

EMTECH



# *PROYECTO 1. INTRODUCCIÓN A PYTHON*

*Ramos Reyes  
Lizeth Montserrat*

A large, stylized graphic featuring the word "PYTHON" in a bold, white, sans-serif font. The text is set against a dark blue background with a network of glowing white lines and nodes. Several rectangular frames containing binary code (0s and 1s) are scattered around the text, creating a digital and technological aesthetic.

**PYTHON**

## Tabla de contenido

Introducción.....	3
Definición del código.....	4
Bienvenida e inicio de sesión .....	4
Menú al iniciar sesión.....	7
1. Ver los productos con mayores ventas.....	8
2. Ver los productos con mayores búsquedas .....	10
3. Ver los productos menos vendidos por categorías.....	13
4. Ver los productos menos buscados por categorías .....	15
5. Ver los productos con mejores reseñas.....	17
6. Ver los productos con peores reseñas.....	20
7. Total de ingresos, ventas promedio mensuales y meses con más ventas al año.....	22
Solución al problema.....	25
Conclusiones .....	27
Anexos.....	28

## Introducción

Durante la primera parte del curso de Fundamentos de Programación con Python, impartido por Emtech, se abarcaron temas correspondientes a los fundamentos de programación y control de flujo de un programa.

En la parte de Fundamentos de Programación, se pudieron revisar los siguientes puntos: entendimiento de lo que es un entorno de desarrollo, creación de variables, uso de números, manejo de textos, aplicación de índices y *slicing*, creación de listas, solicitud de ingreso de datos por el usuario, comparaciones, relaciones, operadores lógicos y de asignación.

Para el control del flujo en un programa, se avanzó con el aprendizaje de los siguientes aspectos: Sentencias If, Bucle While, Bucle For y Control de Bucles con Break y Continue.

Para poder aplicar todos los conocimientos adquiridos y evaluar su entendimiento, se desarrolló un proyecto práctico para analizar y clasificar datos de la tienda virtual llamada LifeStone, que está presentando una acumulación elevada de inventario, una reducción de búsquedas en gran cantidad de productos y, por lo tanto, una disminución relevante en sus ventas.

Bajo esta situación, se solicitó realizar un análisis de la rotación de los inventarios, cubriendo tres consignas específicas:

1. Presentar los productos más vendidos y rezagados.
2. Analizar los productos por reseña en el servicio.
3. Sugerir una estrategia de productos a retirar y de reducción de acumulación de inventario.

A continuación, se presenta el desarrollo, los resultados y las propuestas generadas.

## Definición del código.

### Bienvenida e inicio de sesión

El programa desarrollado comienza con una bienvenida al sistema de análisis, a continuación, presenta la pregunta para iniciar sesión o registrarse:

```
Este es el sistema de análisis Emtech. Por favor, inicie sesión  
¿Ya tiene una cuenta? (si/no) █
```

El programa tiene predefinida una lista de usuarios con su contraseña, siendo esta: usuarios\_registrados = [["Emtech", "Curso"] ["Lizeth", "Ramos"]]. En caso de contestar "si", solicitará el usuario y contraseña, de contestar correctamente, se podrá iniciar sesión y entrar al menú de opciones a analizar:

```
Este es el sistema de análisis Emtech. Por favor, inicie sesión  
¿Ya tiene una cuenta? (si/no) si  
Ingresa el usuario: Emtech  
Ingresa la contraseña: Curso  
Sesión iniciada  
  
    ingrese el número de la opción de lo que quiere revisar  
  
    1. Ver los productos con mayores ventas  
    2. Ver los productos con mayores búsquedas  
    3. Ver los productos menos vendidos por categorías  
    4. Ver los productos menos buscados por categorías  
    5. Ver los productos con mejores reseñas  
    6. Ver los productos con peores reseñas  
    7.Total de ingresos, ventas promedio mensuales y meses con más ventas al año  
    8.Salir
```

En caso de poner datos, incorrectos, se tienen 2 intentos más para corregirlos, de lo contrario, se detendrá el programa:

```
Este es el sistema de análisis Emtech. Por favor, inicie sesión  
¿Ya tiene una cuenta? (si/no) si  
Ingresa el usuario: liz  
Ingresa la contraseña: Ramos  
Datos incorrectos  
Ingresa nuevamente el usuario: Lizeth  
Ingresa nuevamente la contraseña: ramos  
Datos incorrectos  
Ingresa nuevamente el usuario: LIZETH  
Ingresa nuevamente la contraseña: RAMOS  
Datos incorrectos, acceso denegado
```

En caso de no tener una cuenta, y contestar con “no” a la pregunta inicial, se solicitará ingresar un nuevo usuario y una nueva contraseña para registrarse, posteriormente se pasa al inicio de sesión:

```
Este es el sistema de análisis Emtech. Por favor, inicie sesión
¿Ya tiene una cuenta? (sí/no) no
ingrese un nombre de usuario Lizeth
ingrese una contraseña RAMOS
se ha registrado con éxito
Ingresa el usuario: Lizeth
Ingresa la contraseña: RAMOS
Sesión iniciada

    ingrese el número de la opción de lo que quiere revisar

    1. Ver los productos con mayores ventas
    2. Ver los productos con mayores búsquedas
    3. Ver los productos menos vendidos por categorías
    4. Ver los productos menos buscados por categorías
    5. Ver los productos con mejores reseñas
    6. Ver los productos con peores reseñas
    7.Total de ingresos, ventas promedio mensuales y meses con más ventas al año
    8.Salir
```

