

Reporte de Caso Práctico 1

Alma Arévalo Loyola

15 de septiembre, 2021

Índice

1. Introducción	1
2. Descripción del código	1
2.1. Log in de usuario	1
2.2. Menú de opciones	2
2.3. Opciones del menú	2
2.4. Listas auxiliares	4
2.5. Ordenamiento y selección	4
3. Solución al problema	5
4. Conclusión	5

Resumen

Este documento es un reporte al proyecto realizado como parte final del curso "Introducción a Python". Dicho proyecto consistió en desarrollar un programa que ayuda a analizar los datos recavados por la empresa Lifestore referentes a sus ventas y búsquedas de productos, con la finalidad de determinar estrategias que mejoren sus ganancias y eviten tener productos rezagados.

1. Introducción

El objetivo de este proyecto como recurso de aprendizaje es aplicar los conocimientos adquiridos sobre las herramientas base del lenguaje de programación Python, así como ver reflejada su utilidad específicamente en el análisis de datos.

Por otro lado, el proyecto planteado como sistema realizado para una tienda de electrónicos, tiene por objetivo revisar datos almacenados relativos a las búsquedas de sus productos, sus ventas y sus reseñas para poder tomar decisiones en beneficio de la empresa.

Al no ser un curso sobre temas de mercado, las conclusiones que se darán son muy básicas o de sentido común, del tipo "ofrecer lo que más se vende y dejar de ofrecer lo que no se vende".

En un ejemplo de pocos productos y pocas ventas esto se podría hacer por simple inspección, viendo las listas y catalogando los productos de acuerdo a su cantidad de búsquedas, ventas y devoluciones.

En este caso, la ventaja de utilizar la programación es lograr evaluar listas grandes de datos con facilidad, automatizar las clasificaciones y ordenaciones, para así poder visualizar datos ya filtrados, por ejemplo, ver solo los 10 productos más vendidos en lugar de todas las ventas de varios años.

2. Descripción del código

2.1. Log in de usuario

El primer requisito para el programa es que incluyera un control de acceso, de manera que solo se pueda acceder con un usuario y contraseña autorizados.

Para esto tuvimos que predefinir una lista de los usuarios autorizados y sus respectivas contraseñas, en variables de tipo *string*.

```
lista_permitidos = [['usuario', 'password'], ['usuario2',  
'1234'], ['usuario3', '8JDk&3t'], ['usuario4', 'password4']]
```

Después utilizamos la función *input* de Python para recibir una cadena escrita por el usuario, para de esta manera compararla con las cadenas almacenadas y determinar si el acceso es correcto.

```
'variables auxiliares para determinar cuando puede ingresar el usuario y cuando  
hizo demasiados intentos'  
acceso = False  
intentos_fallidos = 0  
  
'este ciclo se repite mientras no se ingresen datos correctos pero no se pase de 3 intentos'  
while (acceso == False and intentos_fallidos < 3):  
  
    usuario = input("Ingrese su nombre de usuario: ")  
    password = input("Ingrese su contraseña: ")  
    for permitido in lista_permitidos:  
        if (usuario == permitido[0] and password==permitido[1]):  
            acceso = True  
  
    if acceso == False:  
        print ('Acceso incorrecto.')  
        intentos_fallidos += 1  
        if intentos_fallidos < 3:  
            print('Intente de nuevo.')  
  
if acceso:  
    print ('Acceso correcto.')  
if intentos_fallidos == 3:  
    print ('Demasiados intentos fallidos. Acceso no permitido.')
```

Notemos que el funcionamiento de un ciclo *while* nos permite repetir la misma acción (en este caso imprimir en pantalla y obtener una respuesta del usuario) hasta que se deje de cumplir la condición, para no volver infinito este ciclo, pusimos la condición de que la variable intentos fallidos solo puede llegar al valor 3, y cada vez que el usuario ingresa algo erróneo esta variable aumenta en 1 su valor.

2.2. Menú de opciones

Una vez que se logra acceder con un usuario y contraseña correctos, se muestra un menú de opciones para elegir lo que se quiere ver. De manera similar a lo explicado anteriormente, se utiliza un ciclo *while* para repetir la acción de mostrar el menú y obtener la respuesta del usuario la cantidad de veces que sea necesario, en este caso la condición del *while* se deja de cumplir cuando el usuario elige voluntariamente salir o cuando ingresa dato erróneo 3 veces.

```
'variables auxiliares para determinar cuándo se muestra el menú'
salir = False
intentos_fallidos = 0

'este ciclo se repite mientras el usuario no decida salir y no tenga más de 3 intentos fallidos de ingreso'
while(salir == False and intentos_fallidos < 3):
    print('MENÚ DE OPCIONES: ')
    print(' 1 Productos más vendidos.')
    print(' 2 Productos menos vendidos.')
    print(' 3 Productos más buscados.')
    print(' 4 Productos menos buscados.')
    print(' 5 Productos mejor puntuados.')
    print(' 6 Productos peor puntuados.')
    print(' 0 Salir')
```

