



ANALISIS DE DATOS

Josue Lobato Vázquez
mat.josuelobato@hotmail.com

Septiembre 2020

Contents

1	Introducción	3
2	Código	4
2.1	Menú principal	4
2.2	Administrador	4
2.2.1	Crear Usuario	5
2.2.2	Eliminar Usuario	6
2.2.3	Mostrar usuarios existentes y su contraseña	7
2.3	Usuario	8
2.3.1	50 Productos con mayores ventas	9
2.3.2	50 productos con menores ventas	12
2.3.3	50 productos con más búsquedas	13
2.3.4	50 productos con menos búsquedas	15
2.3.5	20 productos con las mejores reseñas	16
2.3.6	20 productos con las peores reseñas	18
2.3.7	Ingresos y ventas por Mes/Año Parte 1	19
2.3.8	Porcentaje ganancias por categoria anual	32
2.3.9	else	33
3	Solución al problema	34
3.1	50 productos con mayores ventas	34
3.2	50 productos menos vendidos	35
3.3	50 productos más buscados	36
3.4	20 productos con peores reseñas	37
3.5	Porcentaje de ganancias por categoria	37
3.6	Ganancias Total	37
4	Conclusión	38

1 Introducción

LifeStore es una tienda virtual que maneja una amplia gama de artículos, recientemente, le gerencia de ventas, se percato 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.

2 Código

Administradores: josue, javier **Contraseñas:** 2468, 4321 respectivamente.

2.1 Menú principal

En el principio tenemos la declaración de un ciclo while que contendrá todo el código consecuentemente, solo se podrá salir del ciclo cuando **seguir_main = False** lo cual suceda si la persona responde que **Si** a la siguiente pregunta: **Desea salir del programa: (Si/No)**
Seguido del while se encuentran dos opciones de ingreso

```
seguir_main = True
while seguir_main == True:
    print("""Opciones para ingresar:
        1 Administrador
        2 Usuario""")
    opcion = input("Escriba el número de la opción seleccionada: ")
```

2.2 Administrador

Al seleccionar la opción de **Administrador** seguirá la opción de ingreso de cuenta y contraseña.

```
if opcion == '1':
    while seguir == True: #Controla salir o seguir en administrador
        admin = input("Ingrese usuario de administrador: ")
        if admin in administradores:
            contraseña = input("Ingrese nip de administrador:")
            indice = administradores.index(admin)
            if contraseña == nips_admins[indice]:
                print("""Opciones:
                    1 Crear usuario
                    2 Eliminar usuario
                    3 Mostar usuarios existente y su contraseña
                    """)
                opcion_administrador = input("Escriba el número de la opción seleccionada: ")
                ...
                ...
            else:
                print("Contraseña incorrecta")
                seguir = False
        else:
            print("Administrador no encontrado")
            seguir = False
```

El while controla salir o continuar en la opción de administrador dependiendo del valor de **seguir**, que en principio tiene el valor **True**. Si el ingreso de la cuenta y su contraseña es exitoso se ingresa al menú de lo que puede hacer el administrador, pero si se equivoca al ingresar sus datos automáticamente **seguir = False** y saldrá del while para regresar al menú principal.

2.2.1 Crear Usuario

Al escoger la primera opción podemos crear un nuevo usuario y su contraseña para entrar al programa, esto no vale para crear nuevos administradores sino que solo podran entrar en la opcion 2 del menú principal. Se encuentra dentro de un while que nos permite crear los usuarios que se desean sin necesidad de regresar al menú de administrador, y al terminar de crear un usuario siempre se nos mostrara la opción de parar al decir que no a: **Desea crear otro usuario: (Si/No)**, consecuentemente se nos preguntara **Desea seguir como administrador: (Si/No)** para regresar al menú principal o seguir como administrador.

Finalmente se nos da la opción de salir del programa.

```
if opcion_administrador == '1':
    seguir_1 = True
    while seguir_1 == True:
        nuevo_usuario = input("Ingrese nombre de nuevo usuario: ")
        print("El nombre ingresado es: "+nuevo_usuario)
        nuevo_nip = input("Ingrese nip: ")
        print("El nip ingresado es:",int(nuevo_nip))
        usuarios.append(nuevo_usuario)
        nips.append(nuevo_nip)
        opcion_administrador_2 = input("Desea crear otro usuario: (Si/No)\n")
        if opcion_administrador_2 == 'Si' or opcion_administrador_2 == 'si' :
            seguir_1 = True
        else:
            seguir_1 = False
            opcion_salir = input("Desea seguir como administrador: (Si/No) \n")
            if opcion_salir == 'Si' or opcion_salir == 'si' :
                seguir = True
            else:
                seguir = False
                opcion_salir_main = input("Desea salir del programa: (Si/No)")
                if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
                    seguir_main = True
                else:
                    seguir_main = False
```

2.2.2 Eliminar Usuario

Permite eliminar a cualquier usuario y su contraseña, se escribe el usuario y el programa busca que se encuentre dentro de la lista, al encontrarlo lo elimina junto con su contraseña que se encuentra en otra lista de nips pero con el mismo índice, en caso de no existir el usuario el programa mandara un mensaje de ello y se lo volvera a pedir.

Tienes las mismas opciones de continuar, regresar al menú de administrador, menú principal, o salir del programa.

```
if opcion_administrador == '2':
    seguir_1 = True
    while seguir_1 == True:
        for usuario in usuarios:
            print(usuario)
        usuario_eliminar = input("Escriba el usuario que desea eliminar: ")
        if usuario_eliminar in usuarios:
            indice = usuarios.index(usuario_eliminar)
            usuarios.pop(indice)
            nips.pop(indice)
            print("El usuario se a eliminado:")
            print("Aquí estan los usuarios vigentes: \n")
            for usuario in usuarios:
                print(usuario)
            opcion_administrador_2 = input("Desea eliminar otro usuario: (Si/No)")
            if opcion_administrador_2 == 'Si' or opcion_administrador_2 == 'si':
                seguir_1 = True
            else:
                seguir_1 = False
                opcion_salir = input("Desea seguir como administrador: (Si/No)")
                if opcion_salir == 'Si' or opcion_salir == 'si' :
                    seguir = True
                else:
                    seguir = False
                    opcion_salir_main = input("Desea salir del programa: (Si/No)")
                    if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
                        seguir_main = False
                    else:
                        seguir_main = True
        else:
            print("El usuario mencionado no existe")
            seguir_1 = True
```

2.2.3 Mostrar usuarios existentes y su contraseña

Muestra en pantalla los usuarios existentes y sus respectivas contraseñas. Se encuentran también las opciones de salida.

```
if opcion_administrador == '3':
    for usuario in usuarios:
        indice = usuarios.index(usuario)
        print(usuario,":",nips[indice])
    opcion_salir = input("Desea seguir como administrador: (Si/No)")
    if opcion_salir == 'Si' or opcion_salir == 'si' :
        seguir = True
    else:
        seguir = False
        opcion_salir_main = input("Desea salir del programa: (Si/No)") #Para salir del
        if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
            seguir_main = False
        else:
            seguir_main = True
```

2.3 Usuario

Al elegir la opción de usuario entraremos a otro menú para obtener los datos que se han solicitado de manera organizada por secciones.

```
if opcion == '2':
    usuario = input("Ingrese su nombre de usuario: ")
    seguir_3 = True
    while seguir_3 == True:
        if usuario in usuarios:
            print("El usuario se encuentra registrado")
            indice = usuarios.index(usuario)
            nip = input("Ingrese su nip de 4 dígitos: \nejemplo: 1234\n")
            contraseña = 0
            while contraseña < 3 and seguir_3 == True:
                if nip == nips[indice]:
                    print("Contraseña correcta\n")
                    continuar_programa = 'Si'
                    while continuar_programa == 'Si' or continuar_programa == 'si':
                        menu = '''
                                Bienvenido
                                Escriba el número de la opción que desea consultar:
                                1  50 productos con mayores ventas
                                2  50 productos con menores ventas
                                3  50 productos más buscados
                                4  50 productos menos buscados
                                5  20 productos con las mejores reseñas
                                6  20 productos con las peores reseñas
                                7  Ingresos y ventas por Mes/Año
                                8  Porcentaje ganancias por categoria anual
                                '''
                        print(menu)
                        seleccion = input("Ingrese su elección: ")
                        ...
                        ...
                else:
                    print("Contraseña incorrecta
                    Tiene 3 oportunidades más o regresara al menu principal")
                    contraseña += 1
                    nip = input("Ingrese su nip de 4 dígitos: \nejemplo: 1234\n")
            else:
                seguir_3 = False
        else:
            print("Usuario no identificado")
            usuario = input("Ingrese su nombre de usuario: ")
```


