

Proyecto 1

Introducción a Python, Caso: Lifestore

Rodrigo Vázquez Espino

```
-----  
Bienvenido a Life Store!  
1. Login  
2. Salir  
  
-----  
Ingresa la opcion que deseas.█
```



```
-----Total de Ingresos-----  
737916  
-----Ventas Promedio Mensuales-----  
avg_sale  
date  
01 1972.513514  
02 2990.944444  
03 3210.951220  
04 2355.250000  
05 3635.315789  
06 3582.125000  
07 2925.923077  
08 1531.000000  
09 2636.857143  
10 1817.888889  
11 2335.666667  
12 3219.000000  
----- 5 Meses con Mayores Ventas-----  
sale_sum  
date  
04 169578  
03 131649  
02 107674  
01 72983  
05 69071  
Presiona Enter para regresar.█
```

Índice.

Índice.	2
Introducción.	3
Lifestore.	3
Instalación.	3
Definición del Código.	4
Definición de Variables.	4
Procesos.	5
mostrar_menu() línea 1483.	5
login() línea 1464.	5
mostar_operaciones() línea 1490.	5
realizar_operacion(op) línea 1557.	6
productos_mas_vendidos() y productos_rezagados() líneas 1498 y 1526.	6
resenias()	8
total_ingresos()	9
Solución del Problema.	10
Conclusión.	11
El caso del producto 54.	11

Introducción.

Lifestore.

Este documento pretende presentar un sistema computacional que se encarga de presentar información correspondiente a la venta de productos de la tienda Lifestore.

Debido a la alta acumulación de datos e información de inventario, la empresa requiere acceder de manera sencilla a reportes mensuales para poder realizar análisis acerca de la venta y rotación de sus productos.

Este sistema utiliza los datos almacenados y proporcionados por la empresa para poder realizar dicho análisis utilizando herramientas que proporciona el lenguaje de programación Python y las librerías Pandas y Numpy.

Instalación.

El sistema fue desarrollado y ejecutado utilizando la versión 3.6.9 de Python.

Adicionalmente utiliza bibliotecas que no se encuentran instaladas por defecto. De no contar con Pandas y Numpy, el sistema se encarga de instalarlas durante su ejecución utilizando [pip](#).

El sistema consiste de un solo archivo *PROYECTO-01-VAZQUEZ-RODRIGO.py* el cual puede ser clonado descargado vía github: <https://github.com/Roppest/Lifestore.git>.

Se puede utilizar el usuario: admin con la clave: admin para ingresar.

Definición del Código.

Definición de Variables.

Comenzando con las variables más sencillas: **opcion**, **op**, **user** y **password**, se utilizan para control de flujo y guardar información ingresada por el usuario. Las cuales serán explicadas a detalle más adelante.

Los datos correspondientes a las ventas, productos y búsquedas están almacenados en variables con los mismos nombres planteados en el archivo original proporcionado de tipo lista de listas, **lifestore_products**, **lifestore_sales** y **lifestore_searches**¹. También la información para ingresar al sistema está almacenada en la lista **user_data**², la cual sigue el formato [usuario, contraseña]. **Se puede utilizar el usuario: admin con la clave: admin para ingresar.**

Las variables **df_sales**, **df_products** y **df_searches**³ son utilizadas para guardar estas tablas y serán manipuladas para obtener el resultado deseado.

¹ Líneas 23, 122 y 408 respectivamente.

² Línea 1477.

³ Primera aparición, línea 1504 en función productos_mas_vendidos().

Procesos.

De acuerdo al orden de ejecución los procesos son los siguientes:

1. mostrar_menu() línea 1483.

Muestra el menú principal el cual solo contiene dos opciones, ingresar o salir.

```
-----  
Bienvenido a Life Store!  
1. Login  
2. Salir  
  
-----  
Ingresa la opcion que deseas.█
```

La opción será ingresada por el usuario mediante el teclado y será utilizada para decidir el siguiente proceso.

2. login() línea 1464.

Pedirá al usuario ingresar sus datos de usuario y contraseña, estos datos son comprobados con la lista de usuarios establecida en el programa y retorna True o False si se dio o negó acceso.

```
Ingresa tu usuario: admin  
Ingresa tu clave. admin  
Ingresando...  
█
```

Se puede utilizar el usuario: admin con la clave: admin para ingresar como se muestra en la imagen.

3. mostar_operaciones() línea 1490.

El valor obtenido en login será utilizado por una condicional para continuar y

```
-----Operaciones-----  
1. Productos mas vendidos y productos rezagados.  
2. Productos por resenia de servicio  
3. Total de ingresos y ventas promedio mensuales,  
total anual y meses con mas ventas al anio.  
Ingresa la opcion que deseas.█
```

mostrar las operaciones disponibles.

El usuario ingresará el nombre de la operación que desee.

4. realizar_operacion(op) línea 1557.