El código que se encarga de estas acciones es el siguiente:

```
6  #Usuarios_registrados[usuario,contraseña]
7  usuarios_registrados=[["Emtech", "Curso"],["Lizeth", "Ramos"]]
8  print("Este es el sistema de análisis Emtech. Por favor, inicie sesión")
9  login=input("¿Ya tiene una cuenta? (si/no) ")
10
11 #inicio de sesión
12 while login=="no":
13     nuevo_usuario=input("ingrese un nombre de usuario ")
14     nueva_contrasena=input("ingrese una contraseña ")
15     usuarios_registrados.append([nuevo_usuario, nueva_contrasena])
16     print("se ha registrado con éxito")
17     login="si"
18
19 usuario_login = input("Ingresa el usuario: ")
20 password = input("Ingresa la contraseña: ")
21
22 registro=0
23 intento = 0
24
25 while login=="si":
26     #intentos de login para ingresar datos correctos
27     while registro != 1 and intento < 2:
28         for usuario in usuarios_registrados:
29             if usuario[0] == usuario_login and usuario[1]==password:
30                 registro = 1
31                 login="no"
32
33         if registro == 0:
34             print("Datos incorrectos")
35             usuario_login = input("Ingresa nuevamente el usuario: ")
36             password = input("Ingresa nuevamente la contraseña: ")
37             intento += 1
38
```

La variable 'login' almacena la respuesta 'si'/'no', para poder pasar al registro o inicio de sesión, la variable 'registro' tiene 0 como señal de que no hay una sesión iniciada todavía, cambia a ser 1 al ingresar un usuario y contraseña correctos para iniciar sesión. La variable 'intento', va contando la cantidad de veces en que se ingresan datos, al llegar a 3, detiene el programa, para evitar ser un bucle infinito.

## Menú al iniciar sesión

Al poder acceder de manera exitosa al programa, se muestra el siguiente menú:

```
ingrese el número de la opción de lo que quiere revisar

1. Ver los productos con mayores ventas
2. Ver los productos con mayores búsquedas
3. Ver los productos menos vendidos por categorías
4. Ver los productos menos buscados por categorías
5. Ver los productos con mejores reseñas
6. Ver los productos con peores reseñas
7.Total de ingresos, ventas promedio mensuales y meses con más ventas al año
8.Salir
```

Mediante estas opciones, se satisfacen las 3 consignas solicitadas, para la primera, que es de “Productos más vendidos y productos rezagados”, para los productos más vendidos se encuentran las opciones 1 y 2, correspondientes a mayores ventas y búsquedas; por otro lado, para los productos rezagados, están las opciones 3 y 4, cuyo análisis está organizado por categoría de los productos.



## 1. Ver los productos con mayores ventas

Al ingresar la opción 1, se ejecuta el siguiente código:

```
64     if accion=="1":
65         print(" 1. Ver los productos con mayores ventas")
66         contador = 0
67         venta_productos = []
68
69         #contar la cantidad de ventas por producto
70         for producto in lifestore_products:
71             for venta in lifestore_sales:
72                 if producto[0] == venta[1]:
73                     contador += 1
74             venta_productos.append([producto[0], contador, producto[3]]) # [id_producto, num_ventas,categoria]
75             contador = 0
76
77         #ordenar las ventas de mayor a menor cantidad
78         ventas_ordenadas=[]
79
80         while venta_productos:
81             minimo=venta_productos[0][1]
82             lista_actual= venta_productos[0]
83             for venta in venta_productos:
84                 if venta[1]> minimo:
85                     minimo=venta[1]
86                     lista_actual=venta
87             ventas_ordenadas.append(lista_actual)
88             venta_productos.remove(lista_actual)
89
90         #mostrar los resultados
91         print("\nEl top 10 de los productos más vendidos es: \n")
92         productos_vendidos=0
93         for valor in ventas_ordenadas:
94             if productos_vendidos<10:
95                 print('El producto ', valor[0], ' se vendió ', valor[1], ' veces, de la categoría ',valor[2])
96                 productos_vendidos+=1
97
98         accion=0
99         print("\nPara análisis de todas las ventas:\n")
100
101         print(tabulate(ventas_ordenadas, headers=["Producto", "Cantidad", "Categoría"])))
```

Lo primero que se realiza, es la creación de la lista `venta_productos`[ID producto, número de ventas, categoría], en donde, mediante un ciclo For, se hace el conteo de las ventas que se tiene por un determinado producto y se añaden a la lista.

A continuación, se ordenan las ventas de manera decreciente al crear la lista `ventas_ordenadas`[], finalmente se presenta el top de los 10 productos más vendidos, presentando el ID, cantidad y la categoría.

Para poder realizar un análisis más profundo, se añade también, en formato de tabla, todas las ventas ordenadas.

