LIFESTORE

PROYECTO 1

ÍNDICE

INTRODUCCIÓN.

En el presente trabajo se presentan los resultados de las ventas, búsquedas y reseñas de los productos de la empresa LIFESTORE durante el presente año, los cuales se procesaron y analizaron para brindar recomendaciones sobre el manejo y stock de estos para posterior planificación.

En la primera parte se explica el código que se implementó para manejar y procesar los datos compartidos, donde se pone en práctica lo aprendido de programación en Python para análisis y clasificación de datos mediante la creación de programas de entrada de usuario, validaciones, definición de variables, manejo de listas, operadores lógicos y condicionales.

En la segunda parte se realiza el análisis económico empresarial para mejorar las prácticas sobre el stock y por último se presentan las conclusiones.

CÓDIGO Y SOLUCIÓN AL PROBLEMA.

```
Administradores = [['Norma Silva','EMTECH'],['Javier
Ramirez','EMTECH']]

Usuario = input ('Ingresa tu usuario :')
Contraseña = input ('Ingresa tu contraseña: ')

Administrador = 0
Intentos = 0

while Administrador != 1 and Intentos < 2:
  for Admin in Administradores:
    if Admin[0] == Usuario and Admin[1] == Contraseña:
      Administrador = 1

  if Administrador == 0:
    print('Usuario o contraseña incorrectos')
    Usuario = input ('Ingresa nuevamente tu usuario :')
    Contraseña = input ('Ingresa nuevamente tu
    contraseña: ')
    Intentos +=1

if Administrador ==1:
  Saludo = 'Buen día  '
  Unir = Saludo + Usuario
  print(Unir)
  Opción = 0
```

Generación de Usuario y Contraseña.

En la primera parte del código se recurrió a definir a los usuarios y la contraseña a través de cadenas. Se esperaba que el usuario del programa pudiera ingresar sus datos por lo que se agregaron inputs, si estos son incorrectos el usuario tiene tres oportunidades para ingresar nuevamente y no salir inmediatamente del programa; por lo que se tuvo que definir una condición esperada de administrador = 0 y un número de intentos menor a 2 a través de operadores logísticos y relacionales. Además, fue posible agregar más de un usuario gracias a la utilización de índices y slicing en la lista de administradores. Una vez completados correctamente los datos el programa procede a saludar a los usuarios de manera personalizada.

```
Administrador = 1

if Administrador == 1:
    print('Selecciona una opción:\n 1. 10 Productos con
    mayores ventas\n 2. 10 Productos con mayores
    búsquedas\n 3. 10 Productos con menores ventas\n 4.
    10 Productos con menores búsquedas\n 5. 10
    Productos con mejores reseñas\n 6. 10 Productos con
    peores reseñas')
    Opción = int(input('Opción: '))

if Opción == 1:
     print('10 Productos con mayores ventas')

elif Opción == 2:
        print('10 Productos con mayores búsquedas')

elif Opción == 3:
        print('10 Productos con menores ventas')

elif Opción == 4:
        print('10 Productos con menores búsquedas')

elif Opción == 5:
        print('10 Productos con mejores reseñas')

elif Opción == 6:
        print('10 Productos con mejores reseñas')

else:
        print('Selecciona un número entre 1 y 6')
```

Menú de Opciones.

Este no es más que un control de flujo, por lo que para realizarlo se utilizaron sentencias if, elif y else que nos permite controlar y alterar el flujo del programa, en este caso para que los usuarios pudieran visualizar qué tipo de información necesitaban sin la necesidad de ver todo el programa, también en caso de seleccionar una opción no disponible, el programa avisa que tiene opciones limitadas gracias a else.

Selección de listas y orden.

```
contador = 0
Ventas_Totales = []

for producto in lifestore_products:
  for venta in lifestore_sales:
    if producto[0] == venta[1]:
      contador += 1 #va sumando cada uno
  formato = [producto[0],producto[1],contador]
  Ventas_Totales.append(formato)
  contador = 0 #resetea el contador
  #print(Ventas_Totales)

Mayores_Ventas = []

while Ventas_Totales:
  maxim = Ventas_Totales[0][2]
  lista_max = Ventas_Totales[0]
  for Venta_Total in Ventas_Totales:
    if Venta_Total[2] > maxim:
      maxim = Venta_Total[2]
      lista_max = Venta_Total
  Mayores_Ventas.append(lista_max)
  Ventas_Totales.remove(lista_max)
  #print(Mayores_Ventas)

if Opción == 1:
    for conteo in range(0,10):
     print('El producto: \n',Mayores_Ventas[conteo][1],
     "\n",'se vendió',Mayores_Ventas[conteo][2],'veces')
```

El principal problema al que nos enfrentamos es que queríamos índices específicos de listas diferentes para poder relacionar los productos con ventas y con diferentes datos por lo que nuevamente se realiza slicin de listas en listas, agregar un formato que permitiera mirar en la consola los datos de forma más estética u ordenada y principalmente trabajar con bucles for que nos permitían iterar sobre las listas y también ordenarlas ya que es el requerimiento más importante del proyecto, ya que de esto depende el análisis.

Una vez más me apoyo en la sentencia if para poder llevar la información que el usuario seleccionó desde el menú; también se aplica un bucle for ya que a través de éste puedo utilizar la sentencia range que permite repetir una acción cierto número de veces, en este caso 10 veces ya que son los productos que nos interesan de cada categoría. Para imprimir y que los resultados se vieran más estéticos se procedió a llamar las listas, espaciar con \n y nuevamente utilizar índices ya que no podía concatenar por la naturaleza de los datos.

Este procedimiento se repitió en cada opción disponible del menú, solo se hacía variación de listas e índices y se cambiaba el orden de los productos.

El código se encuentra disponible para su consulta en: https://replit.com/@EstefaniaSilva/PROYECTO-1#main.py

CONCLUSIONES.

Para la estrategia de negocio se recomienda a LifeStore reducir la adquisición de productos con menores ventas[1] y de esta manera evitar la sobre acumulación de inventario al tiempo que mantenemos en almacén los productos con mayores ventas.

Por otro lado, para aumentar la búsqueda de productos que se han visto disminuídos, se podría empujar una estrategia de publicidad para pautar y atraer nuevos clientes al sitio web.

[1] Opción 3 del menú del Proyecto 1.