Pero antes se pedira su usuario si este no se ingresa correctamente se lo volvera a pedir, después se pedira su contraseña, tendra 3 oportunidades o regresara al menú principal, ya que **while contraseña < 3 and seguir_3 == True:** tiene un **else:** indexado que cambia **seguir_3 = False** después de las 3 oportunidades, y saldra del **while seguir_3 == True:** que es el principal en la opción de usuario para continuar o salir de allí. Todas las opciones se escogeran con un **if-else** anidado.

2.3.1 50 Productos con mayores ventas

La primera parte del código se usa tanto para la opción 1 y la 2 que son respectivamente 50 productos con mayores ventas y 50 productos con menores ventas.

Este código cuenta las ventas de cada producto, recorriendo la lista **lifestore_sales** dentro de un **while** que termina cuando se haya recorrido toda la lista o se haya llegado al limite del **id** de productos registrados que son **96**. Los datos se ponen en una lista llamada **cantidad_ventas_producto** que es una lista de listas, donde las sublistas son de la siguiente manera: **[id,ventas]**.

Después se utiliza el método burbuja para ordenar la lista de listas de acuerdo al número de ventas de mayor a menor, esto utilizando una variable auxiliar.

```
if seleccion == '1' or seleccion == '2':
    cantidad_ventas_producto = []
    id = 1          #Va desde 1 a 96 que es el total de productos
    contador = 0    #Para saber cuantas ventas tuvo del mismo producto
    indice = 0      #Para movernos el la lista principal
    while (id <= 96) and (indice < len(lifestore_sales)) :
        sale = lifestore_sales[indice]
        if sale[1] == id :
            contador += 1
            indice += 1
        else :
            cantidad_ventas_producto.append([id,])
            cantidad_ventas_producto[id-1].append(contador)
            contador = 0
            id += 1
    cantidad_ventas_producto.append([id,contador])
    #Adjuntamos la ultima cantidad del ultimo id en este caso 94

    #Método burbuja
    len(cantidad_ventas_producto)-1 iteraciones para terminar
    n = len(cantidad_ventas_producto)
    for i in range(1,n):
        for j in range(0,n-i):
            if(cantidad_ventas_producto[j+1][1] > cantidad_ventas_producto[j][1]):
                aux=cantidad_ventas_producto[j];
                cantidad_ventas_producto[j]=cantidad_ventas_producto[j+1];
                cantidad_ventas_producto[j+1]=aux;
```

El código para imprimir los **50 productos con mayores ventas**, **stock_sobran** es para ver lo que sobra de los productos con mayores ventas, para saber si debe adquirir más. Se imprime una tabla con los 50 productos enumerados, sus ventas, stock restante y el nombre del producto.

Los **if-elif** anidados son para que las líneas de la tabla queden alineadas y sea estético.

```
if seleccion == '1':
    '''50 productos más vendidos'''
    stock_sobran = []
    for cantidad in cantidad_ventas_producto:
        for producto in lifestore_products:
            if producto[0] == cantidad[0]:
                stock_sobran.append(producto[4])
                break

print("50 productos con mayores ventas: ")
print("N |Ventas|Stock|Producto")
print("-----")
for i in range(50):
    id = cantidad_ventas_producto[i][0]
    ventas = cantidad_ventas_producto[i][1] #Ventas del producto
    producto = lifestore_products[id-1][1] #Para obtener nombre del producto
    #Para que la tabla se vea totalmente alineada
    if i+1 <= 9:
        if stock_sobran[i] <= 9:
            if stock_sobran[i] < ventas:
                print(i+1," |",ventas," |",'%d |'% stock_sobran[i],
                    "\033[1;31m"+producto+"\033[0;30m")
            else:
                print(i+1," |",ventas," |",'%d |'% stock_sobran[i],producto)
        elif stock_sobran[i] > 9 and stock_sobran[i] <= 99:
            if stock_sobran[i] < ventas:
                print(i+1," |",ventas," |",'%d |'% stock_sobran[i],
                    "\033[1;31m"+producto+"\033[0;30m")
            else:
                print(i+1," |",ventas," |",'%d |'% stock_sobran[i],producto)
        else:
            if stock_sobran[i] < ventas:
                print(i+1," |",ventas," |",'%d |'% stock_sobran[i],
                    "\033[1;31m"+producto+"\033[0;30m")
            else:
                print(i+1," |",ventas," |",'%d |'% stock_sobran[i],producto)
```

```

else:
    if stock_sobrante[i] <= 9:
        if stock_sobrante[i] < ventas:
            print(i+1,"|",ventas,"    |",'%d    |'% stock_sobrante[i],
                  "\033[1;31m"+producto+"\033[0;30m")
        else:
            print(i+1,"|",ventas,"    |",'%d    |'% stock_sobrante[i],producto)
    elif stock_sobrante[i] > 9 and stock_sobrante[i] <= 99:
        if stock_sobrante[i] < ventas:
            print(i+1,"|",ventas,"    |",'%d    |'% stock_sobrante[i],
                  "\033[1;31m"+producto+"\033[0;30m")
        else:
            print(i+1,"|",ventas,"    |",'%d    |'% stock_sobrante[i],producto)
    else:
        if stock_sobrante[i] < ventas:
            print(i+1,"|",ventas,"    |",'%d    |'% stock_sobrante[i],
                  "\033[1;31m"+producto+"\033[0;30m")
        else:
            print(i+1,"|",ventas,"    |",'%d    |'% stock_sobrante[i],producto)

print("-----")
print("\033[1;31m"+"Articulos en rojo se recomienda rellenar stock,
ya que el producto se ha agotado o se tiene una cantidad inferior a
la demanda mostrada por las ventas del producto"+" \033[0;30m")
print("-----")

opcion_usuario = input("Desea volver al menu: (Si/No)\n ")
if opcion_usuario == 'Si' or opcion_usuario == 'si':
    continuar_programa = 'Si'
else:
    continuar_programa = 'No'
opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
    seguir_3 = False
    seguir_main = False
else:
    seguir_3 = False
    seguir_main = True

```

Lo último en el código son las opciones de permanencia y salida.

2.3.2 50 productos con menores ventas

Se manda a imprimir los productos enumerados, sus ventas y el nombre del producto.