Como resultado de las acciones del código, se muestra en pantalla la siguiente información:



## 1. Ver los productos con mayores ventas

El top 10 de los productos más vendidos es:

```
El producto 54 se vendió 50 veces, de la categoría discos duros
El producto 3 se vendió 42 veces, de la categoría procesadores
El producto 5 se vendió 20 veces, de la categoría procesadores
El producto 42 se vendió 18 veces, de la categoría tarjetas madre
El producto 57 se vendió 15 veces, de la categoría discos duros
El producto 29 se vendió 14 veces, de la categoría tarjetas madre
El producto 2 se vendió 13 veces, de la categoría procesadores
El producto 4 se vendió 13 veces, de la categoría procesadores
El producto 47 se vendió 11 veces, de la categoría discos duros
El producto 12 se vendió 9 veces, de la categoría tarjetas de video
```

Para análisis de todas las ventas:

Producto	Cantidad	Categoría
54	50	discos duros
3	42	procesadores
5	20	procesadores
42	18	tarjetas madre
57	15	discos duros
29	14	tarjetas madre
2	13	procesadores
4	13	procesadores
47	11	discos duros
12	9	tarjetas de video
48	9	discos duros
7	7	procesadores
31	6	tarjetas madre
44	6	tarjetas madre
18	5	tarjetas de video
8	4	procesadores
6	3	procesadores
11	3	tarjetas de video
49	3	discos duros

Con estos datos se puede notar que las categorías que genera la mayor cantidad de ventas en la tienda son 3: Discos duros, procesadores, tarjetas madre.

## 2. Ver los productos con mayores búsquedas

Al terminar de mostrar los resultados de la opción elegida, se muestra nuevamente el menú de acciones, para volver a elegir qué tarea realizar. Para analizar los productos con la mayor cantidad de búsquedas, el programa cuenta con el siguiente código:

```
elif accion=="2":
    print("2. Ver los productos con mayores búsquedas")
    contador = 0
    busqueda_productos = []
    #conteo de cantidad de búsquedas
    for producto in lifestore_products:
        for busqueda in lifestore_searches:
            if producto[0] == busqueda[1]:
                contador += 1
    busqueda_productos.append([producto[0], contador, producto[3]]) # [id_producto,
    num_ventas, categoria]
    contador = 0

    #ordenar cantidad de búsquedas de mayor a menor
    busquedas_ordenadas=[]

    while busqueda_productos:
        minimo=busqueda_productos[0][1]
        lista_actual= busqueda_productos[0]
        for buscar in busqueda_productos:
            if buscar[1]> minimo:
                minimo=buscar[1]
                lista_actual=buscar
        busquedas_ordenadas.append(lista_actual)
        busqueda_productos.remove(lista_actual)

    #imprimir los 10 productos más buscados
    print("\nLos 10 productos con mayores búsquedas:\n")
    productos_buscados=0
    for valor in busquedas_ordenadas:
        if productos_buscados<10:
            print('El producto ', valor[0], ' se buscó ', valor[1], ' veces', ' de la categoría '
            '+ valor[2])
            productos_buscados+=1

    print("\nPara análisis de todas las búsquedas:\n")

    print(tabulate(busquedas_ordenadas, headers=["Producto", "Búsquedas", "Categoria"]))
    accion=0
```

Para esta acción, se repite el código de la primera opción, pero cambiando de ventas a búsquedas. Se inicia contando la cantidad de búsquedas por producto, se pasa a ordenar de manera descendiente y se presenta el top 10 de los productos vendidos. De igual manera, para fines de análisis profundo, se permite ver la tabla de todos los productos.

Como resultado, el usuario recibe:

## 2. Ver los productos con mayores búsquedas

Los 10 productos con mayores búsquedas:

```
El producto 54 se buscó 263 veces de la categoría discos duros
El producto 57 se buscó 107 veces de la categoría discos duros
El producto 29 se buscó 60 veces de la categoría tarjetas madre
El producto 3 se buscó 55 veces de la categoría procesadores
El producto 4 se buscó 41 veces de la categoría procesadores
El producto 85 se buscó 35 veces de la categoría audifonos
El producto 67 se buscó 32 veces de la categoría pantallas
El producto 7 se buscó 31 veces de la categoría procesadores
El producto 5 se buscó 30 veces de la categoría procesadores
El producto 47 se buscó 30 veces de la categoría discos duros
```

Para análisis de todas las búsquedas:

Producto	Búsquedas	Categoría
54	263	discos duros
57	107	discos duros
29	60	tarjetas madre
3	55	procesadores
4	41	procesadores
85	35	audifonos
67	32	pantallas
7	31	procesadores
5	30	procesadores
47	30	discos duros
48	27	discos duros
44	25	tarjetas madre
2	24	procesadores
42	23	tarjetas madre
8	20	procesadores
12	15	tarjetas de video
21	15	tarjetas de video
66	15	pantallas
18	11	tarjetas de video

Se puede notar una diferencia respecto a los resultados obtenidos al seleccionar la opción 1, pues, si bien el producto 54 se mantiene liderando la lista, en esta ocasión, las categorías que más búsquedas reciben son: Discos duros, tarjetas madre, procesadores, añadiéndose audífonos, y pantallas en 6° y 7° posición. Lo que puede mostrar que los clientes están interesados en los productos que se ofertan en esta categoría, pero la

información que encuentran no les convence para proceder a la compra. Otro aspecto interesante para resaltar es que, a pesar de que el producto 3 no se encuentra en las listas de los artículos más buscados, sí está en la segunda posición de los más vendidos, con lo que se puede suponer que los clientes ya conocen la marca o el producto y optan por esa opción debido que logran satisfacer mediante esta sus necesidades. Lo mismo sucede con el artículo 42, 2 y 12, que se encuentran en el top 10 de la lista de ventas.

