



LIFE STORE

PROYECTO 01 – INTRODUCCIÓN A PYTHON

EMTECH - SANTANDER

ARANTZA SOFÍA VILICAÑA PRIETO

CONTENIDO

Introducción	3
Descripción de Código.....	3
Log-In	3
Productos más vendidos y rezagados	4
Productos por reseña en servicio	5
Ingresos y ventas	6
Solución al problema.....	7
Conclusión.....	8

Introducción

2020, un año que todos esperábamos con ansias; nuevos inicios, metas, crecimiento... Sin embargo, se ha convertido un recordatorio constante de un mensaje muy claro: *adaptarse o morir*. **Lifestore** no fue la excepción.

Desde sus inicios, esta tienda ha permanecido en manos de una familia interesada en ser los pioneros de proveeduría tecnológica de su comunidad y esto fue así, hasta que el COVID-19 tocó sus puertas amenazando con el desplome de la empresa, siendo que a lo largo de los años habían optado por llevar una administración mucho más “íntima” al únicamente tener una base de datos a pluma y papel.

Una vez más, dos opciones: adaptarse o cerrar sus puertas. Fue así como ***lifestore*** eligió subir de nivel y unirse a la nueva era digital.

En el siguiente documento, se observarán de los primeros pasos para adentrarse a un nuevo mundo.

Descripción del Código

Inicialmente, se hizo una recopilación de la mayor cantidad de datos que se pudieran obtener en los registros de la tienda. Se hicieron 3 principales:

1. Inventario de productos en la tienda
2. Ventas realizadas
3. Solicitudes de búsqueda de productos

LOG IN

Seguidamente, se procedió a limitar quiénes pueden visualizar estos datos.

```
1431 #lista con usuarios de inicio ya definidos
1432 lista_admin = [{"admin", "admin"}, {"user", "123"}]
1433
1434 #menu principal
1435 print("BIENVENIDO a LifeStore \na) Iniciar Sesión \nb) Registrarme")
1436 v_menu_principal = input("Ingrese a continuación la opción elegida: ")
1437
1438 bandera = False
```

Primeramente, se definió la lista en donde serán almacenados los usuarios junto con sus respectivas contraseñas. De la misma manera, se definieron dos usuarios por default para acceso a administradores.

Después, se imprime un menú con dos opciones: 1) Iniciar sesión 2) Añadir un nuevo usuario. El usuario ingresará la opción elegida en un input y será almacenada en la variable **v_menu_principal** para ser evaluada a continuación.

```

1440 #ciclo para verificar opcion de menu
1441 if v_menu_principal == "a" or v_menu_principal == "A":
1442     v_usuario = input("USUARIO: ")
1443     v_contra = input("CONTRASEÑA: ")
1444     for a in range(0, len(lista_admin)):
1445         if lista_admin[a][0] == v_usuario and lista_admin[a][1] == v_contra:
1446             bandera = True
1447     if bandera == True:
1448         print("BIENVENIDO A LIFESTORE")
1449     else:
1450         print("FAVOR DE INGRESAR UN USUARIO VALIDO")
1451
1452 elif v_menu_principal == "b" or v_menu_principal == "B":
1453     v_nuevo_usuario = input("Ingrese su usuario: ")
1454     v_nuevo_contra = input("Ingrese su contraseña: ")
1455     lista_admin.append([v_nuevo_usuario, v_nuevo_contra])
1456     print("SE HA REGISTRADO CON EXITO")
1457 else:
1458     print("Favor de seleccionar una de las opciones del menú")
1459

```

Seguidamente, procedemos a evaluar la variable **v_menu_principal**. Optamos por hacer uso de un bucle IF debido a que las opciones son sumamente limitadas y es mucho más fácil evaluar la variable.

Se evalúa la primera opción teniendo dos opciones y minimizar el error. Se verifica que el usuario y contraseña correspondan y estén en la base de datos. Si es así, la bandera se modifica a "TRUE" para mandar un mensaje de bienvenida, de lo contrario, se mantiene como "FALSE" para mandar un mensaje de usuario no válido.

En caso de ser la opción B, únicamente se le pide al cliente un usuario y contraseña y se agrega a lista con usuarios ya definidos.

Finalmente, si no fue elegida ninguna de las opciones anteriores, se imprime un mensaje pidiendo una opción válida.