El **for** es para ir imprimiendo estos datos recorriendo la lista **cantidad_ventas_producto** pero empezando desde el último elemento que es el menos vendido y así en retroceso, haciendo slicing con números negativos.

```
print("50 productos con menores ventas: ")
print("N |Ventas|Producto")
print("-----")
for i in range(1,51):
    id = cantidad_ventas_producto[-i][0]
    ventas = cantidad_ventas_producto[-i][1] #Ventas del producto
    producto = lifestore_products[id-1][1] #Para obtener nombre del producto
    #Para que la tabla se vea totalmente alineada
    if i <= 9:
        print(i," |",ventas," |",producto)
    else:
        print(i,"|",ventas," |",producto)
print("-----")
opcion_usuario = input("Desea volver al menu: (Si/No)\n ")
if opcion_usuario == 'Si' or opcion_usuario == 'si':
    continuar_programa = 'Si'
else:
    continuar_programa = 'No'
opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
    seguir_3 = False
    seguir_main = False
else:
    seguir_3 = False
    seguir_main = True
```

2.3.3 50 productos con más búsquedas

La primera parte del código se aplica para la opción 3 y 4, y es lo mismo que en el caso 1 y 2, se crea la lista de mayor a menor respecto a búsquedas, y se ordena por método burbuja

```
elif seleccion == '3' or seleccion == '4':
    '''50 Productos con mayores búsquedas o menores búsquedas'''
    cantidad_busqueda_producto = []
    id = 1 #Va desde 1 a 96 que es el total de productos
    contador = 0 #Para saber cuantas búsquedas tuvo del mismo producto
    indice = 0 #Para movernos en la lista lifestore_searches

    while (id <= 96) and (indice < len(lifestore_searches)) :
        busqueda = lifestore_searches[indice]

        if busqueda[1] == id :
            contador += 1
            indice += 1 #Pasamos a la siguiente lista en lifestore_searches
        else :
            cantidad_busqueda_producto.append([id,])
            cantidad_busqueda_producto[id-1].append(contador)
            contador = 0 #volvemos a empezar
            id += 1
    cantidad_busqueda_producto.append([id,contador])

    #Método burbuja
    n = len(cantidad_busqueda_producto)
    for i in range(1,n):
        for j in range(0,n-i):
            if(cantidad_busqueda_producto[j+1][1] > cantidad_busqueda_producto[j][1]):
                aux=cantidad_busqueda_producto[j];
                cantidad_busqueda_producto[j]=cantidad_busqueda_producto[j+1];
                cantidad_busqueda_producto[j+1]=aux;
```

Veamos para imprimir los **50 productos más buscados**

Se imprime una tabla enumerada donde el 1 es el producto más buscado, y se pone su nombre, los **if-else** anidados son para acomodar la tabla y sea vea mejor.

Al final tenemos las mismas opciones de seguir o salir.

```
if seleccion == '3':
    print("50 productos más buscados: ")
    print("N   |busquedas|Producto")
    print("-----")
    for i in range(50):
        id = cantidad_busqueda_producto[i][0]
        busquedas = cantidad_busqueda_producto[i][1]
        producto = lifestore_products[id-1][1]
        #Para que la tabla se vea totalmente alineada
        if i+1 <= 9 and i+1 > 2:
            print(i+1,"   |",busquedas,"       |",producto)
        elif busquedas <= 99 and busquedas >= 10:
            print(i+1," |",busquedas,"       |",producto)
        elif busquedas <= 9:
            print(i+1," |",busquedas,"       |",producto)
        else :
            print(i+1,"   |",busquedas,"       |",producto)
    print("-----")
    opcion_usuario = input("Desea volver al menu: (Si/No)\n ")
    if opcion_usuario == 'Si' or opcion_usuario == 'si':
        continuar_programa = 'Si'
    else:
        continuar_programa = 'No'
    opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
    if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
        seguir_3 = False
        seguir_main = False
    else:
        seguir_3 = False
        seguir_main = True
```

2.3.4 50 productos con menos busquedas

El **for** va de 1 a 51 para llegue al 50, que serían justamente los elementos que queremos, **i** se va moviendo en la lista **[1,2,...,50]** pero de modo inverso al colocar **[-i]** porque al final de la lista estan los productos menos buscados, ya que **cantidad_busqueda_producto** esta ordenada de mayor a menor en busquedas.

```
else:
    '''50 productos menos buscados'''
    print("50 productos menos buscados: ")
    print("N |busquedas|Producto")
    print("-----")
    for i in range(1,51):
        id = cantidad_busqueda_producto[-i][0]
        busquedas = cantidad_busqueda_producto[-i][1]
        producto = lifestore_products[id-1][1]
        #Para que la tabla se vea totalmente alineada
        if i <= 9:
            print(i," |",busquedas," |",producto)
        else:
            print(i,"|",busquedas," |",producto)
    print("-----")

opcion_usuario = input("Desea volver al menu: (Si/No)\n ")
if opcion_usuario == 'Si' or opcion_usuario == 'si':
    continuar_programa = 'Si'
else:
    continuar_programa = 'No'
    opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
    if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
        seguir_3 = False
        seguir_main = False
    else:
        seguir_3 = False
        seguir_main = True
```

2.3.5 20 productos con las mejores reseñas

Aquí sucede lo mismo que en las opciones anteriores, con la diferencia de que sólo tomara en cuenta los productos que tengan reseña, y los que no no serán tomados en cuenta.

```
elif seleccion == '5' or seleccion == '6':
    '''20 Productos con las mejores reseñas o peores reseñas'''
    reseñas_producto = []
    id = 1
    contador = 0
    puntaje = 0
    indice = 0

    while (id <= 96) and (indice < len(lifestore_sales)) :
        sale = lifestore_sales[indice]
        if sale[1] == id :
            puntaje += sale[2]
            contador += 1
            indice += 1
        else :
            if contador != 0:
#En la lista reseñas_producto habrá productos con reseñas y quitara los que no tienen
                reseñas_producto.append([id,])
                reseñas_producto[-1].append(puntaje/contador)
            puntaje = 0
            contador = 0
            id += 1

    reseñas_producto.append([id,puntaje/contador])

#Método burbuja
n = len(reseñas_producto)
for i in range(1,n):
    for j in range(0,n-i):
        if(reseñas_producto[j+1][1] > reseñas_producto[j][1]):
            aux=reseñas_producto[j];
            reseñas_producto[j]=reseñas_producto[j+1];
            reseñas_producto[j+1]=aux;
```


Aquí va la lista de las mejores 20 reseñas pero en promedio, obtuvimos las reseñas de cada producto y se obtuvo el promedio de ellas, de esa manera se acomodaron

```
if seleccion == '5':
    print("20 productos con las mejores reseñas: ")
    print("N |Reseña |Producto")
    print(" |Promedio| ")
    print("-----")
    for i in range(20):
        id = reseñas_producto[i][0]
        reseña_promedio = reseñas_producto[i][1] #Ventas del producto
        producto = lifestore_products[id-1][1] #Para obtener nombre del producto
        #Para que la tabla se vea totalmente alineada
        if i+1 <= 9:
            print(i+1," |%.2f" % reseña_promedio," |",producto)
        else:
            print(i+1,"|%.2f" % reseña_promedio," |",producto)
    print("-----")

opcion_usuario = input("Desea volver al menu: (Si/No)\n ")
if opcion_usuario == 'Si' or opcion_usuario == 'si':
    continuar_programa = 'Si'
else:
    continuar_programa = 'No'
opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
    seguir_3 = False
    seguir_main = False
else:
    seguir_3 = False
    seguir_main = True
```

2.3.6 20 productos con las peores reseñas

Es lo mismo que lo anterior pero recorriendo la lista de manera inversa para obtener los **20 productos con las peores reseñas**

```
else:
    '''20 Productos con las peores reseñas
    Solo se toman en cuenta productos que recibieron reseñas
    '''

    print("20 productos con las peores reseñas: ")
    print("N |Reseña |Producto")
    print(" |Promedio| ")
    print("-----")
    for i in range(1,21):
        id = reseñas_producto[-i][0]
        reseña_promedio = reseñas_producto[-i][1]
        producto = lifestore_products[id-1][1]
        #Para que la tabla se vea totalmente alineada
        if i <= 9:
            print(i," |%.2f" % reseña_promedio," |",producto)
        else:
            print(i,"|%.2f" % reseña_promedio," |",producto)
    print("-----")

    opcion_usuario = input("Desea volver al menu: (Si/No)\n ")
    if opcion_usuario == 'Si' or opcion_usuario == 'si':
        continuar_programa = 'Si'
    else:
        continuar_programa = 'No'
        opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
        if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
            seguir_3 = False
            seguir_main = False
        else:
            seguir_3 = False
            seguir_main = True
```

2.3.7 Ingresos y ventas por Mes/Año Parte 1

Primero se separan los datos por mes, esta opción a diferencia de las otras tendrá su propio menú interno para ver los datos por Mes o Total, pero primero se crean las listas de cada mes en donde se encontrara **[Ganacias,cantidad vendidos, devoluciones]**.

