# 2021

# Análisis de rotación de productos LifeStore



Mónica Yazmín García Barrón

## Índice

Introducción	2
Definición del código	
Solución al problema	
Conclusión	10
Anexos	

#### Introducción

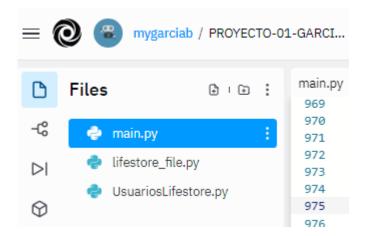
I 2020 fue un año que quedará grabado en la historia por la cantidad de cambios que se realizaron a partir de él: la pandemia que aún a la fecha nos afecta no solo atentó contra la vida, sino que también afectó de forma directa cada una de las actividades que realizamos. Los sectores económico y comercial no son la excepción, y aunque en algunos lugares inició el COVID-19, como su nombre lo indica, en el 2019, la afectación más grande, hasta el momento está registrada en el 2020. Constantemente en las noticas se escucha que debemos adaptarnos a la nueva normalidad, en todos los aspectos, sector educativo, religioso, deportivo, comercial, etc., pero el proceso no es nada fácil.

El presente documento es un análisis de la rotación de productos de **LifeStore**, tienda virtual que maneja una amplia gama de productos de cómputo y tecnología, surge a partir de la problemática reportada por la Gerencia de Ventas ya que existe una importante acumulación de inventario y se ha detectado una reducción en las búsquedas de un grupo importante de productos, lo que ha redundado en una disminución sustancial de sus ventas.

El análisis de la rotación de productos aborda principalmente los siguientes elementos: productos más vendidos y productos rezagados, productos por mejor y peor reseña en el servicio, total de ingresos y ventas promedio mensuales, así como el total anual y meses con más ventas al año.

### Definición del código

a programación del sistema es a través de Python, se puede acceder mediante el enlace que se encuentra en la sección de anexos. Para el correcto funcionamiento del sistema se utilizan dos bases de datos: "lifestore\_file.py" el cual incluye la información proporcionada por la Gerencia de Ventas de LifeStore y "UsuariosLifestore.py" donde se encuentran los cuatro usuarios-administradores registrados al momento para utilizar el sistema.



Al ingresar encontrará una pantalla para el login de usuario, como la que se muestra a la derecha de este texto.

Deberá ingresar correctamente el usuario y la contraseña para poder acceder. Al tercer intento incorrecto, el sistema le impedirá continuar intentando el acceso.



Después de ingresar correctamente los datos, se encuentra el *Menú Principal*, donde se tiene acceso a las siguientes opciones:

- A. Productos más vendidos y productos rezagados.
- B. Productos por reseña en el servicio.
- C. Total de ingresos y ventas promedio.
- D. Salir del sistema.

Debe seleccionar una opción para continuar.

Al seleccionar "Productos más vendidos y productos rezagados" se tienen las opciones que se muestran en la imagen.



Al seleccionar desde el menú principal la opción de "Productos por reseña en el servicio" se tienen las opciones que se muestran en la imagen.



La opción "C" del *Menú Principal*, nos llevará a la pantalla del "*Total de ingresos y ventas promedio*".



Finalmente, pasa salir del sistema, puede seleccionar la opción "D" desde el *Menú Principal*.

Salir del sistema.....(0)

Seleccione una opción del menú (A / B / C / D): D

Saliendo del sistema.