Un aspecto más que pudiera inferirse con los resultados vistos hasta el momento, es que se requiere una campaña de publicidad efectiva para la propaganda y oferta de los productos con los que se cuenta, pues se puede ver del producto 54, el porcentaje que pasa de búsquedas a compra exitosa es menor al 20%. Se pudiera pensar que los clientes, al buscar por el producto que requieren, revisan numerosas opciones y están encontrando en la competencia ofertas que hacen que opten por ellas, por lo cual puede ser beneficioso que se implementen estrategias de benchmarking para retener de manera exitosa a una mayor cantidad de clientes potenciales.

### 3. Ver los productos menos vendidos por categorías

Al entrar en esta opción, se obtiene como resultado lo contrario a la opción 1, pues se regresa una lista con los productos menos vendidos, pero separados por categoría. Se utiliza el siguiente código:

```
143 print('3. Ver los productos menos vendidos por categorías')
144 contador = 0
145 venta_productos = []
146 for producto in lifestore_products:
147     for venta in lifestore_sales:
148         if producto[0] == venta[1]:
149             contador += 1
150     venta_productos.append([producto[0], contador, producto[3]]) # [id_producto,
151                             num_ventas, categoria]
152     contador = 0
153     ventas_ordenadas=[]
154
155 while venta_productos:
156     minimo=venta_productos[0][1]
157     categorias=venta_productos[0][2]
158     lista_actual= venta_productos[0]
159     for venta in venta_productos:
160         if venta[1] > minimo:
161             minimo=venta[1]
162             lista_actual=venta
163     ventas_ordenadas.append(lista_actual)
164     venta_productos.remove(lista_actual)
165     ventas_ordenadas_categoria=[]
166     while ventas_ordenadas:
167         categorias=ventas_ordenadas[0][2]
168         lista_actual= ventas_ordenadas[0]
169         for venta in ventas_ordenadas:
170             if venta[2] == categorias:
171                 categorias=venta[2]
172                 lista_actual=venta
173         ventas_ordenadas_categoria.append(lista_actual)
174         ventas_ordenadas.remove(lista_actual)
175     print("\nPara análisis de los productos con menos ventas por categorías
176         ordenados de menor a mayor:\n")
177
178     print(tabulate(ventas_ordenadas_categoria, headers=["Producto", "ventas",
179                                                         "Categoria"])))
```

En este, se comienza añadiendo todos los productos, con su cantidad de ventas y categoría a la lista `venta_productos=[]`, a continuación se ordenan de manera ascendente en una nueva lista, llamada `ventas_ordenadas`, y se continúa con la creación de una nueva lista llamada `ventas_ordenadas_categoria=[]`, en donde se agrupan todos los artículos según la categoría a la que pertenecen, con esto se tiene de resultado:

### 3. Ver los productos menos vendidos por categorías



Para análisis de los productos con menos ventas por categorías ordenados de menor a mayor:

Producto	ventas	Categoría
59	0	discos duros
58	0	discos duros
56	0	discos duros
55	0	discos duros
53	0	discos duros
50	1	discos duros
52	2	discos duros
51	3	discos duros
49	3	discos duros
48	9	discos duros
47	11	discos duros
57	15	discos duros
54	50	discos duros
9	0	procesadores
1	2	procesadores
6	3	procesadores
8	4	procesadores
7	7	procesadores
4	13	procesadores
2	13	procesadores
5	20	procesadores
3	42	procesadores
43	0	tarjetas madre
41	0	tarjetas madre
39	0	tarjetas madre
38	0	tarjetas madre
37	0	tarjetas madre
36	0	tarjetas madre
35	0	tarjetas madre
34	0	tarjetas madre
32	0	tarjetas madre

Con estos datos, es importante notar que las categorías que generan una menor cantidad de ventas son: Bocinas y audífonos. Bocinas no figura ni en mayores ventas, ni búsquedas, por otro lado, la categoría de audífonos sí cuenta con artículos en el top 10 de búsquedas.

Por lo que, una campaña de marketing adecuada puede aumentar la cantidad de ventas de esta categoría, pudiendo tomar como primera acción la oferta de precios especiales para que las personas interesadas completen la compra después de realizar su búsqueda.

#### 4. Ver los productos menos buscados por categorías

En esta siguiente opción, se sigue un proceso semejante al anterior, ahora respecto a las búsquedas, comienza creando la lista `búsqueda_productos=[]` y llenándola con los datos de `lifestore_productos` y `lifestore_searches`, posteriormente se ordenan de manera ascendente. El código regresa, a manera de tabla, los resultados de la lista nombrada `búsquedas_ordenadas_categoria`.

```
180 #Productos menos buscados por categorías
181 elif accion=="4":
182     print("4. Ver los productos menos buscados por categorías")
183     contador = 0
184     busqueda_productos = []
185     for producto in lifestore_productos:
186         for busqueda in lifestore_searches:
187             if producto[0] == busqueda[1]:
188                 contador += 1
189     busqueda_productos.append([producto[0], contador, producto[3]]) # [id_producto, num_ventas,categoria]
190     contador = 0
191
192     busquedas_ordenadas=[]
193
194     while busqueda_productos:
195         minimo=busqueda_productos[0][1]
196         categorias=busqueda_productos[0][2]
197         lista_actual= busqueda_productos[0]
198         for buscar in busqueda_productos:
199             if buscar[1] > minimo:
200                 minimo=buscar[1]
201                 lista_actual=buscar
202     busquedas_ordenadas.append(lista_actual)
203     busqueda_productos.remove(lista_actual)
204     busquedas_ordenadas_categoria=[]
205     while busquedas_ordenadas:
206         categorias=busquedas_ordenadas[0][2]
207         lista_actual= busquedas_ordenadas[0]
208         for buscar in busquedas_ordenadas:
209             if buscar[2] == categorias:
210                 categorias=buscar[2]
211                 lista_actual=buscar
212     busquedas_ordenadas_categoria.append(lista_actual)
213     busquedas_ordenadas.remove(lista_actual)
214
215     print("\nPara análisis de los productos con menos búsquedas por categoría ordenados de menor a mayor:\n")
216
217     print(tabulate(busquedas_ordenadas_categoria, headers=["Producto", "búsquedas", "Categoría"]))
218     ...
```



