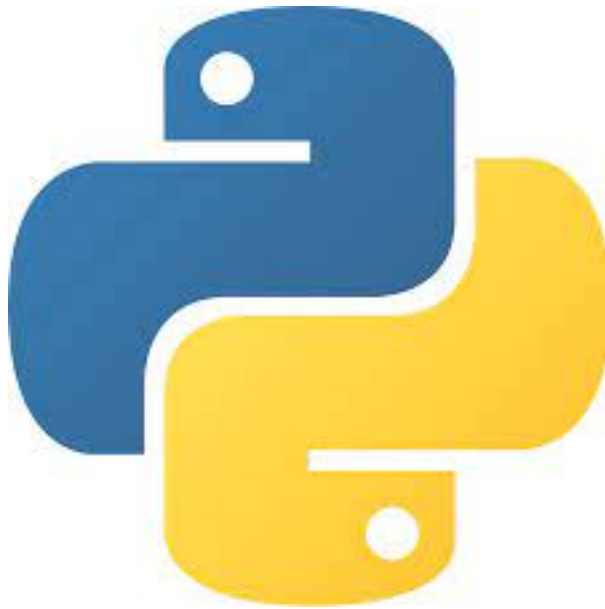


Proyecto 1
Introducción a Python
EMTECH

Brian Gabriel Barajas Guerrero
Grupo 2
Data Science

2021



Índice

1. Introducción	2
2. Definición del código	3
3. Conclusión	6
4. Solución al problema	7

1. Introducción

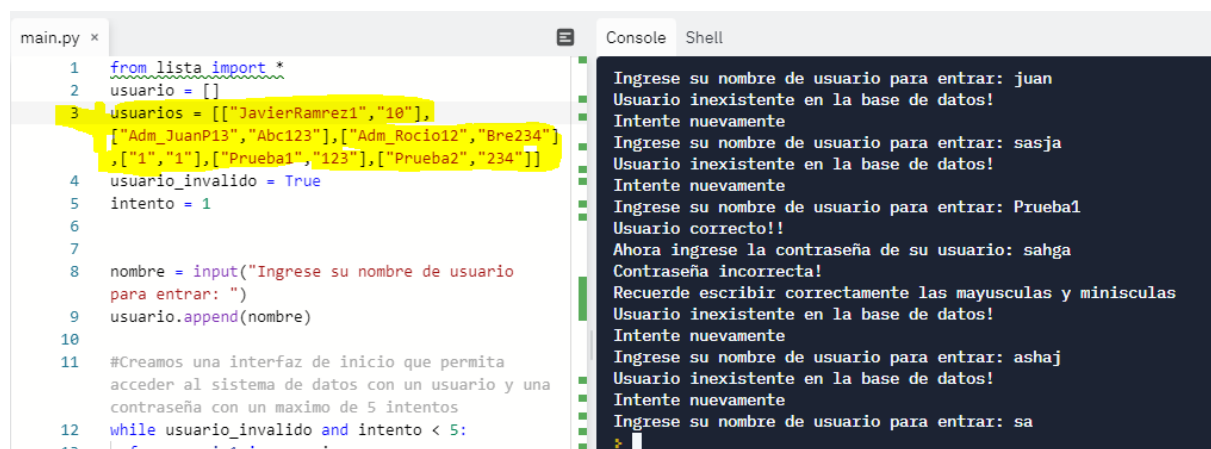
En este proyecto se analizó un caso de una tienda virtual llamada **'LifeStore'** que presentaba un problema de acumulación de productos derivado de, entre otras cosas, la disminución de búsquedas en ciertos productos. La gerencia de ventas a solicitado la realización de un análisis de los datos registrados en la tienda referentes a las ventas y la sugerencia de una estrategia a seguir para solucionar el problema.

En base a esa consigna he realizado un programa en lenguaje Python que dada una lista de datos con las ventas de los productos y la cantidad de productos realiza una organización de datos relevantes que permitan mediante un login de usuario acceder a los datos de forma más 'amigable' y más cómoda para el usuario para facilitar el análisis.

Este código se puede hacer mucho más optimizado usando herramientas más avanzadas de Python pero para propósitos del curso se prefirió por usar solamente los comandos básicos de Python y lo visto solamente en la parte 1 del curso.