El sistema permite el acceso a través de los usuarios y contraseñas registradas en el archivo "UsuariosLifestore.py". Se permiten hasta tres intentos, que, al obtener éxito, dan acceso al sistema. En el código se lleva a cabo el proceso mencionado mediante un bucle while, que junto con el bucle for y las sentencias if -else, break y continue, realizan el control de seguridad, utilizando las variables "username" y "password" para nombre de usuario y contraseña respectivamente.

```
25
    username = input("Usuario: ")
26
   password = input("Contraseña: ")
27
28 usuario_registrado = 0 #Variable para indicar si el usuario y password se encuentran en la base de datos.
   intentos = 1 #Variable para contar los intentos de ingresar al sistema.
29
31 while usuario registrado != 1 and intentos <= 2: #Bucle while para login de usuario, permite 3 intentos.
32
    for administrador in administradores:
      if username == administrador[0] and password == administrador[1]:
33
      usuario_registrado = 1
34
35
    if usuario_registrado == 0: #Sentencia para cuando los datos ingresados del usuario son incorrectos de
36
     acuerdo a la base de datos.
37
      print("El usuario y/o contraseña son incorrectos.")
      username = input("Usuario: ")
38
     password = input("Contraseña: ")
intentos += 1
39
40
41
   if usuario_registrado == 1: #Sentencia para cuando los datos ingresados del usuario son correctos de acuerdo
42
    a la base de datos.
   43
44
   -----
45
46
   ====== B I E N V E N I D O ======
47
48
   49
50
   _____
   51
```

```
974

975 ☐ else: #Sentencia para cuando el usuario ingresa incorrectos tres veces los datos del login.

976 ☐ | print("""Los datos de usuario son incorrectos.

977 Acceso Denegado.""")
```

Los menús utilizados en el sistema requieren de las variables "opción\_del\_menu\_nombre" y "mostrar\_menu\_nombre", como podemos observar en las imágenes que aparecen a continuación. Se manejan bucles while, sentencias if, elif, así como las sentencias break y continue para el control de los bucles.

```
395
       opcion_del_menu_principal = "A"
 396
      mostrar_menu_principal = "si"
 397
 398 ₪ while mostrar_menu_principal == "si": #Bucle while del menú principal.
 399 print("""------
 400
 401 ====== M E N U P R I N C I P A L ======
 404
 405
      _____
 406
     Productos más vendidos y productos rezagados.....(A)
 407
      Productos por reseña en el servicio.....(B)
 408 Total de ingresos y ventas promedio.....(C)
      Salir del sistema.....(D)""")
 410 opcion_del_menu_principal = input("""
 411 Seleccione una opción del menú (A / B / C / D): """)
       if opcion_del_menu_principal == "A": #Sentencia para productos más vendidos y productos rezagados.
 412 🖯
           opcion_del_menu_mas_vendidos_y_rezagados = "a"
 413
 414
           mostrar_menu_mas_vendidos_y_rezagados = "si"
 415
           while mostrar_menu_mas_vendidos_y_rezagados == "si": #Bucle while del menú más vendidos y rezagados.
 416 🗏
            print("""
 417
 418
          M E N U
 419
        PRODUCTOS MÁS VENDIDOS
 420
      Y PRODUCTOS MAS VENDIDOS
 421
 422
 423
      <<<<<<< \l><</li></l></l></l></l></l></l
 426 Listado de productos con mayores ventas.....(a)
     Listado de productos con mayores búsquedas.....(b)
 427
 428 Listado por categoría con menores ventas ......(c)
 429
     Listado por categoría con menores búsquedas.....(d)
430 Regresar al menú principal.....(e)""")
        opcion_del_menu_mas_vendidos_y_rezagados = input("Seleccione una opción del menú (a / b / c / d / e): ")
```

Al inicio del código se encuentran varias sentencias de iteración y/u ordenación, lo que permite acceder a ellas desde diferentes menús, con la finalidad de disminuir la cantidad de código. Para las iteraciones se utilizan bucles for, así como índices, slicing, operadores relacionales, operadores booleanos, operadores de comparación, operadores de asignación y operadores lógicos entre otros.

```
380
       #Búsquedas anuales
381
       contador_searches = 0
       busquedas = []
383
       longitud_products = len(lifestore_products)
       longitud_searches = len(lifestore_searches)
384
for search in lifestore_searches:
386 ⊟
         if product[0] == search[1]:
387 □
388
           contador_searches +=1
         elif contador_searches !=0:
389 ⊟
           busquedas.append([product[1], contador_searches])
contador_searches = 0
390
391
392 ☐ if product[0] == longitud_products and contador_searches != 0:
393
       busquedas.append([product[1],contador_searches])
394
        #Ventas Anuales sin devoluciones
   64
          #sales = [id_sale, id_product, score (from 1 to 5), date, refund (1 for true or 0 to false)]
         sales = []
   65
         for sale in lifestore_sales:
   66
          if sale[3][6:10] == "2020" and sale[4] == 0:
    sales.append(sale)
   67
   68
   69
   70
```