PRODUCTOS MÁS VENDIDOS Y REZAGADOS

Seguidamente, fue necesario obtener la cantidad de productos más vendidos y rezagados con el objetivo de establecer una estrategia de ventas y contar con un inventario mucho más optimizado.

```

1465 #PARTE 1
1466 #PRODUCTOS MAS VENDIDOS Y REZAGADOS // 50 M VENTAS Y 100 M BUSQUEDAS
1467
1468 #Esta funcion nos va a ser usada como nuestra propia key para ordenar
1469 def sort_id_product(var):
1470     return var[1]
1471
1472 tamaño = len(lifestore_searches) + 1
1473 lifestore_searches_cont = []
1474 lista_contador = [0] * 100
1475
1476 for item in range(0, tamaño):
1477     lista_contador[lifestore_searches[item][1]] += 1
1478
1479 for item in range(0, len(lista_contador) + 1):
1480     if lista_contador[item] != 0:
1481         lifestore_searches_cont.append([item, lista_contador[item]])
1482

```

En esta sección, podemos observar una función que será usada como nuestra propia llave para ordenar la lista; funcionando éste como pivote.

Seguidamente, establecemos las variables globales que serán usadas en esta función. Iniciando con el tamaño de la matriz de búsqueda, seguidamente de una matriz que será usada para almacenar cada elemento de la matriz en su posición correspondiente dependiendo del id del producto.

Debido a que en la matriz de datos al ser ordenada hay productos que no tienen búsquedas, se hizo un segundo bucle de tipo FOR con el objetivo de evaluar cada elemento y si la cantidad es diferente a cero, agregarlo a una lista final.

```

1483:     liststore_searches_aux = sorted(liststore_searches_cont, key=sort_id_product)
1484:     print("PRODUCTOS MAS BUSCADOS")
1485:     for item in range(0, 50):
1486:         print("Id Producto: {0}, C. Busquedas: {1}", liststore_searches_aux[item][0], liststore_searches_aux[item][1])
1487:     liststore_searches_aux2 = sorted(liststore_searches_cont, key=sort_id_product, reverse=True)
1488:     print("PRODUCTOS MENOS BUSCADOS")
1489:     for item in range(0, 50):
1490:         print("Id Producto: {0}, C. Busquedas: {1}", liststore_searches_aux2[item][0], liststore_searches_aux2[item][1])

```

Esta sección tiene como único propósito la impresión de los 50 productos mayor y menos buscados/solicitados. Para ello, se hizo uso de la función sorted y se almacenó en una matriz auxiliar para la ordenada de mayor a menor tanto como la inversa.

```

1492 tamaño_ventas = len(lifstore_sales) + 1
1493 lifstore_sales_cont = []
1494 lista_sales_contador = [0] * 100
1495
1496 for item in range(0, tamaño_ventas):
1497     lista_sales_contador[lifstore_sales[item][1]] += 1
1498
1499 for item in range(0, len(lista_sales_contador) + 1):
1500     if lista_sales_contador[item] != 0:
1501         lifstore_sales_cont.append([item, lista_sales_contador[item]])
1502

```

En esta imagen se puede visualizar que se hizo uso de una lógica similar a la del caso anterior, en donde se obtiene la longitud de la lista, definen variables globales y seguidamente los ciclos for en donde de la misma manera, se añade un elemento a la posición correspondiente del id.

En el siguiente ciclo de la misma forma se hace un ciclo para validar que únicamente tengamos elementos válidos ($!= 0$) en una nueva lista.

```

1503: lifestore_sales_aux = sorted(lifestore_sales_cont, key=sort_id_product)
1504: print("PRODUCTOS MAS VENDIDOS")
1505: for item in range(0, len(lifestore_sales_aux) + 1):
1506:     print("Id Producto: {0}, C. Ventas: {1}".format(lifestore_sales_aux[item][0], lifestore_sales_aux[item][1]))
1507: lifestore_sales_aux2 = sorted(lifestore_sales_cont, key=sort_id_product, reverse=True)
1508: print("PRODUCTOS MENOS VENDIDOS")
1509: for item in range(0, len(lifestore_sales_aux2) + 1):
1510:     print("Id Producto: {0}, C. Ventas: {1}".format(lifestore_sales_aux2[item][0], lifestore_sales_aux2[item][1]))

```

En este caso, se hizo uso de la misma función para ordenar la lista dependiendo de la columna correspondiente. Seguidamente se hicieron dos ciclos para imprimir cada elemento de la lista.