2. Definición del código

El código comienza importando las listas con los datos de la tienda que deben ser 'procesados', lo siguiente es definir los usuarios que tendrán acceso a los datos y para eso los colocamos en una lista acompañados de su contraseña. Una vez hecho esto se debe realizar el código que permita el acceso solo a los usuarios autorizados por medio de su contraseña, en caso de introducir un usuario no registrado en el sistema o una contraseña incorrecta se indicara su error al usuario y se le dará un máximo de 5 oportunidades para acceder correctamente al sistema. Las repuestas tecleadas del usuario se guardan en una lista llamada 'usuario' y si el usuario o contraseña es incorrecto se borra la respuesta para almacenar una nueva en el siguiente intento.



```
main.py x
1 from lista import *
2 usuario = []
3 usuarios = [{"JavierRamrez1","10"},
4             [{"Adm_JuanP13","Abc123"},{"Adm_Rocio12","Bre234"}],
5             [{"1","1"},{"Prueba1","123"},{"Prueba2","234"}]]
6 usuario_invalido = True
7 intento = 1
8 nombre = input("Ingrese su nombre de usuario para entrar: ")
9 usuario.append(nombre)
10
11 #Creamos una interfaz de inicio que permita acceder al sistema de datos con un usuario y una contraseña con un maximo de 5 intentos
12 while usuario_invalido and intento < 5:
13     for usuario1 in usuarios:
```

```
Console Shell
Ingrese su nombre de usuario para entrar: juan
Usuario inexistente en la base de datos!
Intente nuevamente
Ingrese su nombre de usuario para entrar: sasja
Usuario inexistente en la base de datos!
Intente nuevamente
Ingrese su nombre de usuario para entrar: Prueba1
Usuario correcto!!
Ahora ingrese la contraseña de su usuario: sahga
Contraseña incorrecta!
Recuerde escribir correctamente las mayusculas y minisculas
Usuario inexistente en la base de datos!
Intente nuevamente
Ingrese su nombre de usuario para entrar: ashaj
Usuario inexistente en la base de datos!
Intente nuevamente
Ingrese su nombre de usuario para entrar: sa
```

Figura 1: Usuarios autorizados y ejemplo de intentos fallidos

Nos aseguramos que se haya introducido un usuario correctamente verificando que no se hayan sobrepasado los 5 intentos y solo en ese caso se ejecutará el bloque de código siguiente. En el código se definen muchas variables temporales que se redefinen después que nos servirán como herramientas secundarias siendo las listas finales las principales

Se define una variable 'continuar' que nos ayudará más adelante y de la línea 43 a la línea 94 se crean listas (a base de bucles) que nos ayudarán a contabilizar las búsquedas de cada producto y se organizaran en los 'mas buscados' y aquellos sin búsquedas.

En la línea 98 se comienza a contabilizar el numero de ventas de cada producto y se guardan los productos 'mas vendidos' y los 'no vendidos'.

```

36 | #print(usuario)
37 | intento += 1
38 | #Ahora empieza el procesamiento de datos
39 | #continuar = True
40 | if intento < 5:
41 |     continuar = True
42 |
43 | #Creacion de listas para contar el numero de busquedas de cada producto
44 | variable = 0
45 | suma = 0
46 | b_sumas = []
47 | b_productos = [0]
48 | for elemento in lifestore_searches:
49 |     if elemento[1] == variable:
50 |         suma += 1
51 |     elif elemento[1] != variable:
52 |         b_sumas.append(suma)
53 |         # print(b_sumas)
54 |         variable = elemento[1]
55 |         b_productos.append(variable)
56 |         suma = 1
57 | b_sumas.append(suma)
58 | b_productos.remove(0)
59 | b_sumas.remove(0)
60 | #print(b_sumas)
61 | #print(b_productos)
62 | # print(len(b_sumas))

```

Figura 2: Identación del código para que se ejecute solo si se introdujo un usuario valido

En la línea 141 comenzamos a promediar a números enteros las calificaciones de los productos y clasificarlos en 'calificación excelente', 'calificación buena' y 'calificación mala' y mostramos sus devoluciones.

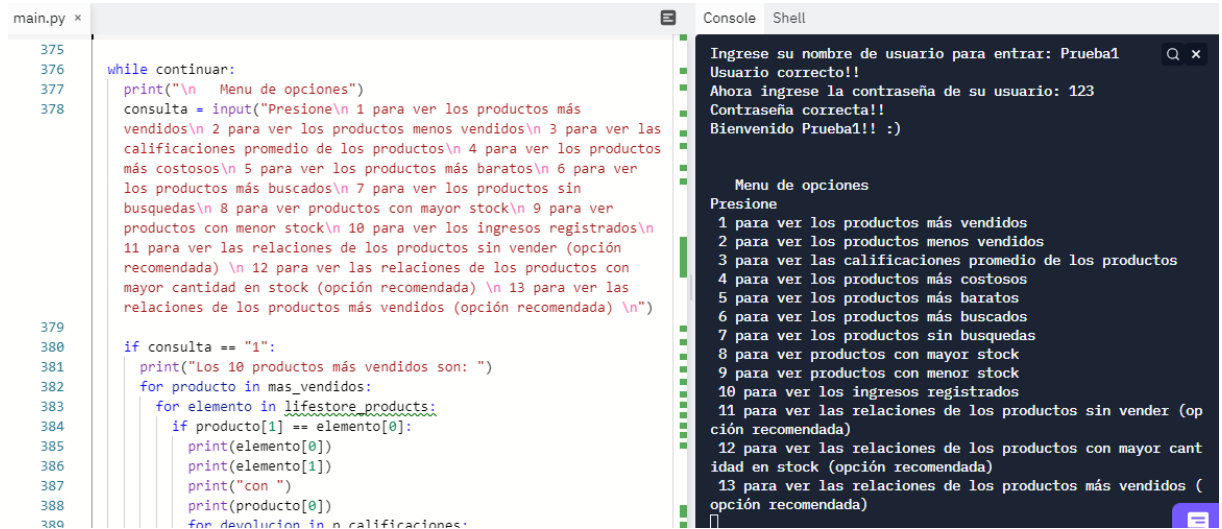
En la línea 174 clasificamos los productos según su precio en los 'mas caros' y 'mas baratos'.