Esto se hace con un **if-else** anidado, se ira recorriendo la lista **lifestore_sales** y en cada una el mes se encuentra en el lugar: **[3][3:5]**, así se separan los datos.

Como en ocasiones anteriores el while es para mantenerse dentro de la opción 7 hasta que se indique lo contrario.

```
elif seleccion == '7':
    menu_7 = True
    while menu_7 == True:
        '''Total de ingresos y ventas promedio mensuales o anual'''
        id = 0
        indice = 0
        Enero = [[0,0,0]]
        Febrero = [[0,0,0]]
        Marzo = [[0,0,0]]
        Abril = [[0,0,0]]
        Mayo = [[0,0,0]]
        Junio = [[0,0,0]]
        Julio = [[0,0,0]]
        Agosto = [[0,0,0]]
        Septiembre = [[0,0,0]]
        Octubre = [[0,0,0]]
        Noviembre = [[0,0,0]]
        Diciembre = [[0,0,0]]
```

Se recorrera el código a la izquierda por estetica del archivo, pero esto va dentro del while principal del menú 7 que se muestra arriba

```
while (id <= 96) and (indice < len(lifestore_sales)) :
    sale = lifestore_sales[indice]
    #Verificamos el mes de la venta, para juntar los datos
    if sale[3][3:5] == '01': #sale[3][3:5] va a dar el mes en string '04'
        #El 0 indica que no hubo devolución y podemos contar la venta y las ganancias
        if sale[4] == 0:
            Enero[0][1] += 1 #Se cuenta la venta
            id = sale[1]
            Enero[0][0] += lifestore_products[id][2] #Se suma la ganancia
            Enero.append([lifestore_products[id][1],lifestore_products[id][2]])
            indice += 1
        else: #Hubo devolución así que la contamos
            Enero[0][2] += 1 #Aumenta devolucion
            Enero[0][1] += 1 #suma la venta pero no las ganancias
            indice += 1
    elif sale[3][3:5] == '02':
```

```
if sale[4] == 0:
    Febrero[0][1] += 1
    id = sale[1]
    Febrero[0][0] += lifestore_products[id][2]
    Febrero.append([lifestore_products[id][1],lifestore_products[id][2]])
    indice += 1
else:
    Febrero[0][2] += 1
    Febrero[0][1] += 1 #suma le venta pero no las ganancias
    indice += 1
elif sale[3][3:5] == '03':
    if sale[4] == 0:
        Marzo[0][1] += 1
        id = sale[1]
        Marzo[0][0] += lifestore_products[id][2]
        Marzo.append([lifestore_products[id][1],lifestore_products[id][2]])
        indice += 1
    else:
        Marzo[0][2] += 1
        Marzo[0][1] += 1 #suma le venta pero no las ganancias
        indice += 1
elif sale[3][3:5] == '04':
    if sale[4] == 0:
        Abril[0][1] += 1
        id = sale[1]
        Abril[0][0] += lifestore_products[id][2]
        Abril.append([lifestore_products[id][1],lifestore_products[id][2]])
        indice += 1
    else:
        Abril[0][2] += 1
        Abril[0][1] += 1 #suma le venta pero no las ganancias
        indice += 1
elif sale[3][3:5] == '05':
    if sale[4] == 0:
        Mayo[0][1] += 1
        id = sale[1]
        Mayo[0][0] += lifestore_products[id][2]
        Mayo.append([lifestore_products[id][1],lifestore_products[id][2]])
        indice += 1
    else:
        Mayo[0][2] += 1
        Mayo[0][1] += 1 #suma le venta pero no las ganancias
        indice += 1
elif sale[3][3:5] == '06':
    if sale[4] == 0:
        Junio[0][1] += 1
        id = sale[1]
```

```

        Junio[0][0] += lifestore_products[id][2]
        Junio.append([lifestore_products[id][1],lifestore_products[id][2]])
        indice += 1
    else:
        Junio[0][2] += 1
        Junio[0][1] += 1 #suma le venta pero no las ganancias
        indice += 1
elif sale[3][3:5] == '07':
    if sale[4] == 0:
        Julio[0][1] += 1
        id = sale[1]
        Julio[0][0] += lifestore_products[id][2]
        Julio.append([lifestore_products[id][1],lifestore_products[id][2]])
        indice += 1
    else:
        Julio[0][2] += 1
        Julio[0][1] += 1 #suma le venta pero no las ganancias
        indice += 1
elif sale[3][3:5] == '08':
    if sale[4] == 0:
        Agosto[0][1] += 1
        id = sale[1]
        Agosto[0][0] += lifestore_products[id][2]
        Agosto.append([lifestore_products[id][1],lifestore_products[id][2]])
        indice += 1
    else:
        Agosto[0][2] += 1
        Agosto[0][1] += 1 #suma le venta pero no las ganancias
        indice += 1
elif sale[3][3:5] == '09':
    if sale[4] == 0:
        Septiembre[0][1] += 1
        id = sale[1]
        Septiembre[0][0] += lifestore_products[id][2]
        Septiembre.append([lifestore_products[id][1],lifestore_products[id][2]])
        indice += 1
    else:
        Septiembre[0][2] += 1
        Septiembre[0][1] += 1 #suma le venta pero no las ganancias
        indice += 1
elif sale[3][3:5] == '10':
    if sale[4] == 0:
        Octubre[0][1] += 1
        id = sale[1]
        Octubre[0][0] += lifestore_products[id][2]
        Octubre.append([lifestore_products[id][1],lifestore_products[id][2]])
        indice += 1

```

```
else:
    Octubre[0][2] += 1
    Octubre[0][1] += 1 #suma le venta pero no las ganancias
    indice += 1
elif sale[3][3:5] == '11':
    if sale[4] == 0:
        Noviembre[0][1] += 1
        id = sale[1]
        Noviembre[0][0] += lifestore_products[id][2]
        Noviembre.append([lifestore_products[id][1],lifestore_products[id][2]])
        indice += 1
    else:
        Noviembre[0][2] += 1
        Noviembre[0][1] += 1 #suma le venta pero no las ganancias
        indice += 1
elif sale[3][3:5] == '12':
    if sale[4] == 0:
        Diciembre[0][1] += 1
        id = sale[1]
        Diciembre[0][0] += lifestore_products[id][2]
        Diciembre.append([lifestore_products[id][1],lifestore_products[id][2]])
        indice += 1
    else:
        Diciembre[0][2] += 1
        Diciembre[0][1] += 1 #suma le venta pero no las ganancias
        indice += 1
else:
    continue #Termina el While
```

Ahora entramos al menu de la opción 7, el código va ligado al de la pagina anterior.

Menú opción 7

```
#Ingresar el mes que se desea checar
intentos = 0
Meses = ['Enero','Febrero','Marzo','Abril','Mayo','Junio','Julio','Agosto','Septiembre','Octubre','Noviembre','Diciembre']
while intentos < 3:
    print('-----')
    print("Enero      Febrero      Marzo")
    print("Abril      Mayo      Junio")
    print("Julio      Agosto      Septiembre")
    print("Octubre     Noviembre     Diciembre")
    print("              Total")
    Mes = input("Ingrese la opción para ver resumen de ganancias y ventas por mes o Total para resumen del año: ")
    if (Mes in Meses) or (Mes == 'Total'):
        break
    else:
        print("\n
                                Incorrecto
        Intente nuevamente, ingrese el mes como se muestra en pantalla \n
        ")
        print("-----")
        intentos += 1
else: #Este else pertenece al while, sentencia while-else
    print("3 intentos fallidos, por defecto se mostrara el Total del Año")
```

Se muestra el menú en donde se tienen 3 oportunidades para escribir la opción correcta o automáticamente se ira a la opción de Total.