De acuerdo a la operación ingresada el sistema evaluará mediante condicionales el siguiente proceso. Los procesos a elegir podrán ser uno de los siguientes.

1. **productos_mas_vendidos() o productos_rezagados()** , ambos dentro de un submenú.
2. **resenias()**
3. **total_ingresos()**

5. productos_mas_vendidos() y productos_rezagados() líneas 1498 y 1526.

Estas operaciones aunque son similares, están separadas en dos procesos debido a que la información que se presenta en pantalla suma 300 filas entre los dos.

Esta y las demás operaciones mencionadas anteriormente comparten algunos pasos.

1. Crear DataFrames utilizando las variables que contienen las listas de inventario y ventas.

```
df_sales = pd.DataFrame(lifestore_sales, columns=['id_sale', 'id_product', 'score',  
'date', 'refund' ])
```

```
df_products = pd.DataFrame(lifestore_products, columns=['id_product', 'name',  
'price', 'category', 'stock'])
```

```
df_searches = pd.DataFrame(lifestore_searches, columns = ['id_search',  
'id_product'])
```

2. Filtrar la información necesaria para la operación, esto resulta en tablas con menos columnas y por ende, más rápidas de analizar.

```
df_sales = df_sales[['id_sale', 'id_product']]
```

```
df_products = df_products[['id_product', 'name']]
```

Para obtener los productos más vendidos, se filtran los 50 productos más vendidos y más buscados agrupados por id y se cuentan el número de ventas obtenidas,

después se realiza un cruce entre tablas para ligar los identificadores de producto con su nombre ya que esta información no está en una misma tabla.

```
top_sales = pd.DataFrame(df_sales.groupby('id_product').size().nlargest(50))
```

```
top_sales = top_sales.join(df_products.set_index('id_product'), on='id_product')
```

Para las búsquedas de productos se obtienen 100 filas.

```
top_searches = pd.DataFrame(df_searches.groupby('id_product').size().nlargest(100))
```

```
top_searches = top_searches.join(df_products.set_index('id_product'), on='id_product')
```

El procedimiento para las menores búsquedas y ventas es similar.

```
worst_sales = pd.DataFrame(df_sales.groupby('id_product').size().nsmallest(50))
```

```
least_searched = pd.DataFrame(df_searches.groupby('id_product').size().nsmallest(100))
```

Se puede apreciar que cambia la clasificación de **nlargest** a **nsmallest** al final de los bloques de código.

```
-----50 Mayores Ventas-----
id_product  total_sales  name
54          50  SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm
3           42  Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, S...
5           20  Procesador Intel Core i3-9100F, S-1151, 3.60GH...
42          18  Tarjeta Madre ASRock Micro ATX B450M Steel Leg...
57          15  SSD Adata Ultimate SU800, 256GB, SATA III, 2.5...
29          14  Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GA...
2           13  Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 3...
4           13  Procesador AMD Ryzen 3 3200G con Gráficos Rade...
47          11  SSD XPG SX8200 Pro, 256GB, PCI Express, M.2
48           9  SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2
12           9  Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 ...
```

```
-----50 Menores Ventas-----
id_product  total_sales  name
94           1  HyperX Audifonos Gamer Cloud Flight para PC/PS...
22           1  Tarjeta de Video MSI NVIDIA GeForce GTX 1050 T...
40           1  Tarjeta Madre Gigabyte XL-ATX TRX40 Designare,...
45           1  Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151...
17           1  Tarjeta de Video Gigabyte AMD Radeon R7 370 OC...
13           1  Tarjeta de Video Asus NVIDIA GeForce GTX 1050 ...
46           1  Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2,...
50           1  SSD Crucial MX500, 1TB, SATA III, M.2
10           1  MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PC...
60           1  Kit Memoria RAM Corsair Dominator Platinum DDR...
66           1  TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Wid...
```

En las imágenes de ejecución se muestran resultados parciales debido a su extensión.

6. resenias()

Definir la mejor o peor reseña que puede recibir un producto no resulta trivial en su totalidad, esto podría considerarse de distintas formas, ya que un producto puede recibir una pésima calificación en una venta, pero una calificación excelente en otra. En este caso utiliza la **calificación mínima** que recibió un producto en una venta. También se suma el total de devoluciones que tuvo el producto.

```
resenias = df_sales.groupby('id_product').agg({'score':min,'refund':np.sum})\
.rename(columns={'score':'min_score'},inplace=False)
```

A diferencia de la operación anterior esta utiliza una función de numpy, np.sum para realizar la suma en la columna de devoluciones.

