



EMERGING TECHNOLOGIES INSTITUTE

TRABAJO FINAL
BLOQUE 1

LÓPEZ GONZÁLEZ ELISSA

Grupo 4
DATA SCIENCE

Fecha:
6 de septiembre de 2020

ÍNDICE

1. INTRODUCCIÓN.....	3
2. DEFINICIÓN DEL CÓDIGO.....	3
3. SOLUCIÓN AL PROBLEMA.....	10
4. CONCLUSIONES.....	10

1. INTRODUCCIÓN

Lifystore es una tienda virtual que maneja una amplia gama de artículos, sin embargo, recientemente nos hemos visto en apuros debido a la acumulación de inventario. Investigaciones han ayudado en la búsqueda del origen de este problema. Los resultados de esta investigación interna evidenciaron una deficiencia en la rotación del inventario, esto quiere decir que existe una reducción en las búsquedas de un importante grupo de productos, lo que explica el sustancial decremento en las ventas del último trimestre.

El propósito de este reporte es detallar y resaltar el análisis de rotación de los productos realizado con la finalidad de dar una conclusión y una posible solución al grave problema que enfrenta la tienda. El reporte consta de una sección en la que se describe el procedimiento realizado, posteriormente una conclusión y un plan de tratamiento del problema.

Cabe señalar que el reporte digital es de acceso restringido por ser considerado como información vulnerable confidencial, esto con la finalidad de evitar posibles filtrados de información que puedan poner en riesgo a la tienda. En el apartado “Definición del código” se incluyen los usuarios y contraseñas.

A continuación se agrega el link para el repositorio de GitHub:

<https://github.com/1608elissa/primer-repositorio-.git>

<https://github.com/1608elissa/primer-repositorio->

2. DEFINICIÓN DEL CÓDIGO

```
#IMPORTAR LAS LISTAS
```

```
from lifestore_file import lifestore_sales  
from lifestore_file import lifestore_products  
from lifestore_file import lifestore_searches
```

```
#VERIFICAR QUE APARECEN LAS LISTAS
```

```
print(lifestore_sales)  
print(lifestore_products)  
print(lifestore_searches)
```

```
#CREAR USUARIO CON CONTRASEÑA MEDIANTE LISTAS DE LISTAS
```

```

admins =
[["Javier","boss"],["Emtech","business"],["Elissa","student"]]
#CONTIENE USUARIO Y LA CONTRASEÑA PERTINENTE PARA CADA USUARIO
es_admin = 0 #SIRVE COMO CONTADOR DE RESPUESTA CORRECTA O
INCORRECTA PARA CONDICIONAR LOS SIGUIENTES PASOS

usuario_entrada = input("Ingresa tu nombre de usuario:") #COMIENZA
PIDIENDO USUARIO
usuario_contraseña = input("Ingresa tu contraseña:") #PIDE LA
CONTRASEÑA

while es_admin != 1: #CALIFICACIÓN DE USUARIO Y CONTRASEÑA, SI
COINCIDEN, PASA A LA SELECCIÓN DEL MENÚ, DE LO CONTRARIO SE REPITE
HASTA QUE COINCIDA
    for usuario in admins:
        if usuario_entrada == usuario[0] and usuario_contraseña ==
usuario[1]:
            print("Bienvenido, has ingresado como administrador")
            es_admin = 1
        if es_admin==0:
            print("Usuario o contraseña incorrectos")
            usuario_entrada = input("Ingresa tu nombre de usuario:")
            usuario_contraseña = input("Ingresa tu contraseña:")

#SELECCIÓN DE LAS POSIBLES OPCIONES
es_admin = 1 #CONDICIONAL, SI INGRESÓ COMO USUARIO, PUEDE
SELECCIONAR EL MENÚ
correcta = 0 #CONSIDERA SI CONTESTÓ BIEN, SE VAN SUMANDO
opción = 0 #DEPENDIENDO LA OPCIÓN SELECCIONADA SUMA DIFERENTES
PUNTOS, SIRVE PARA SABER QUÉ OPCIÓN DEBE IMPRIMIR

if es_admin == 1:#SELECCIÓN DEL MENÚ, SI NO RESPONDE ALGUNA DE LAS
OPCIONES, SE REPITE
    print("Bienvenido...")
    opción_selec = input("Para continuar, elige una opción: \n 1.
Productos más vendidos y buscados \n 2. Productos menos vendidos y
buscados \n 3. Productos con mejor y peor reseña \n 4. Total de
ingresos mensuales \n Selecciona tu opción:")

    while correcta != 1:
        if opción_selec == "1":
            print("Seleccionaste 1. Productos más vendidos y buscados")
            opción = 1
            correcta += 1
        elif opción_selec == "2":