PRODUCTOS POR RESEÑA EN SERVICIO

```

1512 #PUNTO 2
1513 #PRODUCTOS POR RESERVA EN EL SERVICIO
1514
1515 def sor_id_review(var):
1516     return var[2]
1517
1518 tamaño_review = len(lifstore_sales) + 1
1519 lifstore_review_cont = []
1520 > lista_review = [[(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),(),())...]]
1521
1522 for item in range(0, tamaño_review):
1523     id_producto = lifstore_sales[item][1]
1524     v_review = lifstore_sales[item][2]
1525     lista_review[id_producto].append(v_review)
1526 
```

De la misma manera, se definió una función que nos retornará el valor de la columna que definimos como pivote para ordenar la lista.

Se establecieron las variables globales a emplear en esta sección del ejercicio y comenzamos a evaluar cada elemento de la lista de ventas para obtener el id del producto y su calificación de evaluación, seguidamente, en la lista_review se agregó el valor de la reseña en la posición del id correspondiente.

```

1531 promedio = 0
1532 suma = 0
1533 for i in range(0, len(lista_review) + 1):
1534     if lista_review[i]:
1535         tamaño_aux = len(lista_review[i])
1536         for j in range(0, tamaño_aux) + 1:
1537             suma += lista_review[i][j]
1538         promedio = suma / tamaño_aux
1539         lifestore_review_cont.append([i, promedio])
1540     suma = 0
1541

```

En esta sección se hizo algo muy interesante en donde podemos ver ciclos anidados de dos tipos: if y for. Esto debido a que, en cada posición de la lista, existe otra lista, PERO de diferentes longitudes. Por lo tanto, fue necesario evaluar la longitud de cada uno de los elementos y hacer una sumatoria de cada uno de los elementos para luego sumar y hacer un promedio. Finalmente se agrega a una lista final en donde se define el promedio de evaluación de cada producto.

Debido a que hacemos uso de las variables promedio y suma como globales para almacenar cada una de las evaluaciones y luego dividir entre sí, se iguala a cero en cada iteración de la lista principal evaluada.

```

1542 lifestore_revire_sort = sorted(lifestore_review_cont, key=sort_id_product)
1543 print("PRODUCTOS MEJORES RESEÑAS")
1544 for item in range(0, 20):
1545     print("Id Producto: {0}, V. Reseña: {1}", lifestore_revire_sort[item][0], lifestore_revire_sort[item][1])
1546 lifestore_revire_sort2 = sorted(lifestore_review_cont, key=sort_id_product, reverse=True)
1547 print("PRODUCTOS MENORES RESEÑAS")
1548 for item in range(0, 20):
1549     print("Id Producto: {0}, V. Reseña: {1}", lifestore_revire_sort2[item][0], lifestore_revire_sort2[item][1])
1550

```

Nuevamente, se ordena la lista y se definen dos listas que serán para almacenar los datos de manera ordenada e inversa.

Se imprimen los primer 20 elementos por medio de un ciclo for junto con los dos principales nombres que reciben las columnas.

INGRESOS Y VENTAS

```

1551 #PUNTO 3
1552 #TOTAL DE INGRESOS Y VENTAS PROMEDIO MENSUALES
1553 #TOTAL ANUAL Y MESES CON MAS VENTAS AL AÑO
1554
1555 lista_precios = []
1556 lista_meses = [[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[],[]]
1557
1558 for item in range(0, len(lifestore_products) + 1):
1559     id_producto_a = lifestore_products[item][0]
1560     id_producto_a = lifestore_products[item][2]
1561     lista_precios.append([id_producto_a, id_producto_a])

```

Como último punto, decidimos calcular el total de ingresos y ventas por mes tanto como a lo largo del año. Una vez más, se definieron las dos principales listas a usar (precios y matriz de meses)

Primeramente, se hizo una lista específicamente que contenga el id del producto y su precio con la finalidad de facilitar el acceso y relación de las ventas con el precio de los productos.

```

1563 for item in range(0, len(lifestore_sales) + 1):
1564     fecha_venta = datetime.strptime(lifestore_sales[item][3], '%d, %m, %y')
1565     month = fecha_venta.month
1566     if lifestore_sales[item][4] != 1:
1567         lista_meses[month - 1].append(lifestore_sales[item][1])
1568