Como resultado, se observa la siguiente tabla:

4. Ver los productos menos buscados por categorías

Para análisis de los productos con menos búsquedas por categoría ordenados de menor a mayor:

Producto	búsquedas	Categoría
58	0	discos duros
55	0	discos duros
53	0	discos duros
59	1	discos duros
56	2	discos duros
52	5	discos duros
50	7	discos duros
49	10	discos duros
51	11	discos duros
48	27	discos duros
47	30	discos duros
57	107	discos duros
54	263	discos duros
43	0	tarjetas madre
41	0	tarjetas madre
38	0	tarjetas madre
37	0	tarjetas madre
36	0	tarjetas madre
34	0	tarjetas madre
33	0	tarjetas madre
32	0	tarjetas madre
30	0	tarjetas madre
45	1	tarjetas madre
35	1	tarjetas madre
39	3	tarjetas madre
46	4	tarjetas madre
40	10	tarjetas madre
31	10	tarjetas madre
42	23	tarjetas madre
44	25	tarjetas madre
29	60	tarjetas madre

Con los resultados presentados, se puede hacer un análisis más a fondo, se puede notar que en una gran cantidad de productos no solo no se tienen ventas, sino que tampoco hay búsquedas respecto a ellos, por lo que se pudieran hacer distintas hipótesis:

- a) El público meta no los conoce.
- b) Sus características no son de interés para el público.

Para esclarecer estos puntos, es necesario que la empresa realice un estudio de mercado y tenga bien estipulado el perfil de su *customer persona*, pues de esta manera, se podrá realizar la estrategia de marketing óptima, con objetivos alcanzables y mayor probabilidad de éxito.

## 5. Ver los productos con mejores reseñas

Para poder ordenar los productos según la calificación registrada sobre venta, se inició con la creación de la lista llamada 'info', en ella se rellenaron los datos con `lifestore_products` y `lifestore_sales`, después se hizo el cálculo de los promedios de evaluación en una lista llamada `info2`, finalmente, los promedios se ordenaron en la lista `resenas_ordenadas` y se mostraron los 20 productos con mejor calificación promedio y su cantidad de devoluciones.

```
224 elif accion=='5':
225     print("5. Ver los productos con mejores reseñas")
226     info=[]#producto, cantidad ventas, calificación, devoluciones
227     for producto in lifestore_products:
228         info.append([producto[0],0,0,0]) #[producto, cantidad de ventas, calificación, devoluciones]
229     for producto in info:
230         id=producto[0]
231         for score in lifestore_sales:
232             if id==score[1]:
233                 info[id-1][2]+=score[2]
234                 info[id-1][1]+=1
235                 info[id-1][3]+=score[4]
236
237     info2=[]#producto, cantidad ventas, evaluación, devoluciones
238     for producto in lifestore_products:
239         info2.append([producto[0],0,0,0]) #[producto, cantidad de ventas, calificación, devoluciones]
240     for producto in info2:
241         id=producto[0]
242         for promedio in info:
243             if id==promedio[0]:
244                 info2[id-1][1]=promedio[1]
245                 info2[id-1][3]=promedio[3]
246                 if promedio[2]>0:
247                     info2[id-1][2]=promedio[2]/promedio[1]
248
249     resenas_ordenadas=[]
250
251     while info2:
252         minimo=info2[0][2]
253         lista_actual= info2[0]
254         for producto in info2:
255             if producto[2] > minimo:
256                 minimo=producto[2]
257                 lista_actual=producto
258         resenas_ordenadas.append(lista_actual)
259         info2.remove(lista_actual)
260     #presentación de los 20 productos mejor calificados
261     print( '\n Los 20 productos con mejor calificación promedio \n')
```

Se muestra también la tabla de todas las ventas y su promedio para analizar.

Producto	1	ventas	2	Calificación	5.0	Devoluciones	0
Producto	6	ventas	3	Calificación	5.0	Devoluciones	0
Producto	7	ventas	7	Calificación	5.0	Devoluciones	0
Producto	8	ventas	4	Calificación	5.0	Devoluciones	0
Producto	11	ventas	3	Calificación	5.0	Devoluciones	0
Producto	21	ventas	2	Calificación	5.0	Devoluciones	0
Producto	22	ventas	1	Calificación	5.0	Devoluciones	0
Producto	25	ventas	2	Calificación	5.0	Devoluciones	0
Producto	28	ventas	1	Calificación	5.0	Devoluciones	0
Producto	40	ventas	1	Calificación	5.0	Devoluciones	0
Producto	49	ventas	3	Calificación	5.0	Devoluciones	0
Producto	50	ventas	1	Calificación	5.0	Devoluciones	0
Producto	52	ventas	2	Calificación	5.0	Devoluciones	0
Producto	60	ventas	1	Calificación	5.0	Devoluciones	0
Producto	66	ventas	1	Calificación	5.0	Devoluciones	0
Producto	67	ventas	1	Calificación	5.0	Devoluciones	0
Producto	84	ventas	1	Calificación	5.0	Devoluciones	0
Producto	85	ventas	2	Calificación	5.0	Devoluciones	0
Producto	57	ventas	15	Calificación	4.866666666666666	Devoluciones	0
Producto	3	ventas	42	Calificación	4.809523809523809	Devoluciones	0
Producto	12	ventas	9	Calificación	4.777777777777778	Devoluciones	0

