



## **Proyecto 1. Introducción a Python**

**Caso: Análisis de Datos de la tienda virtual LifeStore**

**Lizeth Moro López**

**6 de septiembre de 2020**

## Contenido

<b>Introducción .....</b>	<b>3</b>
<b>Definición del código .....</b>	<b>4</b>
<b>Solución al problema .....</b>	<b>8</b>
<b>Conclusión .....</b>	<b>9</b>

## Introducción

En este caso de estudio se analizó la base de datos de la tienda virtual LifeStore, la cual se dedica a vender productos de informática como procesadores, audífonos, bocinas, tarjetas madre, tarjetas de video, etc. Esta empresa tiene una importante acumulación de inventario de productos de informática y ha detectado una reducción en las búsquedas de algunos productos, lo que ha provocado una reducción sustancial de sus ventas del último trimestre.

Debido a esta situación, el gerente de ventas de la tienda me solicitó que realizaré un análisis de su base datos, para poder identificar los siguientes elementos:

- 1.- Productos más vendidos y menos vendidos/rezagados
- 2.- Productos con mayores búsquedas y los de menores.
- 3.- Productos con las mejores reseñas y los de las peores.
- 4.- Productos frecuentemente devueltos.

Al mismo tiempo, se busca obtener el total de ingresos y ventas promedio mensuales, total anual y meses con más ventas al año.

Todo esto ayudará al gerente a tomar mejores decisiones estratégicas para optimizar el manejo de su inventario y buscar soluciones para las áreas deficientes de su empresa.

Con el conocimiento de esta información, la empresa logrará potencializar su competitividad con respecto a las amenazas externas del entorno como lo son sus demás contrincantes. Podrá utilizar los datos para determinar los hábitos de sus consumidores y poder ofrecerles realmente lo que ellos desean.

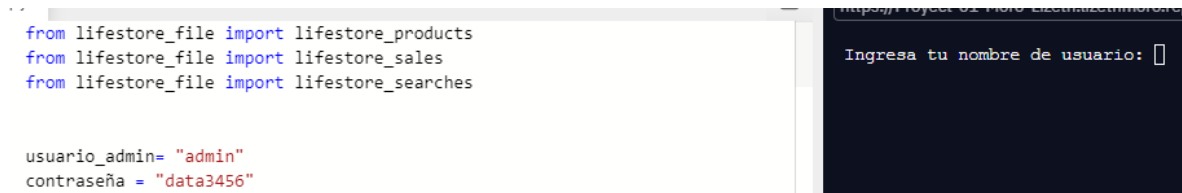
Esto representa una gran ventaja competitiva dentro de un mercado saturado de empresas que continúan tomando decisiones con base a la intuición y en lo que les ha funcionado hasta ahora. En este mundo tan acelerado y cambiante, hay que recurrir a la tecnología y a la información para innovar en cualquier negocio buscando siempre el beneficio mutuo entre la empresa y el cliente.

## Definición del código

Enlace para acceder al código: <https://github.com/lizmorero/PROYECTO-01-MORO-LIZETH>

A continuación, se presenta cada uno de los procesos ejecutados en el código.

El programa inicialmente solicita el nombre de usuario y la contraseña. Para acceder es necesario ingresar el usuario= admin y la contraseña = data3456. Este el único usuario que puede acceder a la base de datos de la tienda.

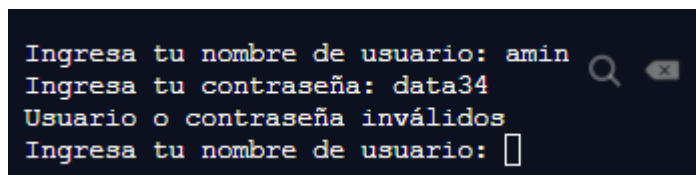


```
from lifestore_file import lifestore_products
from lifestore_file import lifestore_sales
from lifestore_file import lifestore_searches

usuario_admin= "admin"
contraseña = "data3456"
```

Ingresa tu nombre de usuario:

Si la contraseña o el usuario son incorrectos, se solicitará ingresar los datos nuevamente.



```
Ingresa tu nombre de usuario: amin
Ingresa tu contraseña: data34
Usuario o contraseña inválidos
Ingresa tu nombre de usuario: 
```

En el menú de opciones el programa solicita, elegir una de las siguientes opciones: 10 productos más vendidos, 20 productos más buscados, 54 productos sin ventas/rezagados, 40 productos sin búsquedas, 20 productos mejor calificados, 20 productos con mayor devolución y los ingresos anuales, mensuales y por producto.