Para presentar la información al usuario, se utiliza el bucle for, la sentencia if y los índices, para mostrar con la función print los datos.

Se hace uso de la función lambda para ordenar algunas de las listas.

```
654
655
     PRODUCTOS CON MAYORES BUSQUEDAS
656
657
658
659
     660
662
                busquedas.sort(key=lambda x:x[1], reverse=True)
663
                  contador_mb = 0
664
                  for busqueda in busquedas:
665 E
                   if contador_mb < 20:
666 E
                      contador_mb_imprimir = contador_mb + 1
print(f"{contador_mb_imprimir}.- El producto: {busqueda[0]}, tiene: {busqueda[1]} búsquedas. ")
667
668
                   contador_mb +=1
669
670
```

## Solución al problema

sando de base la información proporcionada por la Gerencia de Ventas de LifeStore, y considerando la información obtenida a través del sistema, se ponen a su disposición las sugerencias que se encuentran a continuación.

Retirar del mercado los productos que presentan las peores reseñas, ya que, al ser una tienda virtual, los clientes potenciales tienen acceso directo a esa información ya sea a través de la misma tienda virtual, o a través de redes sociales donde se hacen recomendaciones de tiendas que ofrecen el mismo giro de productos.

Los productos con devolución pueden ofrecerse a un costo más bajo analizando previamente su estado actual o incluso revisando las garantías tal vez se puedan recuperar con los proveedores.

Considerar la reducción de acumulación de inventario, esto puede implementarse con la ayuda de alguna campaña de Marketing Digital, en donde se ofrezcan algunas promociones, descuentos o algún sistema de manejo de puntos por compras para el usuario.

Realizar una campaña de ofertas o promociones, principalmente con los artículos que se encuentran en grandes cantidades en el stock, para tener un flujo de activos.

Al adquirir productos con los proveedores, prestar especial atención a los que tienen mayores y menores búsquedas, así como los que presentas mayores ventas.

#### Conclusión

a situación de las ventas en LifeStore en los últimos meses del año 2020 es preocupante, por lo que es necesario tomar medidas urgentes para estabilizar la tienda y obtener utilidades.

Se podrían sugerir otras estrategias para la tienda virtual, pero la información proporcionada es insuficiente, podríamos implementar un módelo matemático (criterios de primera y segunda derivadas, límites, razón de cambio promedio, medidas de tendencia central y medidas de dispersión) con el que se obtenga un punto de equilibrio, análisis de oferta y demanda, establecer un colchón de seguridad y principalmente encontrar el punto de costos mínimos, así como de ingresos y utilidad máxima. Se presentará formalmente una propuesta a la Gerencia para obtener la información requerida.

Con la información proporcionada, solo podemos aplicar la media aritmética, la cuál no es altamente confiable para sugerir una estrategia a diferencia de las medidas de variación.

Finalmente es importante recordar, que el 2020, fue un año atípico, que aunque la tienda es virtual, también vio afectados sus ingresos en consecuencia de la pandemia en la que aún nos encontramos.

#### Anexos

Enlace al sistema en Replit:

https://replit.com/@mygarciab/PROYECTO-01-GARCIA-BARRON-MONICA-YAZMIN#main.py

Enlace al proyecto en GitHub:

https://github.com/mygarciab/REPORTE-01-GARCIA-BARRON-MONICA-YAZMIN