Veamos lo que sucede con cada una de esas opciones.
En cada mes calcula las ganancias descontando las devoluciones, la cantidad de articulos vendidos, y la cantidad de devoluciones.
Siempre se tendra la opción de salir en cada mes y en el Total

Menú opción 7

```
if Mes == Meses[0]:
    print('-----')
    print('Ganancias Enero descontando devoluciones: $ %d'%Enero[0][0])
    print('Cantidad de articulos vendidos: %d'%Enero[0][1])
    print('Cantidad de devoluciones: %d'%Enero[0][2])
    print('\n')
    print('Lista de productos vendidos: ')
    print('$Precio| Producto')
    Enero.pop(0)
    for datos_producto in Enero:
        print('$',datos_producto[1], '|',datos_producto[0])
    print('-----')
    opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
    if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
        menu_7 = False
        continuar_programa = 'No'
        seguir_3 = False
        seguir_main = False
    else:
        opcion_usuario = input("Desea volver al menu de usuario: (Si/No)\n ")
        if opcion_usuario == 'Si' or opcion_usuario == 'si':
            menu_7 = False
            continuar_programa = 'Si'
        else:
            menu_7 = True

elif Mes == Meses[1]:
    print('-----')
    print('Ganancias Febrero descontando devoluciones: $ %d'%Febrero[0][0])
    print('Cantidad de articulos vendidos: %d'%Febrero[0][1])
    print('Cantidad de devoluciones: %d'%Febrero[0][2])
    print('\n')
    print('Lista de productos vendidos: ')
    print('$Precio| Producto')
    Febrero.pop(0)
    for datos_producto in Febrero:
        print('$',datos_producto[1], '|',datos_producto[0])
    print('-----')
    opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
    if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
```



```

menu_7 = False
continuar_programa = 'No'
seguir_3 = False
seguir_main = False
else:
opcion_usuario = input("Desea volver al menu de usuario: (Si/No)\n ")
if opcion_usuario == 'Si' or opcion_usuario == 'si':
menu_7 = False
continuar_programa = 'Si'
else:
menu_7 = True
elif Mes == Meses[2]:
print('-----')
print('Ganancias Marzo descontando devoluciones: $ %d'%Marzo[0][0])
print('Cantidad de articulos vendidos: %d'%Marzo[0][1])
print('Cantidad de devoluciones: %d'%Marzo[0][2])
print('\n')
print('Lista de productos vendidos: ')
print('$Precio| Producto')
Marzo.pop(0)
for datos_producto in Marzo:
print('$',datos_producto[1],'|',datos_producto[0])
print('-----')
opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
menu_7 = False
continuar_programa = 'No'
seguir_3 = False
seguir_main = False
else:
opcion_usuario = input("Desea volver al menu de usuario: (Si/No)\n ")
if opcion_usuario == 'Si' or opcion_usuario == 'si':
menu_7 = False
continuar_programa = 'Si'
else:
menu_7 = True
elif Mes == Meses[3]:
print('-----')
print('Ganancias Abril descontando devoluciones: $ %d'%Abril[0][0])
print('Cantidad de articulos vendidos: %d'%Abril[0][1])
print('Cantidad de devoluciones: %d'%Abril[0][2])
print('\n')
print('Lista de productos vendidos: ')
print('$Precio| Producto')
Abril.pop(0)
for datos_producto in Abril:
print('$',datos_producto[1],'|',datos_producto[0])

```

```

print('-----')
opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
    menu_7 = False
    continuar_programa = 'No'
    seguir_3 = False
    seguir_main = False
else:
    opcion_usuario = input("Desea volver al menu de usuario: (Si/No)\n ")
    if opcion_usuario == 'Si' or opcion_usuario == 'si':
        menu_7 = False
        continuar_programa = 'Si'
    else:
        menu_7 = True
elif Mes == Meses[4]:
    print('-----')
    print('Ganancias Mayo descontando devoluciones: $ %d'%Mayo[0][0])
    print('Cantidad de articulos vendidos: %d'%Mayo[0][1])
    print('Cantidad de devoluciones: %d'%Mayo[0][2])
    print('\n')
    print('Lista de productos vendidos: ')
    print('$Precio| Producto')
    Mayo.pop(0)
    for datos_producto in Mayo:
        print('$',datos_producto[1],'|',datos_producto[0])
    print('-----')
    opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
    if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
        menu_7 = False
        continuar_programa = 'No'
        seguir_3 = False
        seguir_main = False
    else:
        opcion_usuario = input("Desea volver al menu de usuario: (Si/No)\n ")
        if opcion_usuario == 'Si' or opcion_usuario == 'si':
            menu_7 = False
            continuar_programa = 'Si'
        else:
            menu_7 = True
elif Mes == Meses[5]:
    print('-----')
    print('Ganancias Junio descontando devoluciones: $ %d'%Junio[0][0])
    print('Cantidad de articulos vendidos: %d'%Junio[0][1])
    print('Cantidad de devoluciones: %d'%Junio[0][2])
    print('\n')
    print('Lista de productos vendidos: ')
    print('$Precio| Producto')

```

```

Junio.pop(0)
for datos_producto in Junio:
print('$',datos_producto[1],'|',datos_producto[0])
print('-----')
opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
menu_7 = False
continuar_programa = 'No'
seguir_3 = False
seguir_main = False
else:
opcion_usuario = input("Desea volver al menu de usuario: (Si/No)\n ")
if opcion_usuario == 'Si' or opcion_usuario == 'si':
menu_7 = False
continuar_programa = 'Si'
else:
menu_7 = True
elif Mes == Meses[6]:
print('-----')
print('Ganancias Julio descontando devoluciones: $ %d'%Julio[0][0])
print('Cantidad de articulos vendidos: %d'%Julio[0][1])
print('Cantidad de devoluciones: %d'%Julio[0][2])
print('\n')
print('Lista de productos vendidos: ')
print('$Precio| Producto')
Julio.pop(0)
for datos_producto in Julio:
print('$',datos_producto[1],'|',datos_producto[0])
print('-----')
opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
menu_7 = False
continuar_programa = 'No'
seguir_3 = False
seguir_main = False
else:
opcion_usuario = input("Desea volver al menu de usuario: (Si/No)\n ")
if opcion_usuario == 'Si' or opcion_usuario == 'si':
menu_7 = False
continuar_programa = 'Si'
else:
menu_7 = True
elif Mes == Meses[7]:
print('-----')
print('Ganancias Agosto descontando devoluciones: $ %d'%Agosto[0][0])
print('Cantidad de articulos vendidos: %d'%Agosto[0][1])
print('Cantidad de devoluciones: %d'%Agosto[0][2])

```

```

print('\n')
print('Lista de productos vendidos: ')
print('$Precio| Producto')
Agosto.pop(0)
for datos_producto in Agosto:
    print('$',datos_producto[1],'|',datos_producto[0])
    print('-----')
opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
    menu_7 = False
    continuar_programa = 'No'
    seguir_3 = False
    seguir_main = False
else:
    opcion_usuario = input("Desea volver al menu de usuario: (Si/No)\n ")
    if opcion_usuario == 'Si' or opcion_usuario == 'si':
        menu_7 = False
        continuar_programa = 'Si'
    else:
        menu_7 = True
elif Mes == Meses[8]:
    print('-----')
    print('Ganancias Septiembre descontando devoluciones: $ %d'%Septiembre[0][0])
    print('Cantidad de articulos vendidos: ',0)
    print('Cantidad de devoluciones: ',0)
    print('\n')
    print('Lista de productos vendidos: ')
    print('$Precio| Producto')
    Septiembre.pop(0)
    for datos_producto in Septiembre:
        print('$',datos_producto[1],'|',datos_producto[0])
        print('-----')
    opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
    if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
        menu_7 = False
        continuar_programa = 'No'
        seguir_3 = False
        seguir_main = False
    else:
        opcion_usuario = input("Desea volver al menu de usuario: (Si/No)\n ")
        if opcion_usuario == 'Si' or opcion_usuario == 'si':
            menu_7 = False
            continuar_programa = 'Si'
        else:
            menu_7 = True
elif Mes == Meses[9]:
    print('-----')