En la línea 190 se realiza un proceso parecido a los anteriores para hacer listas con los productos en mayor cantidad 'mayor stock' y en menor cantidad 'menor stock'.

A partir de la línea 203 comenzamos a buscar algunas relaciones que considero útiles y esto lo hacemos mediante comparar que elemento hay en común dentro de las distintas listas para así saber que elementos no han sido buscados pero tienen precios muy bajos por ejemplo.

En la línea 259 declaramos las listas donde almacenaremos el registro de ventas por mes para sacar datos requeridos por la gerencia como el total de ingresos por mes o el ingreso total al año. En esta parte del código si es evidente que hay formas más óptimas de realizar este proceso pero eso no quita la eficacia del mismo.

Al llegar a la línea 376 finalmente vamos a interactuar con el usuario de nuevo, también usaremos la variable 'continuar' con un valor booleano que definimos al inicio para crear un bucle donde el usuario pueda revisar los datos que el quiera las veces que quiera. Se le presenta al usuario un menú de opciones al que puede acceder tecleando el numero indicado para cada opción. Se muestran opciones para acceder a datos que faciliten el análisis y corroboren la estrategia planteada así como las relaciones claves desarrolladas arriba acompañadas por la recomendación '(opción recomendada)'.



The image shows a code editor window with a file named 'main.py' and a terminal window below it. The code in the editor is a Python script that runs a loop 'while continuar:' and prints a menu of options. The terminal shows the user inputting 'Prueba1' as a username and '123' as a password, both of which are accepted. The terminal then displays the menu options.

```
375
376 while continuar:
377     print("\n  Menu de opciones")
378     consulta = input("Presione\n 1 para ver los productos más
    vendidos\n 2 para ver los productos menos vendidos\n 3 para ver las
    calificaciones promedio de los productos\n 4 para ver los productos
    más costosos\n 5 para ver los productos más baratos\n 6 para ver
    los productos más buscados\n 7 para ver los productos sin
    búsquedas\n 8 para ver productos con mayor stock\n 9 para ver
    productos con menor stock\n 10 para ver los ingresos registrados\n
    11 para ver las relaciones de los productos sin vender (opción
    recomendada) \n 12 para ver las relaciones de los productos con
    mayor cantidad en stock (opción recomendada) \n 13 para ver las
    relaciones de los productos más vendidos (opción recomendada) \n")
379
380 if consulta == "1":
381     print("Los 10 productos más vendidos son: ")
382     for producto in mas_vendidos:
383         for elemento in lifestore_products:
384             if producto[1] == elemento[0]:
385                 print(elemento[0])
386                 print(elemento[1])
387                 print("con ")
388                 print(producto[0])
389         for devolucion in n_calificaciones:
```

Console Shell

```
Ingrese su nombre de usuario para entrar: Prueba1
Usuario correcto!!
Ahora ingrese la contraseña de su usuario: 123
Contraseña correcta!!
Bienvenido Prueba1!! :)

  Menu de opciones
Presione
1 para ver los productos más vendidos
2 para ver los productos menos vendidos
3 para ver las calificaciones promedio de los productos
4 para ver los productos más costosos
5 para ver los productos más baratos
6 para ver los productos más buscados
7 para ver los productos sin búsquedas
8 para ver productos con mayor stock
9 para ver productos con menor stock
10 para ver los ingresos registrados
11 para ver las relaciones de los productos sin vender (op
ción recomendada)
12 para ver las relaciones de los productos con mayor cant
idad en stock (opción recomendada)
13 para ver las relaciones de los productos más vendidos (
opción recomendada)
```

Figura 3: Menú de opciones después de ingresar un usuario valido

A partir de ahí (línea 380) se usan las listas que se crearon anteriormente y por medio de ciclos 'for' y condicionales se usa el ID del producto guardado en las listas para mostrar la información con el nombre del producto directamente. Al terminar de mostrar la información solicitada se pregunta si se desea consultar algún otro dato y de no contestar si se termina de ejecutar el programa. Para algunas opciones como la opción '3' se pusieron condicionales para mostrar al usuario un dato más específico.