  

Análisis de todos los productos según su calificación promedio

Producto	Ventas	Evaluación	Devoluciones
-----	-----	-----	-----
1	2	5	0
6	3	5	0
7	7	5	0
8	4	5	0
11	3	5	0
21	2	5	0
22	1	5	0
25	2	5	0
28	1	5	0
40	1	5	0
49	3	5	0

Con los resultados obtenidos de esta opción se pueden recalcar los siguientes aspectos:

Es posible confirmar la hipótesis anterior, una gran cantidad de artículos no son conocidos, pues no cuentan con búsquedas, ni compras, por lo tanto, tampoco con evaluación. Se necesita decidir si se van a continuar ofertando estos productos o se van a abandonar, pues al mantenerlos en stock, se incurre en egresos importantes para la empresa.

89	1	3	0
46	1	2	1
31	6	1.83333	3
17	1	1	1
45	1	1	1
9	0	0	0
14	0	0	0
15	0	0	0
16	0	0	0
19	0	0	0
20	0	0	0
23	0	0	0
24	0	0	0
26	0	0	0
27	0	0	0
30	0	0	0
32	0	0	0
34	0	0	0
35	0	0	0
36	0	0	0
37	0	0	0
38	0	0	0
39	0	0	0
41	0	0	0
43	0	0	0
53	0	0	0
55	0	0	0
56	0	0	0
58	0	0	0
59	0	0	0
61	0	0	0
62	0	0	0
63	0	0	0
64	0	0	0

También se pueden notar otros productos que tienen ventas escasas y devoluciones totales o mayores al 50%, así como bajas calificaciones. Es necesario retirar estos productos del mercado para no dañar la reputación de la calidad de productos ofertados por la empresa, pues puede conllevar a una baja importante de ventas y pérdida de clientes. Es importante que se retiren los productos 89, 46, 31, 17 y 45.

Además, la empresa debe trabajar en su sistema de CRM, pues los clientes que han adquirido, evaluado y devuelto estos productos son las personas a las que se les debe ofrecer alternativas que puedan remendar la imagen que tiene la empresa ante ellas. Es bien conocido en el marketing que un cliente insatisfecho va a comunicar su experiencia negativa con una mucha mayor cantidad de gente que alguien satisfecho.

## 6. Ver los productos con peores reseñas

En esta opción se repite el código de la opción pasada, pero presentando en el orden contrario.

```
276 elif accion=='6':
277     print("6. Ver los productos con peores reseñas")
278     info=[]#producto, cantidad ventas, calificación, devoluciones
279     for producto in lifestore_products:
280         info.append([producto[0],0,0,0]) #[producto, cantidad de ventas, calificación, devoluciones]
281     for producto in info:
282         id=producto[0]
283         for score in lifestore_sales:
284             if id==score[1]:
285                 info[id-1][2]+=score[2]
286                 info[id-1][1]+=1
287                 info[id-1][3]+=score[4]
288
289     4
290     info2=[]#producto, cantidad ventas, promedio evaluación, devoluciones
291     for producto in lifestore_products:
292         info2.append([producto[0],0,0,0]) #[producto, cantidad de ventas, calificación promedio, devoluciones]
293     for producto in info2:
294         id=producto[0]
295         for promedio in info:
296             if id==promedio[0]:
297                 info2[id-1][1]=promedio[1]
298                 info2[id-1][3]=promedio[3]
299                 if promedio[2]>0:
300                     info2[id-1][2]=promedio[2]/promedio[1]
301
302     resenas_ordenadas=[]
303
304     while info2:
305         minimo=info2[0][2]
306         lista_actual= info2[0]
307         for producto in info2:
308             if producto[2] < minimo:
309                 minimo=producto[2]
310                 lista_actual=producto
311         resenas_ordenadas.append(lista_actual)
312         info2.remove(lista_actual)
313
314
315     print( '\n Los 50 productos con las calificaciones más bajas \n')
```

Los resultados en esta opción nos comprueban que una gran cantidad de productos no han sido conocidos por el público meta, pues no tienen ninguna venta, así como diversos de ellos no han tenido ninguna búsqueda registrada. Se pudiera concluir con que la empresa maneja mucho inventario fantasma que cuenta con muy pocas probabilidades de rotar debido a desconocimiento de su oferta.

