Nombre del Producto: TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Widescreen, Negro
Obtuvo: 1 venta(s)

6 ID: 68
Nombre del Producto: Kit Memoria RAM Corsair Dominator Platinum DDR4, 3200MHz, 16GB (2x 8GB), Non-ECC, CL16, XMP
Obtuvo: 1 venta(s)
```

10. Filtrado de datos productos con menores búsquedas

```
elif subMenu_produc_vent == "4":

    # Filtrado de datos
    # Se iniciara la iteración en la lista lifestore_sales:
    for linea_list_menorBusq in lifestore_searches:

        # Si se encuentra un articulo_menosBusq se pondra en falso
        art_menosBusq_ubicado = False

        # la variable articulo_menosBusq se iguala al valor que valla teniendo la variable linea_list_menorBusq
        articulo_menosBusq = linea_list_menorBusq[1]

        # Se hará una iteración en lista_menorBusq_tot
        for linea_menorBusq_tot in lista_menorBusq_tot:

            # Si el articulo_menosBusq es igual al valor de linea_menorBusq_tot en la posición cero entonces...
            if articulo_menosBusq == linea_menorBusq_tot[0]:
                # Se autoincrementara linea_menorBusq_tot
                linea_menorBusq_tot[1] = linea_menorBusq_tot[1] + 1

                # articulo_menosBusq se pondra en verdadero
                art_menosBusq_ubicado = True

        # Si producto_encontrado es igual a falso entonces...
        if art_menosBusq_ubicado == False:
            # Se agregara el nuevo articulo a lista_menorBusq_tot
            lista_menorBusq_tot.append([articulo_menosBusq, 1])

    # Se hará una iteración de lista_menorBusq_tot
    for linea_menorBusq_tot in lista_menorBusq_tot:
```

```

# articulo_menosBusq se igualará al valor de linea_menorBusq_tot en la posición cero
articulo_menosBusq = linea_menorBusq_tot[0]

# proporcion_menosBusq se igualará a linea_menorBusq_tot en la posición 1
proporcion_menosBusq = linea_menorBusq_tot[1]

# Si indice_menorBusq_tot es igual a cero entonces...
if indice_menorBusq_tot == 0:

    # Se insertará a lista_menorBus_ord los valores de articulo_menosBusq y proporcion_menosBusq
    lista_menorBus_ord.insert(0, [articulo_menosBusq, proporcion_menosBusq])

# si la condición no se cumple entonces...
else:

    # ubicado_menorBusq se pondrá en falso
    ubicado_menorBusq = False

    # indice_busq se reiniciará a cero
    indice_busq = 0

    # Se hará una iteración a lista_menorBus_ord
    for linea_búsquedas_ord in lista_menorBus_ord:

        # if proporcion_menosBusq es mayor o igual que linea_búsquedas_ord en la posición 1 entonces...
        if proporcion_menosBusq <= linea_búsquedas_ord[1]:
            # Se insertarán datos a la lista_menorBus_ord.
            lista_menorBus_ord.insert(indice_busq, [articulo_menosBusq, proporcion_menosBusq])

```

```

        # ubicado_menorBusq será verdadero
        ubicado_menorBusq = True

        # Terminará el bucle
        break

    # indice_busq se autoincrementará de 1 en 1
    indice_busq = indice_busq + 1

# Si ubicado_menorBusq es igual a falso entonces...
if ubicado_menorBusq == False:
    # Se agregarán los campos artículo y proporción a listas_busquedas_ord
    lista_menorBus_ord.append([articulo_menosBusq, proporcion_menosBusq])