2.3. Opciones del menú

Una vez que el usuario ingresa una opción válida del menú de opciones, se muestra en pantalla los datos de la opción correspondiente, se muestran únicamente datos que se consideran relevantes, por ejemplo, se muestra el nombre del producto en lugar del ID. Esta filtración de datos y el ordenamiento correspondiente a lo que pide cada lista se hizo mediante listas auxiliares que serán explicadas en la siguiente sección.

En esta parte del código, únicamente se hace uso de los comandos *if*, *elif* y *else* para mostrar en pantalla lo correspondiente a la opción elegida por el usuario.

```
'al elegir una opcion del menú se muestran los datos relevantes de la lista elegida
y siempre se da la opción de salir o volver al menú'
seleccion = input('Elija una opción del menú ingresando el número correspondiente: ')
if seleccion == '0':
    salir = True
elif seleccion == '1':
    print('Los productos más vendidos son:')
    for i in range(10):
        print(i+1)
        print(mas_vendidos[i][0][1])
        print('Ventas: ', mas_vendidos[i][1])
        print('Devoluciones: ', mas_vendidos[i][2])
    salir = (input('Ingresa cualquier número para volver al menú o 0 para salir: ') == '0')

elif seleccion == '2':
    print('Los productos menos vendidos son:')
```

```

for i in range(10):
    print(i+1)
    print(menos_vendidos[i][0][1])
    print('Ventas: ', menos_vendidos[i][1])
    print('Devoluciones: ', menos_vendidos[i][2] )

    salir = (input('Ingresa cualquier número para volver al menú o 0 para salir: ')=='0')
elif seleccion == '3':
    print('Los productos más buscados son:')
    for i in range(10):
        print(i+1)
        print(mas_buscados[i][0][1])
        print('Búsquedas: ', mas_buscados[i][6])
        print('Ventas: ', mas_buscados[i][1])
        print('Devoluciones: ', mas_buscados[i][2])
    salir = (input('Ingresa cualquier número para volver al menú o 0 para salir: ')=='0')
elif seleccion == '4':
    print('Los productos menos buscados son:')
    for i in range(10):
        print(i+1)
        print(menos_buscados[i][0][1])
        print('Búsquedas: ', menos_buscados[i][6])
        print('Ventas: ', menos_buscados[i][1])
        print('Devoluciones: ', menos_buscados[i][2])
    salir = (input('Ingresa cualquier número para volver al menú o 0 para salir: ')=='0')
elif seleccion == '5':
    print('Los productos mejor puntuados son:')
    for i in range(10):
        print(i+1)
        print(mejor_puntuados[i][0][1])
        print('Puntuación promedio: ', mejor_puntuados[i][4] )
        print('Búsquedas: ', mejor_puntuados[i][6])
        print('Ventas: ', mejor_puntuados[i][1])
        print('Devoluciones: ', mejor_puntuados[i][2])
    salir = (input('Ingresa cualquier número para volver al menú o 0 para salir: ')=='0')
elif seleccion == '6':
    print('Los productos peor puntuados son:')
    for i in range(10):
        print(i+1)
        print(peor_puntuados[i][0][1])
        print('Puntuación promedio: ', peor_puntuados[i][4] )
        print('Búsquedas: ', peor_puntuados[i][6])
        print('Ventas: ', peor_puntuados[i][1])
        print('Devoluciones: ', peor_puntuados[i][2])
    salir = (input('Ingresa cualquier número para volver al menú o 0 para salir: ')=='0')
else:
    print ('Opcion inexistente.')
    intentos_fallidos += 1
    if intentos_fallidos < 3:
        print('Intente de nuevo.')

if intentos_fallidos == 3:
    print ('Demasiados intentos fallidos. Acceso no permitido.')

if (salir):
    print ('La sesión ha terminado.' )

```

2.4. Listas auxiliares

Originalmente nos fueron dadas tres listas:

```
lifestore_searches = [id_search, id product]
lifestore_sales = [id_sale, id_product, score (from 1 to 5), date, refund (1 for true or 0 to false)]
lifestore_products = [id_product, name, price, category, stock]
```