Una vez obtenido el resultado y aplicando un ordenamiento descendente de los valores por calificación se obtienen las 20 primeras y últimas filas.

```
-----Mejores 20 resenias-----
      min_score  refund                                name
id_product
1              5        0  Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Q...
21             5        0  Tarjeta de Video MSI AMD Mech Radeon RX 5500 X...
85             5        0  Logitech Audifonos Gamer G635 7.1, Alámbrico, ...
84             5        0  Logitech Audifonos Gamer G332, Alámbrico, 2 Me...
67             5        0  TV Monitor LED 24TL520S-PU 24, HD, Widescreen,...
66             5        0  TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Wid...
60             5        0  Kit Memoria RAM Corsair Dominator Platinum DDR...
```

```
-----Peores 20 resenias-----
      min_score  refund                                name
id_product
45             1        1  Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151...
17             1        1  Tarjeta de Video Gigabyte AMD Radeon R7 370 OC...
29             1        1  Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GA...
31             1        3  Tarjeta Madre AORUS micro ATX B450 AORUS M (re...
46             2        1  Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2,...
```


7. total_ingresos()

La última operación consiste en agrupar las ventas por mes y sumar el precio de los productos vendidos. Primero se ignoran los productos que fueron retornados ya que representan pérdidas en las ventas.

```
no_refund = df_sales['refund'] == 0
df_sales = df_sales[no_refund]
```

Se realizan cruces de tablas para obtener el precio del producto.

```
df_products = df_products[['id_product','price']]
monthly_sales = df_sales.join(df_products.set_index('id_product'), on='id_product')
```

Para las ventas promedio se calcula el promedio mensual y para el total de ventas se calcula la suma de ventas mensuales. Dichas operaciones serán obtenidas con ayuda de numpy, **np.mean** y **np.sum**.

```
avg_monthly_sales =
monthly_sales.groupby(monthly_sales['date'].dt.strftime('%m')).agg({'price':np.mean})
top_monthly_sales =
monthly_sales.groupby(monthly_sales['date'].dt.strftime('%m')).agg({'price':np.sum}).nlargest(5,columns='price')
```

El resultado es corto y puede apreciarse en su totalidad.

```
-----Total de Ingresos-----
737916
-----Ventas Promedio Mensuales-----
      avg_sale
date
01    1972.513514
02    2990.944444
03    3210.951220
04    2355.250000
05    3635.315789
06    3582.125000
07    2925.923077
08    1531.000000
09    2636.857143
10    1817.888889
11    2335.666667
12    3219.000000
----- 5 Meses con Mayores Ventas-----
      sale_sum
date
04         169578
03         131649
02         107674
01          72983
05          69071
Presiona Enter para regresar.
```

Solución del Problema.

Las empresas que mantienen un registro de sus productos y ventas eventualmente enfrentan problemas como el que fue presentado, para ello es necesario implementar herramientas que puedan ayudar a dar sentido a todos los datos que se almacenan.

El sistema utiliza la clase **DataFrame** implementada en Pandas, para poder modelar las listas de datos como tablas y poder ordenar, sumar y filtrar información como se podría hacer en una hoja de cálculo o en una base de datos.

La información se muestra como texto en pantalla para que el usuario pueda consultar de manera rápida y eficaz estados mensuales acerca de sus productos.

Conclusión.

El almacenamiento de datos es de suma utilidad para las empresas que los almacenan en el momento de tomar decisiones. Siempre que se mantenga un esquema consistente de almacenamiento esta tarea puede ser automatizada y puede ahorrar tiempo al realizar análisis.

Existen varias maneras en que se pueden tomar decisiones a partir de los resultados obtenidos como por ejemplo: ofrecer promociones para los productos rezagados, resaltar los productos más vendidos y mejor calificados; así como encontrar patrones de venta de ciertos productos.

Un patrón curioso que encontré con estos reportes es que existen productos que son buscados con mucha frecuencia pero, comparado con las búsquedas, las ventas son bajas.

El caso del producto 54.

Podemos observar que este producto lidera las búsquedas con un total de 263.

```
-----100 Mayores Búsquedas-----
```

id_product	total_searches	name
54	263	SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm
57	107	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5...
29	60	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GA...
3	55	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, S...

Este producto también lidera las ventas por cantidad de producto con 50 unidades vendidas.

```
-----50 Mayores Ventas-----
```

id_product	total_sales	name
54	50	SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm
3	42	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, S...

Sin embargo, cuando es comparado con otros productos, uno de ellos expuesto a continuación, podemos observar que el índice búsqueda / venta es bajo. Esto quiere decir que cerca de 1/5 de búsquedas resultan en venta para este producto.

Contrastando contra el segundo producto más vendido, el producto 3, cuyas ventas por unidad son de 42, pero sus búsquedas totales son de 55. Esto corresponde a un índice de 1.3, esto quiere decir que casi cada búsqueda de este producto ha resultado en una venta. Los productos no pertenecen a la misma categoría, pero este tipo de descubrimientos pueden despertar una curiosidad por realizar análisis más a profundidad, y eventualmente resultar en una estrategia de venta que pueda favorecer a la empresa.