Producto	Ventas	Evaluación	Devoluciones
9	0	0	0
14	0	0	0
15	0	0	0
16	0	0	0
19	0	0	0
20	0	0	0
23	0	0	0
24	0	0	0
26	0	0	0
27	0	0	0
30	0	0	0
32	0	0	0
34	0	0	0
35	0	0	0
36	0	0	0
37	0	0	0
38	0	0	0
39	0	0	0
41	0	0	0
43	0	0	0
53	0	0	0
55	0	0	0
56	0	0	0
58	0	0	0
59	0	0	0
61	0	0	0
62	0	0	0
63	0	0	0
64	0	0	0
65	0	0	0
68	0	0	0
69	0	0	0
70	0	0	0
71	0	0	0
72	0	0	0
73	0	0	0
75	0	0	0
76	0	0	0
77	0	0	0
78	0	0	0
79	0	0	0
80	0	0	0

## 7. Total de ingresos, ventas promedio mensuales y meses con más ventas al año

En esta opción se realizaron una mayor cantidad de procesos. Primero se realizó una lista llamada 'info' para registrar la venta del producto por [ID, Mes, Año, Devolución, Costo, Ganancia], para esto se utilizaron las listas de `lifestore_sales` y `lifestore_products`.

Una vez que se tuvieron estos datos, se ordenaron por año al crear una lista llamado `year_ordenadas`, con esto se pudo encontrar que la cantidad de ventas realizadas en años diferentes a 2020 era mínima y no representaban una diferencia significativa, por lo que

```
337 info=[]
338
339 for producto in lifestore_sales:
340     info.append([producto[1], producto[3][3:5], producto[3]
341                 [6:10], producto[4], 0, 0]) #0 producto, 1 mes, 2 año,
342                 3 devolución, 4 costo, 5 ganancia
341 i=0
342 for producto in info:
343     id=producto[0]
344
345
346     for venta in lifestore_products:
347         if id==venta[0]:
348             info[i][4]=venta[2]
349             if info[i][3]==1:
350                 info[i][5]=0
351             else:
352                 info[i][5]=info[i][4]
353             i+=1
354
355 #orden por años
356 year_ordenadas=[]
357 while info:
358     year2=int(info[0][2])
359     lista_actual= info[0]
360     for fecha in info:
361         if int(fecha[2])> int(year2):
362             year2=fecha[2]
363             lista_actual=fecha
364     year_ordenadas.append(lista_actual)
365     info.remove(lista_actual)
366
367 #orden por meses
368 mes_ordenados=[]
369 while year_ordenadas:
370     mes2= int(year_ordenadas[0][1])
371     lista_actual= year_ordenadas[0]
372     for mes in year_ordenadas:
373         if int(mes[1])> int(mes2):
374             mes2=mes[1]
375             lista_actual=mes
376     mes_ordenados.append(lista_actual)
377     year_ordenadas.remove(lista_actual)
```



se omitió un desarrollo más profundo de esta clasificación y se pasó a ordenar los meses mediante la lista llamada `mes_ordenados`.

Una vez que se ordenaron las ventas por mes, se pasó a contar la cantidad de ganancias totales de la empresa, nombrando a la variable 'ganancias'.

A continuación, se desarrollo el cálculo de ganancias totales por mes y su promedio.

Más adelante, se ordenaron los meses según sus ganancias y promedios

```
379     i=0
380     ganancia=0
381     #Cálculo del total de ganancias
382     for registro in mes_ordenados:
383         ganancia+=mes_ordenados[i][5]
384         i+=1
385     #total_ingresos.append(ganancia)
386     print("\nEl total de ganancias de la empresa es de: $",
ganancia, '.00\n')

387
388     i=0
389     ganancia=0
390     mes=12
391     total=0
392     ventas_por_mes=[]#mes,ventas
393     for producto in mes_ordenados:
394         #mes=mes-1
395         ganancia=0
396         mes=int(producto[1])
397         if int(producto[1])==int(mes):
398             ganancia+=producto[5]
399             total+=ganancia
400
401     ventas_por_mes.append([int(producto[1]),ganancia])
402
403
404     i=0
405     ganancia=0
406     mes=12
407     total=0
408     #ventas_por_mes=[]#mes,ventas
409     for producto in mes_ordenados:
410         #mes=mes-1
411         ganancia=0
412         mes=int(producto[1])
413         if int(producto[1])==int(mes):
414             ganancia+=producto[5]
415             total+=ganancia
416
417     ganancia=0
418     i=0
419     while mes>0 and mes<12:
420         for venta in ventas_por_mes:
421             if venta[0]==mes:
ganancia+=venta[1]
```

Se presenta en la interfaz:

Se puede ver que los meses con mayores ganancias promedio son el 4, 1, 3 y 2, respectivamente. Se puede notar un primer decremento importante en las ganancias a partir del segundo semestre del año, después un segundo decremento importante a partir del mes 8, que se mantiene durante todo el trimestre siguiente.

```
El total de ganancias de la empresa es de: $ 737916 .00

Ventas mensuales por ganancia total y promedio

Mes Ventas Ganancia Promedio
-----
11 1 0 0
10 0 0 0
9 1 0 0
8 3 3077 1025.67
7 11 26949 2449.91
6 11 36949 3359
5 36 91936 2553.78
4 75 191066 2547.55
3 51 162931 3194.73
2 41 107270 2616.34
1 53 117738 2221.47

Orden de los meses según las mayores ganancias totales obtenidas:

Mes 4 Cantidad de ventas 75 Ganancia 191066
Mes 1 Cantidad de ventas 53 Ganancia 117738
Mes 3 Cantidad de ventas 51 Ganancia 162931
Mes 2 Cantidad de ventas 41 Ganancia 107270
Mes 5 Cantidad de ventas 36 Ganancia 91936
Mes 7 Cantidad de ventas 11 Ganancia 26949
Mes 6 Cantidad de ventas 11 Ganancia 36949
Mes 8 Cantidad de ventas 3 Ganancia 3077
Mes 11 Cantidad de ventas 1 Ganancia 0
Mes 9 Cantidad de ventas 1 Ganancia 0
Mes 10 Cantidad de ventas 0 Ganancia 0

Ventas por mes ordenados por ganancias promedio

Mes Ventas Ganancias Promedio
-----
4 75 191066 2547.55
1 53 117738 2221.47
3 51 162931 3194.73
2 41 107270 2616.34
5 36 91936 2553.78
7 11 26949 2449.91
6 11 36949 3359
```