Al elegir una opción, aparece un mensaje de texto describiendo que opción se eligió.

```
main.py
11
12 while usuario_contrasena != contrasena or usuario_entrada != usuario_admin:
13     print('Usuario o contraseña inválidos')
14     usuario_entrada = input('Ingresa tu nombre de usuario: ')
15     usuario_contrasena = input('Ingresa tu contraseña: ')
16 else:
17     print('¡Bienvenido a lifestore')
18     Elige una opción:
19     a. 10 Productos más vendidos
20     b. 20 Productos más buscados
21     c. 54 Productos sin ventas/rezagados
22     d. 40 Productos sin búsquedas
23     e. 20 Productos mejor calificados
24     f. 20 Productos con mayor devolución
25     g. Ingresos por producto
26     h. Ingresos totales anuales
27     i. Ventas promedio mensuales'''
28
29 opcion_elegida = input('Ingresa la letra de la opción elegida: '\n' )
30
31 if opcion_elegida == "a":
32     print("Seleccionaste: lista de los 10 productos más vendidos")
33 elif opcion_elegida == "b":
34     print("Seleccionaste: lista de los 20 productos más buscados")
35 elif opcion_elegida == "c":
36     print("Seleccionaste: lista de los 54 productos sin ventas/rezagados")
37 elif opcion_elegida == "d":
38     print("Seleccionaste: lista de los 40 productos sin búsquedas")
39 elif opcion_elegida == "e":
40     print("Seleccionaste: lista de los 20 productos mejor calificados")
41 elif opcion_elegida == "f":
42     print("Seleccionaste: lista de los 20 productos con mayor devolución")
43 elif opcion_elegida == "g":
44     print("Seleccionaste: Ingresos por producto")
45 elif opcion_elegida == "h":
46     print("Seleccionaste: Ingresos totales anuales")
47 elif opcion_elegida == "i":
48     print("Seleccionaste: Ventas promedio mensuales")
49 else:
50     print("Opción no válida. Por favor, elige una opción válida.")
```

Hasta este punto, las herramientas utilizadas fueron: el bucle while, for, comparaciones e input para que el usuario ingresara sus datos.

Para determinar los productos más vendidos y los rezagados, utilice un contador junto con un bucle for dentro de otro for porque necesitaba extraer datos de dos listas diferentes. Posteriormente, ordene las ventas con respecto al id del producto.

```

contador = 0
ventasxproducto = []

for producto in lifestore_products:
    for venta in lifestore_sales:
        if producto[0]==venta [1]:
            contador += 1

formato = [producto[1],contador]
ventasxproducto.append(formato)
contador =0

# ventasxproducto = [[producto, ventas]]
ventas_ordenadas = []
while ventasxproducto:
    minimo = ventasxproducto [0][1]
    lista_actual = ventasxproducto [0]
    for venta in ventasxproducto:
        if venta [1]> minimo:
            minimo = venta [1]
            #lista_actual = ventasxproducto
            lista_actual = venta #Asignamos lista actual
    ventas_ordenadas.append(lista_actual)
    ventasxproducto.remove ([lista_actual])

if opcion_elegida == "a":
    for indice in range(0,10):
        print("Nombre del producto:","\n",ventas_ordenadas[indice][0],"\n","unidades vendidas:",
        ventas_ordenadas[indice][1], "\n")

if opcion_elegida == "c":
    for indice in range(42,96):
        print("Nombre del producto:","\n",ventas_ordenadas[indice][0],"\n","unidades vendidas:",
        ventas_ordenadas[indice][1], "\n")