Creamos una nueva lista, de manera que a cada producto de la lista *lifestore_products* se le asigna su cantidad total de ventas, de devoluciones, de búsquedas y el puntaje promedio de sus ventas. Esto se hace mediante ciclos for sobre las listas correspondientes.

```
for producto in lifestore_products:
    'se va a crear un elemento por cada producto existente en la tienda y cada uno tendrá
    asignados los siguientes parámetros'
    sales=0
    refunds =0
    suma_scores = 0
    searches = 0

for venta in lifestore_sales:
    'ciclo para contar el número de ventas y el número de devoluciones con los datos de
    la lista lifestore_sales'
    if producto[0]== venta[1]:
        sales += 1
        refunds += venta[4]
        suma_scores += venta[2]
if (sales == 0):
    'en caso de no tener ventas, no tendrá puntuación'
    puntuado = 0
    score_promedio = 'N/A'
else:
    'en caso de tener ventas, se almacena la puntuación promedio de todas las ventas'
    puntuado = 1
    score_promedio = suma_scores/sales
for search in lifestore_searches:
    if producto[0] == search[1]:
        searches += 1

'se almacena el elemento con todos sus atributos'
productos_ventas.append([producto, sales, refunds, sales - refunds, score_promedio,
puntuado, searches])
```

2.5. Ordenamiento y selección

Todas las siguientes listas son creadas de manera muy similar, haciendo uso de la función *sorted* para ordenar de acuerdo a un índice y seleccionando un rango de índices de 0 a 10, para mostrar los 10 menores de alguna característica o los 10 mayores, en cuyo caso se hace uso del atributo *reverse*.

```
'se almacenan en una nueva lista los 10 más vendidos en orden'
mas_vendidos = sorted(productos_ventas, key=itemgetter(3), reverse=True)[0:10]
```

```
'se almacenan en una nueva lista los 10 menos vendidos en orden'
```

```

menos_vendidos = sorted(productos_ventas, key=itemgetter(3))[0:10]

productos_puntuados = []

'para las listas referentes a puntuación, solo se considerarán los productos que sí están puntuados'
for producto in productos_ventas:
    if producto[5]:
        productos_puntuados.append(producto)

'se almacenan en una nueva lista los 10 mejores puntuados'
mejor_puntuados = sorted(productos_puntuados, key=itemgetter(4), reverse= True)[0:10]

'se almacenan en una nueva lista los 10 peores puntuados'
peor_puntuados = sorted(productos_puntuados, key=itemgetter(4))[0:10]

'se almacenan en una nueva lista los 10 más buscados'
mas_buscados = sorted(productos_ventas, key=itemgetter(6), reverse=True)[0:10]

'se almacenan en una nueva lista los 10 menos buscados'
menos_buscados = sorted(productos_ventas, key=itemgetter(6))[0:10]

```

Todas estas listas fueron utilizadas para mostrar lo que el usuario pide desde el menú de opciones, como estas ya tienen los datos ordenados, en el archivo main, solamente se importan y se pasan como argumento de la función *print* cuando corresponde.

3. Solución al problema

Lo que se puede observar con los datos recopilados y organizados es que hay muchos productos que no se están vendiendo, y que hay muchos productos repetidos, es decir que se tienen varios modelos o marcas de lo mismo y solo uno o dos se venden. Los productos más vendidos son procesadores, y los menos vendidos son varias de las tarjetas de video. De igual manera de los menos vendidos es de los que hay más stock, por lo que la sugerencia es ponerlos en oferta de manera que no haya pérdidas pero salgan esos productos para no desperdiciar espacio en bodega y no volverlos a resurtir. También se puede observar que los productos peor puntuados tienen pocas ventas y algunos tienen devoluciones, y los productos con más ventas tienen muy buenas puntuaciones. Por lo que sería una buena estrategia para la empresa enfocarse en los productos más vendidos y que sean esos los que se mantengan en stock, en este caso son los procesadores y los discos duros. Entre los 10 más vendidos solo se encuentra una tarjeta madre y una tarjeta de video, y en contraste los menos vendidos son en su mayoría tarjetas de video, por lo que se podría reducir los modelos que se ofrecen de estas y quedarse solo con las que sí se venden.

4. Conclusión

Este proyecto me pareció una excelente forma de aplicar inmediatamente lo aprendido en la primera parte del curso así como una oportunidad de visualizar la utilidad por un lado del análisis de datos en la vida cotidiana, que nos ayuda a tomar decisiones más adecuadas y por otro lado de la programación para realizar estos análisis de manera práctica, reduciendo grandes cantidades de datos a algo muy sintetizado que se puede evaluar mucho más fácilmente.

Estoy convencida de que conforme avance el curso y adquiera más herramientas, a proyectos de este tipo se les puede sacar un mayor provecho, considero que mi implementación fue aún muy rudimentaria con herramientas muy básicas como ciclos for y while, sin embargo es una gran motivación ver lo que se puede hacer desde lo más simple y solo me deja con interés de aprender más para en algún momento desarrollar programas más complejos, mejor estructurados y con mayor utilidad real.