# La variable indice_menorBusq_tot se autoincrementará de 1 en 1
indice_menorBusq_tot = indice_menorBusq_tot + 1

# La bandera se pondrá en verdadero
validar_submenu_venta = 1

# Mensaje para el usuario
print("-----Lista ordenada de los productos con menos búsquedas -----\\n \\n ")

# Se hará una iteración en la lista lista_ventas_ord
for linea_busquedas_ord in lista_menorBus_ord:

    # Se hará una iteración en la lifestore_products:
    for linea_producto_busq in lifestore_products:

```

```

        # Si los datos de la lista linea_ventas_ord en el índice [0] son iguales que
        # los datos de la lista linea_productos en el índice[0]:
        if linea_busquedas_ord[0] == linea_producto_busq[0]:
            # Se mostrará la información al usuario
            print(cont_menorBusqu_ord, "ID:", linea_producto_busq[0])

            print("Nombre del Producto: ", linea_producto_busq[1])

            print("Obtuvo: ", linea_busquedas_ord[1], "busqueda(s)\\n")

# Si contador_productos_ord es mayor a 50 entonces...
if cont_menorBusqu_ord >= 100:
    # Se detendrá el conteo
    break

cont_menorBusqu_ord = cont_menorBusqu_ord + 1

```

Ilustración 19 Resultado del análisis en pantalla

1 ID: 93	Nombre del Producto: Giga Audifonos con Micrófono GI18ADJ0818T-R0, Bluetooth, Alámbrico/Inalámbrico, 3.5mm, Rojo
2 ID: 88	Nombre del Producto: Giga Bocina Portátil BX800, Bluetooth, Inalámbrico, 2.1 Canales, 31W, USB, Negro
3 ID: 78	Nombre del Producto: Samsung Smart TV LED 43, Full HD, Widescreen, Negro
4 ID: 59	Nombre del Producto: SSD Samsung 860 EVO, 1TB, SATA III, M.2
5 ID: 45	Nombre del Producto: Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel
6 ID: 35	Nombre del Producto: Tarjeta Madre Gigabyte micro ATX Z390 M GAMING, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel
7 ID: 27	Nombre del Producto: Tarjeta de Video VisionTek AMD Radeon HD5450, 2GB GDDR3, PCI Express x16

```

# validar las opciones del sub-menu de los Productos por reseña en el servicio
while validar_submenu_resena != 1:

    # Si la opción seleccionada en el sub-menu es 1
    if subMenu_produc_resena == "1":

        # Se iniciara la iteración en la lista lifestore_sales:
        for linea_resena in lifestore_sales:

            # Si se encuentra un producto se pondra en falso
            producto_localizado = False

            # la variable articulo_cal se iguala al valor que valla teniendo la
            # variable linea_resena en el indice[2]
            articulo_cal = linea_resena[2]

            # Se hará una nueva iteración
            for linea_resenas_tot in lista_resenas_tot:

                # Si articulo_cal es igual a lista_resenas_tot en la posición cero entonces:
                if articulo_cal == linea_resenas_tot[0]:
                    # La variable linea_resenas_tot se autoincrementará
                    linea_resenas_tot[1] = linea_resenas_tot[1] + 1

            # Si producto_localizado ==false entonces. . .
            if producto_localizado == False:
                # Se añadirá el nuevo producto a lista_resenas_tot
                lista_resenas_tot.append([articulo_cal, 1])

```

```

# SE hará una nueva iteración
for linea_resenas_tot in lista_resenas_tot:

    # Si articulo_cal es igual a linea_resenas_tot en la posición cero entonces...
    articulo_cal = linea_resenas_tot[0]

    # cantidad_valor se igualara a linea_resenas_tot en la posición cero
    cantidad_valor = linea_resenas_tot[1]

    # Si indice_resenas_tot es igual a cero entonces...
    if indice_resenas_tot == 0:

        # Se insertara un nuevo dato a lista_resenas_ord
        lista_resenas_ord.insert(1, [articulo_cal, cantidad_valor])

    indice_resenas_tot += 1

```



```

else:

    # localizado se volvera falso
    localizado = False

    # Se reiniciara valor_indice
    valor_indice = 0

    # SE hará una nueva iteración
    for linea_resena_ord in lista_resenas_ord:

        # Si cantidad_valor es mayo o igual a lin en la posición cero entonces...
        if cantidad_valor >= linea_resena_ord[1]:
            # Se insertará un nuevo valor a lista_resenas_ord
            lista_resenas_ord.insert(valor_indice, [articulo_cal, cantidad_valor])