El programa permite mostrar una cantidad diversa de datos de forma organizada para buscar relaciones manualmente así como también muestra algunas relaciones importantes seleccionando la opción, se seguirán mostrando datos mientras el usuario lo solicite y en caso.

3. Conclusión

Mirando directamente en la opción 11 del menú se aprecia claramente que la mayoría de los productos que no se vendieron fueron productos que no fueron buscados y contrastando con los datos arrojados de la opción 13 donde vemos que ninguno de los productos más vendidos registraron cero búsquedas por lo que podemos observar la importancia de las búsquedas para las ventas.

En la opción 12 podemos ver que hay algunos productos que presentan un gran acumulamiento de stock sin embargo también son de los productos más vendidos por lo que es de esperar que el acumulamiento baje pronto.

Hay una gran cantidad de productos que no fueron vendidos pero están entre los productos más baratos por lo que podemos atribuir su acumulamiento a su poca visibilidad porque ninguno de los productos más vendidos están entre los productos más caros.

Los datos muestran que a principio de año hay un fuerte ingreso a la tienda, sin embargo, disminuye a medida que avanza el año y tiene una caída fuerte a partir de Agosto.

```
5 para ver los 3 meses con menos ven  
El ingreso en Enero fue:  
120237  
El ingreso en febrero fue:  
110139  
El ingreso en marzo fue:  
164729  
El ingreso en abril fue:  
193295  
El ingreso en mayo fue:  
96394  
El ingreso en junio fue:  
36949  
El ingreso en julio fue:  
26949  
El ingreso en agosto fue:  
3077  
El ingreso en septiembre fue:  
4199  
El ingreso en octubre fue:  
0  
El ingreso en noviembre fue:  
4209  
El ingreso en diciembre fue:  
0
```

Figura 4: Ventas registradas por mes

Tomando estas consideraciones principalmente crearemos una propuesta de solución al problema de la tienda.

4. Solución al problema

Una parte importante del problema es la falta de visibilidad de los productos debido a la baja de búsquedas así que se recomienda fuertemente la implementación de una sección de sugerencias en la página de la tienda virtual, de preferencia que esté al inicio de la página y principalmente se deberán incluir los productos con bajos costos y sin ventas para incentivar el flujo de productos. También se recomienda retirar del mercado una cantidad importante de los productos con mayor costo y sobre todo aquellos que coincidan con los productos no vendidos.

Retirar del mercado todos los productos con promedio de calificaciones malas para evitar dar una mala imagen de la tienda y evitar las devoluciones y acumulamiento de dichos productos.

Se recomienda tener un abastecimiento normal (por mes) solo la primera mitad del año pero para la segunda se recomienda fuertemente cambiar los productos que pueda ofrecer la tienda pues los productos que funcionan la primera mitad del año no lo hacen la segunda, también se podría intentar dar exposición a los productos con pocas búsquedas mediante la plataforma a ver si funcionan para esos meses pero igualmente es preferible bajar la adquisición de productos durante esa última mitad del año para evitar un acumulamiento.

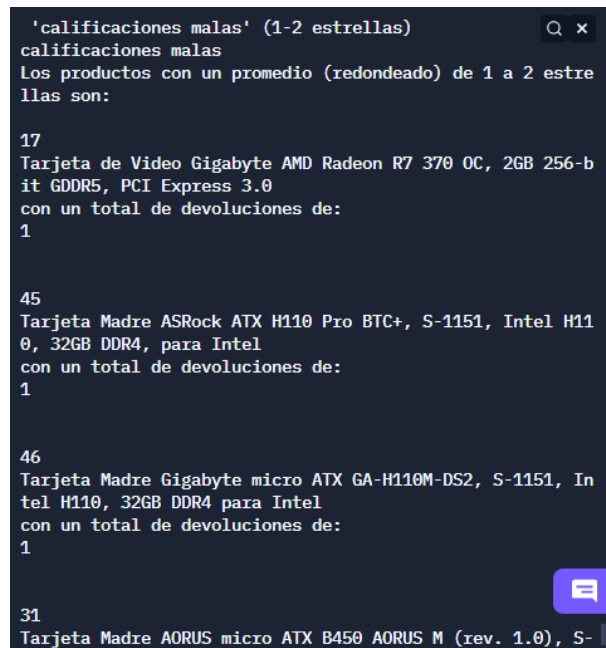


Figura 5: Lista de productos con calificación baja y también son los que presentan mayor numero de devoluciones