Estos decrementos son un foco rojo en la empresa, pues al no contar con ventas, está comprometiendo de manera riesgosa su liquidez y sostenibilidad.

## Solución al problema

Si bien se estuvieron dando soluciones a través de todo el análisis de los resultados obtenidos, se resumen a continuación:

1. las categorías que genera la mayor cantidad de ventas en la tienda son 3: Discos duros, procesadores, tarjetas madre, por lo que es importante que se les impulse en su promoción y compra, pues cuenta con una gran cantidad de búsquedas, se pueden usar estrategias de publicidad para promover su compra. Como promociones, ofertas especiales por tiempo limitado, descuentos al referir a más personas, etc.
2. Muchas personas buscan productos en la empresa, pero no concluyen con la compra, por lo que se debe revisar la estrategia de comunicación, de tracción de clientes y de atención a clientes. Se puede pedir que dejen su contacto para recibir promociones especiales de los productos que buscan o relacionados y así fidelizarlos, motivar a la compra y conseguir nuevos clientes. Se pudiera pensar que los clientes, al buscar por el producto que requieren, revisan numerosas opciones y están encontrando en la competencia ofertas que hacen que opten por ellas, por lo cual puede ser beneficioso que se implementen estrategias de benchmarking para retener de manera exitosa a una mayor cantidad de clientes potenciales.
3. Los productos que generan menos ventas son de la categoría de Bocinas y audífonos, además, muchos productos no cuentan con ningún tipo de búsqueda, por lo que se tienen las siguientes hipótesis: El público meta no los conoce o sus características no son de interés para el público. Para esclarecer estos puntos, es necesario que la empresa realice un estudio de mercado y tenga bien estipulado el perfil de su *customer* persona, pues de esta manera, se podrá realizar la estrategia de marketing óptima, con objetivos alcanzables y mayor probabilidad de éxito.
4. Una gran cantidad de artículos no son conocidos, pues no cuentan con búsquedas, ni compras, por lo tanto, tampoco con evaluación. Se necesita decidir si se van a continuar ofertando estos productos o se van a abandonar, pues al mantenerlos en stock, se incurre en egresos importantes para la empresa. Unos productos tienen ventas escasas, devoluciones alta y bajas calificaciones. Es necesario retirar estos productos del mercado para no dañar la reputación de la calidad de productos ofertados por la empresa, pues puede conllevar a una baja importante de ventas y pérdida de clientes.

5. Trabajar en el sistema CRM de la empresa, pues los clientes que han adquirido, evaluado y devuelto estos productos son las personas a las que se les debe ofrecer alternativas que puedan remendar la imagen que tiene la empresa ante ellas. Así como dar seguimiento a quienes están satisfechos con su compra, para fidelizar y continuar con su consumo.
6. La empresa maneja mucho inventario fantasma que cuenta con bajas probabilidades de compra debido a desconocimiento de su oferta, se debe elegir entre retirar del mercado o invertir en propaganda de estos productos, pues para el mercado serían nuevas ofertas.
7. Respecto a ganancias mensuales, se puede notar un primer decremento importante a partir del segundo semestre del año, después un segundo decremento importante a partir del mes 8, que se mantiene durante todo el trimestre siguiente.

Para la reactivación de las ventas se propone la publicación de ofertas especiales de productos que llevan tiempo en stock, mayor inversión en marketing digital, inclusión de clientes para aumentar público meta al ofrecerles beneficios por invitar a más personas a consumir en LifeStore.

8. Como recomendación, también se considera una estrategia beneficiosa la implementación de manejo de inventarios JIT (Just In Time), en la cual se requiere tener un proceso de logística eficiente, pues se abastece el producto hasta el momento en que el cliente lo solicita, de esta forma no se incurre en gastos de almacenamiento y no hay producto rezagado. Se pueden integrar a este modelo aquellos productos que no cuentan con búsquedas ni ventas actualmente.

## Conclusiones

El proyecto sirvió de gran manera para comprobar el entendimiento de todos los elementos revisados a lo largo de la primera parte del curso, además fue el acercamiento perfecto al tema de ciencia de datos para la toma de decisiones. Mediante el manejo, manipulación, presentación y análisis de los datos presentados, se puede analizar de manera profunda la situación presentada para ofrecer las soluciones óptimas.

La situación de LifeStore es un gran ejemplo de manejo de registros e implementación de programas.

Se cumplieron los objetivos de manera satisfactoria.

## Anexos

- Link de proyecto de replit.com
  - <https://replit.com/join/cusgpgpxow-lizeth1998x>
- Link de repositorio en GitHub
  - <https://github.com/lizeth3098/PROYECTO-01-RAMOSREYES-LIZETHMONTSERRAT->