```

En este bucle, usamos una variable auxiliar para almacenar la fecha de la venta para obtener únicamente el mes de la compra y almacenarlo en la posición correspondiente de dicha compra. Esto, sólo en caso de que la venta sea diferente a 1, es decir, que no haya tenido devolución. Finalmente, se agrega esta acción a la lista de su mes correspondiente.

```
1569 suma_mes = 0
1570 promedio_ventas = []
1571 ingreso_ventas = []
1572 for item in range(0, len(lista_meses) + 1):
1573     tamano_mes = len(lista_meses[item]) + 1
1574     for month in range(0, tamano_mes):
1575         id_producto_ingreso = lista_meses[item][0]
1576         precio = lista_precios[id_producto][1]
1577         suma_mes += precio
1578     promedio_ingreso = suma_mes / tamano_mes
1579     promedio_ventas.append(promedio_ingreso)
1580     ingreso_ventas.append(suma_mes)
1581     suma_mes = 0
```

Ahora bien, en este ciclo for vamos a sumar y contar cada elemento de cada mes para obtener el promedio de ventas y contar cada uno de los elementos de la lista. Por último, se agrega al ingreso de ventas y se igualan las variables a cero.

```
1583 total_anual = 0
1584 for item in range(0, len(ingreso_ventas) + 1):
1585     total_anual += ingreso_ventas[item]
1586
1587 sort_ventas = []
1588 for item in range(0, len(ingreso_ventas) + 1):
1589     sort_ventas.append([(item + 1), ingreso_ventas[item]])
1590
1591 sort_ventas_2 = sorted(sort_ventas, key=sort_id_product)
1592
```

En este ciclo sumamos el ingreso anual, haciendo una sumatoria de cada uno de los elementos de la lista de los ingresos mensuales haciendo uso de un ciclo for.

```
1593 #INGRESO VENTAS
1594 print("INGRESO DE VENTAS")
1595 for item in range(0, len(ingreso_ventas) + 1):
1596     print("Mes: {0}, Ingreso: {1}", (item + 1), ingreso_ventas[item])
1597 #VENTAS PROMEDIO
1598 print("PROMEDIO DE VENTAS")
1599 for item in range(0, len(promedio_ventas) + 1):
1600     print("Mes: {0}, Ingreso: {1}", (item + 1), promedio_ventas[item])
1601 #TOTAL ANUAL
1602 print("TOTAL INGRESO ANUAL")
1603 print("Total: {0}", total_anual)
1604 #MESES CON MAS VENTAS
1605 print("MESES DE MAYORES VENTAS")
1606 for item in range(0, len(sort_ventas_2) + 1):
1607     print("Mes: {0}, Total venta: {1}", sort_ventas_2[item][0], sort_ventas_2[item][1])
```

Para finalizar, imprimimos el ingreso de ventas, promedio, ingreso anual y la lista ordenada por mayores ingresos mensuales.

Solución al Problema

Después de haber implementado estas metodologías en las rutinas diarias de Lifestore podemos implementar de un método mucho más enfocado en un alcance y, por lo tanto, utilidades.

Ahora bien, obteniendo los datos de las mayores búsquedas por usuarios podemos relacionarlo con la cantidad de productos comprados, esto para analizar qué factor es lo que los lleva a realizar o no la compra. Incluso, se tiene el objetivo a corto plazo de aumentar en un mínimo de 15% las ventas teniendo en cuenta estos datos y hacer contenido específico para estos productos.

Después de haber ordenado los productos más y menos vendidos, se piensa optimizar el inventario de la empresa debido a que finalmente el inventario es dinero estancado el cual puede estar invertido en mayor infraestructura y métodos de alcance para empresas de mayor renombre. De la misma manera, teniendo estos datos se calcula al menos un 0.8 vueltas de inventario; teniendo esta estructura se espera aumentar a un 1.3 para el próximo año y en dos 1.6. Esto, para comenzar un plan de ampliación de sedes a lo largo de la zona Bajío.

Por último, al haber analizado los meses de mayores ingresos se realizó un plan de trabajo y estrategia para aumentar los ingresos en un mínimo 10% e implementar un equipo de marketing para favorecer las ventas en los meses de ventas mínimas. De esta manera, lograr presencia en este mercado.

Conclusión

A manera de conclusión, puedo decir que este proyecto fue de gran interés para desarrollar debido a que fue todo un reto analizar los métodos y la lógica que debe ser implementada para un adecuado funcionamiento. De la misma manera, es de gran interés cómo en los datos se pueden visualizar patrones, sumándole a esto el contexto, se puede obtener una certera proyección de la empresa y, lo más importante; los clientes.

Ahora bien, a través de este proyecto logramos identificar cómo con determinadas cantidades de datos específicas es posible realizar un amplio análisis de la estructura de la empresa y algunas proyecciones, relacionando de manera adecuada los datos.

Este siglo se nos ha dado la oportunidad de tener el mundo en nuestras manos, sólo es cuestión de aprovecharlo.