```

Para recabar los productos que tenían más búsquedas y los que no tenían también utilice un contador que iba sumando cada que el id se repetía dentro de la lista `lifestore_products` y en `lifestore_searches` y también las ordene de mayor a menor.

```

contador = 0
busqueda_productos = []
for producto in lifestore_products:
    for busqueda in lifestore_searches:
        if producto[0]==busqueda [1]:
            contador += 1

formato = [producto[1],contador]
busqueda_productos.append(formato)
contador =0

busquedas_ordenadas = []
while busqueda_productos:
    minimo = busqueda_productos [0][1]
    lista_actual = busqueda_productos [0]
    for busqueda in busqueda_productos:
        if busqueda [1]> minimo:
            minimo = busqueda [1]
            lista_actual = busqueda #Asignamos lista actual
    busquedas_ordenadas.append(lista_actual)
    busqueda_productos.remove(lista_actual)

#print (busquedas_ordenadas)

if opcion_elegida == "b":
    for indice in range(0,20):
        print("Nombre del producto:", "\n", busquedas_ordenadas [indice][0], "\n", "búsquedas:",
            busquedas_ordenadas[indice][1], "\n")

if opcion_elegida == "d":
    for indice in range(56,96):
        print("Nombre del producto:", "\n", busquedas_ordenadas[indice][0], "\n", " búsquedas:",
            busquedas_ordenadas[indice][1], "\n")

```

Los productos mejor calificados también los obtuve utilizando un contador y una variable llama suma\_score que se igualaba a cero, esta variable almacenada la suma de las calificaciones por cada producto, el contador se utilizó para dividir la suma y así, obtener una media de las calificaciones.

```

contador = 0
suma_score = 0
productos_promedio = []
for producto in lifestore_products:
    for venta in lifestore_sales:
        if producto[0] == venta[1]:
            contador += 1
            suma_score += venta[2]
    promedio = suma_score // contador
    final = [producto[1], promedio]
    productos_promedio.append(final)

promedios_ordenados = []
while productos_promedio:
    minimo = productos_promedio [0][1]
    lista_actual = productos_promedio [0]
    for promedi in productos_promedio:
        if promedi [1]> minimo:
            minimo = promedi [1]
            lista_actual = promedi #Asignamos lista actual
    promedios_ordenados.append(lista_actual)
    productos_promedio.remove(lista_actual)

if opcion_elegida == "e":
    for indice in range(0,20):
        print("Nombre del producto:", "\n", promedios_ordenados [indice][0], "\n", "score:",
            promedios_ordenados [indice][1], "\n")

```

Los productos devueltos se obtuvieron de igual manera con el bucle for y un contador que sumaba cuando aparecía en las listas aparecía el número 1 que determinaba que el producto había sido devuelto.

```
contador = 0
devoluciones_producto = []
for producto in lifestore_products:
    for venta in lifestore_sales:
        if producto[0] == venta[1]:
            if venta[4] == 0:
                contador += 1
    formato = [producto[1], contador]
    devoluciones_producto.append(formato)
    contador = 0
#print(devoluciones_producto)

devoluciones = []
while devoluciones_producto:
    minimo = devoluciones_producto[0][1]
    lista_actual = devoluciones_producto[0]
    for devolucion in devoluciones_producto:
        if devolucion[1] > minimo:
            minimo = devolucion[1]
            lista_actual = devolucion #Asignamos lista actual
    devoluciones.append(lista_actual)
    devoluciones_producto.remove(lista_actual)

if opcion_elegida == "f":
    for indice in range(0,20):
        print("Nombre del producto:", "\n", devoluciones[indice][0], "\n", "devoluciones:", devoluciones[indice][1], "\n")
```

