



Fundamentos de programación con Python

Tutores curso:

Edgar Egurrola

Jasiel Mariscal

UNIVERSIDAD TECNOLÓGICA DE ESCUINAPA

Alumno:

Kevin Omar Llamas López

Grupo #8

Proyecto Final de Fundamentos de Programación con Python

Link Github:

<https://github.com/KevinLlamas01/emtech-trabajos.git>

Escuinapa, 19 de Junio del 2022

índice:

Portada y link github.....	1
Introducción	3
Definición de código.....	3-9
Solución al problema	9
Conclusión	9

Introducción: El presente documento se realizó con la intención de mostrar una solución a la tienda virtual LifeStore para reducir la acumulación de inventario, todo esto basado por medio de información obtenida a través de un código en lenguaje Python cuyos resultados nos permiten analizar que se debe hacer para evitar este problema en la tienda.

Definición de código:

Primeramente, están declaradas las listas con las se crearemos el resto de código para obtener los resultados a analizar las cuales son 3 y se componen así:

```
lifestore_searches = [id de búsqueda, id de producto]
```

```
lifestore_sales = [id de venta, id de producto, calificación (de 1 a 5), fecha, refund  
(1 for true or 0 to false)]
```

```
lifestore_products = [id de producto, nombre, precio, categoría, el stock]
```

Después de eso se generó el código necesario para el login:

Lista de usuarios:

```
usuarios=['emtech']
```

Lista de contraseñas:

```
contraseñas=["caso1"]
```

La variable de acceso:

```
access=False
```

El proceso de validación de contraseña que es un while break este analiza si lo ingresado corresponde a elementos de la lista de usuarios y contraseñas, si coincide la contraseña con el respectivo usuario el bucle termina y nos acceso a los resultados a analizar , pero si no coincide el bucle continuara hasta que se ingresen los datos correctos

```
while True:
```

```
    user =input ("Ingresa tu usuario: ")
```

```
    pswd=input("Ingresa tu contraseña: ")
```

```
    if user in usuarios and pswd in contraseñas:
```

```
        if usuarios.index(user)== contraseñas.index(pswd):
```

```
            access=True
```

```
            break
```

```
else:
    print("Intenta nuevamente")
else:
    print("Intenta nuevamente")
#luego de una validacion correcta se da la bienvenida
if access:
    print("Aqui se agrega el codigo para el analisis de los productos: \n")
```

Después de eso se declaración la mayoría de las variables necesarias para el resto del código:

Aquí se da una declaracion de listas necesarias para el almacenado de todos productos buscado y vendidos

```
todos_p_vendidos=[]
```

```
todos_p_buscados=[]
```

Lista para id de productos

```
idproductos=[]
```

lista donde se almacenaran todas las categorias que existen de los productos

```
categorias=[]
```

declaracion de lista donde se almacenara la cantidad de productos vendidos, id del producto y su respectiva categoria

```
lasventas=[]
```

declaracion de lista donde se almacenara la cantidad de busquedas totales por producto y su idproducto y su categoria

```
busquedas_totales=[]
```

declaracion de lista donde se almacenara las listas de ventas separadas por categoria

```
lista_generalventas = []
```

declaracion de lista donde se almacenaran las listas de busquedas separadas por categoria

```
lista_generalbusquedas = []
```

lista donde se almacenara el top 5 ventas

top5ventas=[]

lista donde se almacenara el top 10 busquedas

top10busquedas=[]

lista para los id de productos

idproductos=[]

lista indice para presentar el idproducto con su respectivo nombre

indice=[]

Por medio de un ciclo for se realizo el almacenado de todos los idproductos

for l in lifestore_products:

idproductos.append(l[0])

Por medio de dos ciclo for se realizo el almacenado para la lista indice de productos

for k in lifestore_products:

for h in idproductos:

if(k[0]==h):

indice.append([h,k[1]])

Por medio de un ciclo for se realizo el almacenamiento de todos los productos vendidos

for k in lifestore_sales:

todos_p_vendidos.append(k[1])

Por medio de un ciclo for se realizo el almacenamiento de todas las busquedas de productos

for x in lifestore_searches:

```
todos_p_buscados.append(x[1])
```

Por medio de dos ciclo for se realizo el almacenamiento de lista lasventas con el conteo de ventas de cada producto, idproducto y su categoria

```
for i in lifestore_products:
    cont=0
    for x in todos_p_vendidos:
        if(i[0]==x):
            cont+=1
    lasventas.append([cont,i[0],i[3]])
```

Por medio de dos ciclo for se realizo almacenamiento de lista busquedas_totales con el conteo de búsquedas de cada producto, idproducto y su categoría

```
for i in lifestore_products:
    cont=0
    for z in todos_p_buscados:
        if(i[0]==z):
            cont+=1
    busquedas_totales.append([cont,i[0],i[3]])
```

Por medio de un ciclo for se realizo almacenamiento de lista categorias que existen en los productos

```
for i in lifestore_products:
    if(i[3] not in categorias):
        categorias.append(i[3])
```