```

```

print('Ganancias Octubre descontando devoluciones: $ %d'%Octubre[0][0])
print('Cantidad de articulos vendidos: %d'%Octubre[0][1])
print('Cantidad de devoluciones: %d'%Octubre[0][2])
print('\n')
print('Lista de productos vendidos: ')
print('$Precio| Producto')
Octubre.pop(0)
for datos_producto in Octubre:
print('$',datos_producto[1],'|',datos_producto[0])
print('-----')
opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
menu_7 = False
continuar_programa = 'No'
seguir_3 = False
seguir_main = False
else:
opcion_usuario = input("Desea volver al menu de usuario: (Si/No)\n ")
if opcion_usuario == 'Si' or opcion_usuario == 'si':
menu_7 = False
continuar_programa = 'Si'
else:
menu_7 = True
elif Mes == Meses[10]:
print('-----')
print('Ganancias Noviembre descontando devoluciones: $ %d'%Noviembre[0][0])
print('Cantidad de articulos vendidos: ',0)
print('Cantidad de devoluciones: ',0)
print('\n')
print('Lista de productos vendidos: ')
print('$Precio| Producto')
Noviembre.pop(0)
for datos_producto in Noviembre:
print('$',datos_producto[1],'|',datos_producto[0])
print('-----')
opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
menu_7 = False
continuar_programa = 'No'
seguir_3 = False
seguir_main = False
else:
opcion_usuario = input("Desea volver al menu de usuario: (Si/No)\n ")
if opcion_usuario == 'Si' or opcion_usuario == 'si':
menu_7 = False
continuar_programa = 'Si'
else:

```

```

menu_7 = True
elif Mes == Meses[11]:
    print('-----')
    print('Ganancias Diciembre descontando devoluciones: $ %d'%Diciembre[0][0])
    print('Cantidad de articulos vendidos: %d'%Diciembre[0][1])
    print('Cantidad de devoluciones: %d'%Diciembre[0][2])
    print('\n')
    print('Lista de productos vendidos: ')
    print('$Precio| Producto')
    Diciembre.pop(0)
    for datos_producto in Diciembre:
        print('$',datos_producto[1],',',datos_producto[0])
    print('-----')
    opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
    if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
        menu_7 = False
        continuar_programa = 'No'
        seguir_3 = False
        seguir_main = False
    else:
        opcion_usuario = input("Desea volver al menu de usuario: (Si/No)\n ")
        if opcion_usuario == 'Si' or opcion_usuario == 'si':
            menu_7 = False
            continuar_programa = 'Si'
        else:
            menu_7 = True

else: #Si no reconoce ningún mes esta es la ultima opción, Total

Total_ingresos = Enero[0][0] + Febrero[0][0] + Marzo[0][0] + Abril [0][0] +
Mayo[0][0] + Junio[0][0] + Julio[0][0] + Agosto[0][0] + Septiembre[0][0] +
Octubre[0][0] + Noviembre[0][0] + Diciembre[0][0]

Total_ventas = Enero[0][1] + Febrero[0][1] + Marzo[0][1] + Abril [0][1] +
Mayo[0][1] + Junio[0][1] + Julio[0][1] + Agosto[0][1] + Septiembre[0][1] +
Octubre[0][1] + Noviembre[0][1] + Diciembre[0][1]

Total_devoluciones = Enero[0][2] + Febrero[0][2] + Marzo[0][2] +
Abril [0][2] + Mayo[0][2] + Junio[0][2] + Julio[0][2] +
Agosto[0][2] + Septiembre[0][2] + Octubre[0][2] + Noviembre[0][2] + Diciembre[0][2]

```

```
#[Mes, Ventas, Ingresos]
Ventas_meses = [[Meses[0],len(Enero)-1,Enero[0][0]], [Meses[1],len(Febrero)-1,
Febrero[0][0]], [Meses[2],len(Marzo)-1,Marzo[0][0]], [Meses[3],len(Abril)-1,
Abril [0][0]], [Meses[4],len(Mayo)-1,Mayo[0][0]], [Meses[5],len(Junio)-1,
Junio[0][0]], [Meses[6],len(Julio)-1,Julio[0][0]], [Meses[7],len(Agosto)-1,
Agosto[0][0]], [Meses[8],len(Septiembre)-1,Septiembre[0][0]],
[Meses[9],len(Octubre)-1,Octubre[0][0]], [Meses[10],len(Noviembre)-1,
Noviembre[0][0]], [Meses[11],len(Diciembre)-1,Diciembre[0][0]]]
```

```
#Ordenar burbuja
n = len(Ventas_meses) #Ordenar los meses por mayor ganancia
for i in range(1,n):
for j in range(0,n-i):
if(Ventas_meses[j+1][2] > Ventas_meses[j][2]):
aux=Ventas_meses[j];
Ventas_meses[j]=Ventas_meses[j+1];
Ventas_meses[j+1]=aux;
```

```
print('-----')
print('Total de ingresos 2020: $%d'%Total_ingresos)
print('Total de ventas 2020: %d'%Total_ventas)
print('Total de devoluciones 2020: %d'%Total_devoluciones)
print('Mes      \t| Ingresos \t| Ventas ')
for ventas in Ventas_meses:
if len(ventas[0]) <= 7 and ventas[1] != 0:
print(ventas[0],'\t| $',ventas[2],'\t|',ventas[1])
else:
print(ventas[0],'\t| $',ventas[2],'\t\t|',ventas[1])
print('-----')
```

```
opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
menu_7 = False
continuar_programa = 'No'
seguir_3 = False
seguir_main = False
else:
opcion_usuario = input("Desea volver al menu de usuario: (Si/No)\n ")
if opcion_usuario == 'Si' or opcion_usuario == 'si':
menu_7 = False
continuar_programa = 'Si'
else:
menu_7 = True
```

2.3.8 Porcentaje ganancias por categoria anual

Recorremos cada producto, contamos cuantas ventas tuvo y las multiplicamos por su precio en lista, buscamos su categoria y ahí sumamos las ganancias, así vamos uno a uno, al final dividimos las ganancias por categoria entre las ganancias totales obtenidas.

```
elif seleccion == '8':
#Obtener porcentaje ganancias por categoria
    cantidad_ventas_producto = []
    id = 1
    contador = 0 #Para saber cuantas ventas tuvo del mismo producto
    indice = 0 #Para movernos el la lista principal
    while (id <= 96) and (indice < len(lifestore_sales)) :
        sale = lifestore_sales[indice]
        if sale[1] == id :
            contador += 1
            indice += 1
        else :
            cantidad_ventas_producto.append([id,])
            cantidad_ventas_producto[id-1].append(contador)
            contador = 0
            id += 1
    cantidad_ventas_producto.append([id,contador])

    cantidad_total = 0
    dinero_total = 0
    dinero = [0,0,0,0,0,0,0,0,0]
    categoria = ['procesadores','tarjetas de video','tarjetas madre',
'discos duros','memorias usb','pantallas','bocinas','audifonos']
    for cantidad in cantidad_ventas_producto:
#cantidad [id, cantidad vendida]
        for producto in lifestore_products:
            #[id_product, name, price, category, stock]
            cantidad_total += cantidad[1]
            if producto[0] == cantidad[0]:
                indice = categoria.index(producto[3])
                dinero[indice] += cantidad[1]*producto[2]
                dinero_total += cantidad[1]*producto[2]
                break

    print('-----')
    print('Porcentaje de ventas por categoria')
    print('procesadores|trajetas de video|tarjetas madre
|discos duros|memorias usb|pantallas|bocinas|audifonos')

    print('    %.2f  '% (dinero[0]/dinero_total*100),'|    %.2f  ',
% (dinero[1]/dinero_total*100),'|    %.2f  '