            # localizado se vuelve verdad
            localizado = True

            # Se detiene el ciclo
            break

        valor_indice = valor_indice + 1

    if localizado == False:
        lista_resenas_ord.append([articulo_cal, cantidad_valor])

indice_resenas_tot = indice_resenas_tot + 1

```

```

# Se hará una iteración en lista_resenas_ord
for resenas_ord in lista_resenas_ord:

    # mensaje para el usuario
    print(contador_puntaje_ord, "EL Producto con ID: ", resenas_ord[1])
    print("Obtuvo la calificación de:", resenas_ord[0], " \n")

    # Si contador_puntaje_ord es mayor a 20 entonces...
    if contador_puntaje_ord >= 20:
        # Se detendra el conteo
        break

    # contador_puntaje_ord se autoincrementará
    contador_puntaje_ord = contador_puntaje_ord + 1

# La bandera se pondra en verdadero
validar_submenu_resena = 1

```


Ilustración 20 Resultado del análisis en pantalla

Listado de productos con mejores reseñas:

Listado de productos con peores reseñas:

La opción seleccionada es: 1

1 EL Producto con ID: 198

Obtuvo la calificación de: 5

2 EL Producto con ID: 197

Obtuvo la calificación de: 5

3 EL Producto con ID: 196

Obtuvo la calificación de: 5

4 EL Producto con ID: 195

Obtuvo la calificación de: 5

5 EL Producto con ID: 194

Obtuvo la calificación de: 5

6 EL Producto con ID: 193

Obtuvo la calificación de: 5

7 EL Producto con ID: 192

Obtuvo la calificación de: 5

8 EL Producto con ID: 191

Obtuvo la calificación de: 5

Solución al problema

- Productos mayores ventas:

Existe un abismo muy notorio en cuanto a las ventas de los productos, ya que por ejemplo entre el primero y el tercero la diferencia es de 30 productos. También se percibe que existen muchos productos de una sola venta. Para aumentar las ventas de la empresa se sugiere hacer un paquete de productos en donde se ofrezca al público un producto de mayor rotación y otro de menor.

- Pocas ventas mucha búsqueda

Se percibió que existen productos que tienen un número alto de búsqueda, pero en contraste la compra del mismo es muy baja, es por ello que se sugiere ofrecer algún tipo de promoción o descuento que incite al usuario a querer comprarlo.

- Pocas ventas poca búsqueda

Otro aspecto importante que se encontró es que existen productos que tienen muy pocas ventas y búsquedas como, por ejemplo:

1. Tarjeta de video MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0
2. Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel
3. Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0

- Tarjeta de video

Son los principales artículos con menos ventas, esto puede deberse al precio de las mismas se recomienda sacar este producto del mercado.

- Audífonos Gamer

Producto con poca venta, poca búsqueda y precio elevado, se sugiere sacar este producto del mercado.

En general, se recomienda realizar descuentos en los productos que tienen gran búsqueda, pero poca compra, para los productos que son poco buscados y vendidos es recomendable mejorar el contenido del mismo, es decir analizar si los productos son fáciles de encontrar en la plataforma de venta y si están en las categorías adecuadas. Otra estrategia para disminuir el stock es vender productos que en gran cantidad y retirar del mercado los productos con pocas ventas.

Conclusiones

En la descripción del proyecto solo se mostró hasta el apartado de 1 de la segunda consigna. Para ser sincera lo faltante no lo realice por que no supe cómo, se me dificultó bastante. En un primer momento cuando leí el proyecto se me hizo fácil, pero conforme lo fui desarrollando me di cuenta de que, si tenía un grado de complejidad y para mí la programación en Python es algo nuevo, creo yo que por eso tuve muchas dificultades.

Aun cuando no lo concluí, decidí enviarlo así para que observaran hasta donde llegue y también para comentarles que lo poco o mucho que realice la verdad me esforcé demasiado, estoy un poco decepcionada de mi desempeño, pero sé que solo es un eslabón que me ha traído mucha experiencia.