Aquí se imprimio el indice de productos para identificar cual es su nombre correspondiente a la id

```
print("Indice producto \n")
```

```
print("[idproducto,nombre de producto] \n")
```

Para mejor visualización la impresión se realizó por medio con ayuda de un for

```
for o in indice:
```

```
    print(o, "\n")
```

Por medio de un for se realizó el almacenamiento de lista top5ventas ordenadas de mayor a menor y posteriormente se imprimieron

```
cont=0
```

```
for k in sorted(lasventas,reverse=True):
```

```
    cont+=1
```

```
    if(cont<=5):
```

```
        top5ventas.append(k)
```

```
print("top 5 productos mas vendidos: \n")
```

```
print("[cantidad,idproducto,categoria] \n")
```

```
print(top5ventas, "\n")
```

Por medio de un for se realizó el almacenamiento de lista top10busquedas ordenadas de mayor a menor y posteriormente se imprimieron

```
cont=0
```

```
for m in sorted(busquedas_totales,reverse=True):
```

```
    cont+=1
```

```
    if(cont<=10):
```

```
        top10busquedas.append(m)
```

```
print("top 10 productos mas buscados: \n")
```

```
print("[cantidad,idproducto,categoria] \n")
```

```
print(top10busquedas, "\n")
```

Por medio de dos for se hizo el almacenamiento para para lista_generalventas

```
for i in categorias:
```

declaracion lista donde se pondran las ventas de productos de misma categoria

```
ventas_por_categoria = []
```

```
for j in lasventas:
```

```
    if i==j[2]:
```

```
        ventas_por_categoria.append(j)
```

aqui en el append se hace un sorted facilitar las listas de menos ventas

```
lista_generalventas.append(sorted(ventas_por_categoria))
```

Por medio de dos for se realizo el almacenamiento para lista_generalbusquedas

```
for i in categorias:
```

declaracion lista donde se pondran las busquedas de productos de misma categoria

```
    busquedas_por_categoria = []
```

```
    for j in busquedas_totales:
```

```
        if i==j[2]:
```

```
            busquedas_por_categoria.append(j)
```

aqui en el append se hace un sorted facilitar las listas de menos busquedas

```
    lista_generalbusquedas.append(sorted(busquedas_por_categoria))
```

```
print("Top 5 productos con menores ventas por categoria: \n")
```

```
print("[cantidad,idproducto,categoria] \n")
```

por medio de este for se hizo la impresión del top 5 menos vendidos de cada categoria

```
for i in lista_generalventas:
```

```
    print("Menos vendidos:",i[:5],"\n")
```

```
print("Top 10 productos con menores busquedas por categoria: \n")
```



```
print("[cantidad,idproducto,categoria] \n")
```

por medio de este for se hizo la impresión del top 10 menos buscados de cada categoría

```
for j in lista_generalbusquedas:
```

```
    print("Menos buscados:",j[:10],"\n")
```

Solución al problema:

En base a los resultados obtenidos los productos que se tendrían que retirar del mercado corresponden a todos los de la categoría memorias usb, lo correcto sería eliminar esa categoría ya que no está funcionando bien ya que en búsquedas y ventas es de 0 o intentar buscar mejor mercancía para esa categoría, que son los del id 60 y 61 , en los procesadores sería retirar el idproducto 9 ya que no es tan es el menos buscado y menos vendido de esa categoría, en tarjetas de video sería los productos de id 14 , 16,19 y 20 ya que en las listas en menos ventas y búsquedas son los principales, y lo mismo con los de tarjetas madres los productos 30 , 32 ,34 y 36 y en los discos duros los productos 53,55,56,58 y 59, en pantallas los productos 62 ,64 ,65 y 68, en bocinas los productos 75,77,78 y 79 y en la categoría audífonos sería retirar los productos 86,87,88,90 y 91 , tomando en cuenta que son los principales en su categoría en menores ventas y búsquedas.

Algo que podría servir para disminuir la acumulación de inventario de estos productos antes de retirarlos del mercado podría funcionar bajarles los precios e incluso dar descuentos de ventas por mayoreo de ellos para sacar mínimo algo de ganancia, y algo extra es aumentar ganancias estaría bien aumentar el inventario del producto mas vendido y buscado que es el 54 ya que este siempre estará disponible y tiene más probabilidades de venderse.

Conclusión: Al realizar el código para encontrar la solución a la problemática me quedo como enseñanza que son muy importantes las iteraciones por medio de ciclos para estos tipos de casos de listas que se relacionan entre sí. Y fue el código mas grande que hicimos hasta ahora en el curso por lo cual fue muy importante la buena declaración de variables en un formato para facilitar su observación y que cada parte del proceso funcionara de manera correcta ya que si una parte no funcionaba bien afectaría los resultados finales o no se lo lograría obtener nada de resultados que nos ayudaran a tener solución a la problemática ,esta práctica quizás no fue demasiado grande pero nos puede dar una visión de que se tiene que seguir un cierto proceso para este tipo de casos para que el usuario pueda visualizar lo deseado de una manera eficaz y cómoda, y como todo lo generado pueda ser muy útil a la hora de tomar decisiones desde un negocio pequeño hasta para las empresas grandes.