Para obtener las ventas por producto se utilizó el mismo procedimiento, pero en este caso se multiplicaron las ventas por el precio de venta. Al sumar todos los ingresos por producto se obtuvo los ingresos anuales, de los cuales solo 8 meses se tiene registro de ventas. Por último, para los ingresos mensuales promedio se realizó la división de los ingresos anuales entre los meses que se registraron ventas.

```

contador = 0
ventas_totales = []
for producto in lifestore_products:
    for venta in lifestore_sales:
        if producto[0] == venta[1]:
            contador += 1
    total = producto [2] * contador
    final = [producto[0], producto[1],contador, total]
    ventas_totales.append(final)

sumo = 0
total =[]
for ven in ventas_totales:
    sumo+=ven[3]

    total.append(sumo)

suma= 0
for elemento in total:
    suma= suma + elemento
#print (suma)
if opcion_elegida == "g":
    for indice in range(0,96):
        print("Nombre del producto:", "\n",ventas_totales[indice][1], "\n", "ingresos:", ventas_totales
[indice][3], "\n")

if opcion_elegida == "h":
    print("Ingresos generados:")
    print (suma)

if opcion_elegida == "i":
    print ("Ventas promedio mensuales:")
    print (suma//8)

```

## Solución al problema

Con los resultados obtenidos se plantea al gerente de la tienda las siguientes sugerencias:

1.- Utilizar promociones de venta para tener una rotación de inventario más rápida y evitar tener exceso de inventario con productos rezagados. Este inventario



acumulado es dinero detenido que se traduce en pérdidas para la empresa. Debe considerar poner en oferta esos productos para recuperar al menos su inversión.

3.- De los productos con más búsquedas en su página web, debe considerar siempre tener suficientes piezas en su inventario de seguridad, para evitar perder una venta si el producto no está en existencia.

4.- En cuanto a la calificación de los productos, todos tienen un puntaje mayor a 4 por lo que la percepción de los consumidores con respecto a la calidad de los artículos que se ofertan es bastante buena. Hay que detectar la razón de porque los artículos son devueltos cuando aparentemente son del agrado del cliente mediante encuestas post-venta que podría ayudar al gerente a recabar información cualitativa de sus clientes.

5.- Definir una estrategia de ventas y de promoción donde al navegar en el sitio web se muestren como sugerencias aquellos productos que nunca se han vendido, puede incluirse información de su utilidad y funcionamiento para captar la atención de los clientes.

6.- Ofrecer un 10% de descuento al suscribirse a la página web, y de esta manera enviar información de la tienda a los correos de los clientes, como descuentos, días especiales de ventas, etc.

7.- En cuanto a los ingresos percibidos, los productos que mayores ingresos generan a la tienda son los audífonos por lo que debería incluir en su portafolio de productos artículos similares a éste o complementarios para aumentar sus ventas.

## **Conclusión**

El análisis de este caso me permitió poner en práctica cada uno de mis conocimientos adquiridos durante esta primera parte del curso de fundamentos de Python. Utilizando cada una de las herramientas cómo es la información histórica, la

programación y la analítica de datos es posible ayudar a las empresas a ser más competitivas, a que tomen mejores decisiones estratégicas y menos operativas.

Las empresas deben de aprovechar la información que obtienen de sus clientes porque esto les ayuda a darse cuenta de las oportunidades que tienen, maximizar sus fortalezas, poner énfasis en sus debilidades y atacar sus amenazas.

Este caso práctico es comúnmente visto en la vida real en la que los gerentes dirigen sus empresas en base a lo que ellos creen qué es correcto o en lo que a ellos les ha funcionado anteriormente.

Este recurso tan valioso que es la información va a marcar la diferencia entre las empresas que van a dirigir el mercado y las que quedarán en el olvido por falta de innovación y por no detectar a tiempo las amenazas y debilidades que tienen dentro de su organización.

Esto solo es un ejemplo de cómo podemos mejorar cada una de las áreas de la empresa siempre y cuando conozcamos lo que está sucediendo.

Python es una herramienta que me permitió utilizar los datos para convertirlos en información, lo cual me resultó una tarea difícil, pero produce grandes ventajas para el gerente cuando esa información la convierte en conocimiento y la aplica para mejorar aquellas áreas deficientes dentro de su empresa y potencializar aquellas fortalezas que tiene con respecto a sus competidores.