```



```
% (dinero[2]/dinero_total*100),'|    %.2f  ' % (dinero[3]/dinero_total*100),
'|    %.2f  '% (dinero[4]/dinero_total*100),'|    %.2f  '
% (dinero[5]/dinero_total*100),'|    %.2f'
% (dinero[6]/dinero_total*100),'|    %.2f'% (dinero[7]/dinero_total*100))

print('-----')

opcion_usuario = input("Desea volver al menu: (Si/No)\n ")
if opcion_usuario == 'Si' or opcion_usuario == 'si':
    continuar_programa = 'Si'
else:
    continuar_programa = 'No'
    opcion_salir_main = input("Desea salir del programa: (Si/No)\n")
    if opcion_salir_main == 'Si' or opcion_salir_main == 'si' :
        seguir_3 = False
        seguir_main = False
    else:
        seguir_3 = False
        seguir_main = True
```

2.3.9 else

Si ninguna elección es reconocida entonces se ejecutara el **else**

```
else:
    print('-----')
    print("Elección incorrecta !!!")
    print("Vuelva a intentar")
    print('-----')
    continuar_programa = 'Si'
```

3 Solución al problema

3.1 50 productos con mayores ventas

Podemos ver al pedir el resultado de los 50 productos con mayores ventas, que en rojo se muestran los artículos agotados o que tienen menos productos de los que se han llegado a vender y eso se debe tomar en cuenta.

N°	Ventas	Stock	Producto
1	50	300	SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm
2	42	987	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth
3	20	130	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)
4	18	0	Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
5	15	15	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm
6	14	10	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
7	13	182	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth
8	13	295	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador
9	11	8	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2
10	9	0	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0
11	9	50	SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2
12	7	114	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)
13	6	120	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
14	6	0	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AMD B450, 64GB DDR4 para AMD
15	5	5	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030, 2GB 64-bit GDDR5, PCI Express x16 3.0
16	4	8	Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake)
17	3	54	Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake)
18	3	2	Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Express 3.0
19	3	3	Kit SSD Kingston KC600, 1TB, SATA III, 2.5, 7mm
20	3	0	SSD Kingston UV500, 480GB, SATA III, mSATA
21	2	16	Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache
22	2	0	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC, 8GB 128-bit GDDR6, PCI Express 4.0
23	2	10	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming, 8GB 128-bit GDDR6, PCI Express 4.0
24	2	43	Tarjeta Madre ASUS ATX PRIME Z390-A, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel
25	2	13	SSD Western Digital WD Blue 3D NAND, 2TB, M.2
26	2	1	Logitech Bocinas para Computadora con Subwoofer G560, Bluetooth, Inalámbrico, 2.1, 120W RMS, USB, negro
27	2	39	Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul
28	1	13	MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0
29	1	1	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix, 4GB 128-bit GDDR5, PCI Express 3.0
30	1	1	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0
31	1	0	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC, 4GB 128-bit GDDR5, PCI Express x16 3.0

Solo son 42 productos los que muestran ventas, y sólo 27 los que muestran más de una venta, esto de 96 artículos con los que cuenta la tienda.

De los productos que tienen ventas, el porcentaje de sus ventas respecto a su stock es mínimo, si se debe reducir el stock de estos productos.

3.2 50 productos menos vendidos

Estos productos que se muestran a continuación deben de quitar de la tienda, no tuvieron ninguna venta y solo acumulan stock.

1	0	Ginja Audifonos con Micrófono GI18ADJ01BT-R0, Bluetooth, Alámbrico/Inalámbrico, 3.5mm, Rojo
2	0	Getttech Audifonos con Micrófono Sonority, Alámbrico, 1.2 Metros, 3.5mm, Negro/Rosa
3	0	Genius GHP-400S Audifonos, Alámbrico, 1.5 Metros, Rosa
4	0	Energy Sistem Audifonos con Micrófono Headphones 1, Bluetooth, Inalámbrico, Negro/Grafito
5	0	Audifonos Gamer Balam Rush Orphix RGB 7.1, Alámbrico, USB, Negro
6	0	Acer Audifonos Gamer Galea 300, Alámbrico, 3.5mm, Negro
7	0	ASUS Audifonos Gamer ROG Theta 7.1, Alámbrico, USB C, Negro
8	0	Ghia Bocina Portátil BX500, Bluetooth, Inalámbrico, 10W RMS, USB, Gris
9	0	Ghia Bocina Portátil BX400, Bluetooth, Inalámbrico, 8W RMS, USB, Negro
10	0	Ghia Bocina Portátil BX900, Bluetooth, Inalámbrico, 2.1 Canales, 34W, USB, Negro - Resistente al Agua
11	0	Ghia Bocina Portátil BX800, Bluetooth, Inalámbrico, 2.1 Canales, 31W, USB, Negro
12	0	Naceb Bocina Portátil NA-0301, Bluetooth, Inalámbrico, USB 2.0, Rojo
13	0	Ghia Bocina Portátil BX300, Bluetooth, Inalámbrico, 40W RMS, USB, Rojo - Resistente al Agua
14	0	Verbatim Bocina Portátil Mini, Bluetooth, Inalámbrico, 3W RMS, USB, Blanco
15	0	Acteck Bocina con Subwoofer AXF-290, Bluetooth, Inalámbrico, 2.1, 18W RMS, 180W PMPO, USB, Negro
16	0	Lenovo Barra de Sonido, Alámbrico, 2.5W, USB, Negro
17	0	Samsung Smart TV LED UN55TU7000FXZX 55, 4K Ultra HD, Widescreen, Negro/Gris
18	0	Hisense Smart TV LED 50H8F 49.5, 4K Ultra HD, Widescreen, Negro
19	0	Samsung Smart TV LED UN32J4290AF 32, HD, Widescreen, Negro
20	0	Samsung Smart TV LED 43, Full HD, Widescreen, Negro
21	0	Hisense Smart TV LED 40H5500F 39.5, Full HD, Widescreen, Negro
22	0	Makena Smart TV LED 40S2 40'', Full HD, Widescreen, Negro
23	0	Samsung Smart TV LED UN70RU7100FXZX 70, 4K Ultra HD, Widescreen, Negro
24	0	Samsung TV LED LH43QMREBGCXGO 43, 4K Ultra HD, Widescreen, Negro
25	0	Seiki TV LED SC-39HS950N 38.5, HD, Widescreen, Negro
26	0	Makena Smart TV LED 32S2 32'', HD, Widescreen, Gris
27	0	Kit Memoria RAM Corsair Vengeance LPX DDR4, 2400MHz, 32GB, Non-ECC, CL16
28	0	SSD Samsung 860 EVO, 1TB, SATA III, M.2
29	0	SSD para Servidor Lenovo Thinksystem S4510, 480GB, SATA III, 2.5'', 7mm
30	0	SSD para Servidor Lenovo Thinksystem S4500, 480GB, SATA III, 3.5'', 7mm
31	0	SSD para Servidor Supermicro SSD-DM128-SMCMVN1, 128GB, SATA III, mSATA, 6Gbit/s
32	0	SSD Addlink Technology S70, 512GB, PCI Express 3.0, M.2
33	0	Tarjeta Madre ASUS ATX ROG STRIX Z390-E GAMING, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel
34	0	Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM, S-1151, Intel H370, HDMI, 64GB DDR4 para Intel
35	0	ASUS T. Madre uATX M4A88T-M, S-AM3, DDR3 para Phenom II/Athlon II/Sempron 100
36	0	Tarjeta Madre Gigabyte Micro ATX H310M DS2 2.0, S-1151, Intel H310, 32GB DDR4 para Intel
37	0	Tarjeta Madre ASRock ATX Z490 STEEL LEGEND, S-1200, Intel Z490, HDMI, 128GB DDR4 para Intel
38	0	Tarjeta Madre Gigabyte micro ATX Z490M GAMING X (rev. 1.0), Intel Z490, HDMI, 128GB DDR4 para Intel
39	0	Tarjeta Madre Gigabyte micro ATX Z390 M GAMING, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel
40	0	Tarjeta Madre ASUS ATX ROG STRIX B550-F GAMING WI-FI, S-AM4, AMD B550, HDMI, max. 128GB DDR4 para AMD
41	0	Tarjeta Madre ASRock Z390 Phantom Gaming 4, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel
42	0	Tarjeta Madre AORUS ATX Z390 ELITE, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel
43	0	Tarjeta de Video VisionTek AMD Radeon HD5450, 2GB GDDR3, PCI Express x16
44	0	Tarjeta de Video VisionTek AMD Radeon HD 5450, 1GB DDR3, PCI Express x16 2.1
45	0	Tarjeta de Video PNY NVIDIA GeForce RTX 2080, 8GB 256-bit GDDR6, PCI Express 3.0
46	0	Tarjeta de Video MSI Radeon X1550, 128MB 64 bit GDDR2, PCI Express x16
47	0	Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC, 8 GB 256 bit GDDR6, PCI Express x16 3.0
48	0	Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile, 4GB 128-bit GDDR5, PCI Express 3.0 x16
49	0	Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming, 6GB 192-bit GDDR6, PCI Express 3.0
50	0	Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0

3.3 50 productos más buscados

Hay altas búsquedas de productos, pero no se tiene esa cantidad de ventas.

50 productos más buscados:

N°	busquedas	Producto
1	263	SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm
2	107	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm
3	60	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
4	55	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth
5	41	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith
6	35	Logitech Audifonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul
7	32	TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro
8	31	Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)
9	30	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)
10	30	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2
11	27	SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2
12	25	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AMD B450, 64GB DDR4 para AMD
13	24	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth
14	23	Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
15	20	Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake)
16	15	Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0
17	15	Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC, 8GB 128-bit GDDR6, PCI Express 4.0
18	15	TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Widescreen, Negro
19	11	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030, 2GB 64-bit GDDR5, PCI Express x16 3.0
20	11	SSD Kingston UV500, 480GB, SATA III, mSATA
21	10	Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache
22	10	Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake)
23	10	Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming, 8GB 128-bit GDDR6, PCI Express 4.0
24	10	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
25	10	Tarjeta Madre Gigabyte XL-ATX TRX40 Designare, S-STRX4, AMD TRX40, 256GB DDR4 para AMD
26	10	Kit SSD Kingston KC600, 1TB, SATA III, 2.5, 7mm
27	10	Logitech Audifonos Gamer G332, Alámbrico, 2 Metros, 3.5mm, Negro/Rojo
28	7	SSD Crucial MX500, 1TB, SATA III, M.2
29	7	Cougar Audifonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro.
30	6	Logitech Bocinas para Computadora con Subwoofer G560, Bluetooth, Inalámbrico, 2.1, 120W RMS, USB, negro
31	6	HyperX Audifonos Gamer Cloud Flight para PC/PS4/PS4 Pro, Inalámbrico, USB, 3.5mm, Negro
32	5	Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Express 3.0
33	5	Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC, 4GB 128-bit GDDR5, PCI Express x16 3.0
34	5	Tarjeta de Video VisionTek AMD Radeon HD 5450, 1GB DDR3, PCI Express x16 2.1
35	5	Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti, 6GB 192-bit GDDR6, PCI Express x16 3.0
36	5	SSD Western Digital WD Blue 3D NAND, 2TB, M.2
37	4	Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0
38	4	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel
39	4	Seiki TV LED SC-39HS950N 38.5, HD, Widescreen, Negro
40	4	Samsung Smart TV LED UN55TU7000FXZX 55, 4K Ultra HD, Widescreen, Negro/Gris
41	3	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0
42	3	ASUS T. Madre uATX M4A88T-M, S-AM3, DDR3 para Phenom II/Athlon II/Sempron 100
43	3	Iogear Audifonos Gamer GHG601, Alámbrico, 1.2 Metros, 3.5mm, Negro
44	2	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix, 4GB 128-bit GDDR5, PCI Express 3.0
45	2	SSD para Servidor Lenovo Thinksystem S4500, 480GB, SATA III, 3.5'', 7mm
46	2	Acteck Bocina con Subwoofer AXF-290, Bluetooth, Inalámbrico, 2.1, 18W RMS, 180W PMPO, USB, Negro
47	2	Genius GHP-400S Audifonos, Alámbrico, 1.5 Metros, Rosa
48	1	Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)
49	1	MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0
50	1	Tarjeta de Video VisionTek AMD Radeon HD5450, 2GB GDDR3, PCI Express x16

3.4 20 productos con peores reseñas

Sobre las reseñas, todos los productos que tienen reseñas en promedio son arriba de 4.5, así que los productos son satisfactorios.

No hay necesidad de cambiar la calidad de los productos

20 productos con las peores reseñas:

N°	Reseña	Producto	Promedio
1	1.00	Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel	
2	1.00	Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0	
3	1.83	Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	
4	2.00	Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel	
5	3.00	Cougar Audifonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro.	
6	4.00	HyperX Audifonos Gamer Cloud Flight para PC/PS4/PS4 Pro, Inalámbrico, USB, 3.5mm, Negro	
7	4.00	Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix, 4GB 128-bit GDDR5, PCI Express 3.0	
8	4.00	MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0	
9	4.14	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	
10	4.23	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth	
11	4.40	Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030, 2GB 64-bit GDDR5, PCI Express x16 3.0	
12	4.46	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador	
13	4.50	Logitech Bocinas para Computadora con Subwoofer G560, Bluetooth, Inalámbrico, 2.1, 120W RMS, USB, negro	
14	4.50	Tarjeta Madre ASUS ATX PRIME Z390-A, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel	
15	4.55	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2	
16	4.56	Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD	
17	4.67	SSD Kingston UV500, 480GB, SATA III, mSATA	
18	4.67	SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2	
19	4.67	Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AMD B450, 64GB DDR4 para AMD	
20	4.70	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)	

3.5 Porcentaje de ganancias por categoria

Vemos que las categorias que producen las mayores ganancias solo son 4, la tienda puede enfocarse solo en esa áreas.

Porcentaje de ventas por categoria

procesadores	trajetas de video	tarjetas madre	discos duros	memorias usb	pantallas	bocinas	audifonos
48.90	17.92	16.75	12.30	0.33	1.48	1.12	1.20

3.6 Ganancias Total

Es en Junio, Julio y Agosto donde las ventas cayeron a menos del 50% de las que se tenían

Total de ingresos 2020: \$1175610		
Total de ventas 2020: 283		
Total de devoluciones 2020: 9		
Mes	Ingresos	Ventas
Abril	\$ 313596	74
Marzo	\$ 216261	49
Mayo	\$ 187006	34
Febrero	\$ 178464	40
Enero	\$ 177138	52
Julio	\$ 51449	11
Junio	\$ 39759	11
Agosto	\$ 11937	3
Septiembre	\$ 0	0
Octubre	\$ 0	0
Noviembre	\$ 0	0
Diciembre	\$ 0	0

4 Conclusión

Se recomienda eliminar las siguientes categorías de productos: Memorias usb, pantallas, bocinas, audifonos. Y solo quedarse con: Procesadores, tarjetas de video, tarjetas madre.

Los productos de rojo en la tabla **50 productos con mayores ventas** se deben rellenar ya que están agotados, y algunos tienen menor cantidad de lo que se ha llegado a vender, así que pueden hacer falta.

De las categorías que queden disponibles por lo antes mencionado, solo se queden los productos que han mostrado ventas en la lista antes mencionada, y agregar un stock mínimo de los productos en la lista **50 productos más buscados**.

No se debe cambiar de distribuidor ya o marcas ya que los productos vendidos muestran buenas reseñas.

De los productos vendidos se recomienda tomar el número de sus ventas en estos meses, y aumentar su stock en aproximadamente un 30% de cada producto, esperando aumentar las ventas en Otoño e invierno.