```

```

        print("Seleccionaste 2. Productos menos vendidos y
buscados")
        opción = 2
        correcta += 1
    elif opción_selec == "3":
        print("Seleccionaste 3. Productos con mejor y peor reseña")
        opción = 3
        correcta += 1
    elif opción_selec == "4":
        print("Seleccionaste 4. Total de ingresos mensuales")
        opción = 4
        correcta += 1
    else:
        print("Tu selección es incorrecta")
        opción_selec = input("Inténtalo nuevamente")

```

#CONTAR PRODUCTOS VENDIDOS

```

contador = 0 #SIRVE PARA EVITAR BUCLES INFINITOS
total_ventas = [] #SE VAN AGREGANDO LOS DATOS SOLICITADOS EN LA
LISTA VACÍA. SE VAN CONTANDO LOS PRODUCTOS PARA POSTERIORMENTE
HACER LA LISTA

```

```

for producto in lifestore_products:
    for venta in lifestore_sales:
        if producto[0] == venta[1]:
            contador += 1 #va sumando cada uno
        formato = [producto[0],producto[1],contador]
        total_ventas.append(formato) #agrega los datos a la lista
        contador = 0 #resetea el contador para evitar bucles infinitos

```

#CONTAR PRODUCTOS BUSCADOS

```

contar = 0 #SIRVE PARA EVITAR BUCLES INFINITOS
total_buscados = [] #SE VAN AGREGANDO LOS DATOS SOLICITADOS EN LA
LISTA VACÍA. SE VAN CONTANDO LOS PRODUCTOS PARA POSTERIORMENTE
HACER LA LISTA

```

```

for producto in lifestore_products:
    for busqueda in lifestore_searches:
        if producto[0] == busqueda[1]:
            contar += 1 #va sumando cada uno
        bonito_formato = [producto[0],producto[1],contar]
        total_buscados.append(bonito_formato) #agrega los datos a la
lista
        contar = 0 #resetea el contador para evitar bucles infinitos

```

```
#ORDENAR LOS PRODUCTOS DEL MÁS AL MENOS VENDIDO (OPCIÓN 1)
grupos_ordenados_venta_maxim = [] #LISTA VACÍA PARA IR INGRESANO
LOS DATOS SOLICITADOS AUTOMÁTICAMENTE DENTRO DE LA LISTA. ESTOS
DATOS YA ESTÁN ORDENADOS DEL MAYOR AL MENOR VENDIDOS
```

```
while total_ventas:
    maxim = total_ventas[0][2]
    lista_max = total_ventas[0]
    for totvent in total_ventas:
        if totvent[2] > maxim: #INDICA EL ORDEN DE MAYOR A MENOR
            maxim = totvent[2]
            lista_max = totvent
    grupos_ordenados_venta_maxim.append(lista_max) #agrega los datos
a la lista
    total_ventas.remove(lista_max) #va eliminando de la lista para
que no busque lo que ya encontró antes
```

```
#ORDENAR LOS PRODUCTOS DEL MÁS AL MENOS BUSCADO (OPCIÓN 1)
grupos_ordenados_busqueda_maxi = [] #LISTA VACÍA PARA IR INGRESANO
LOS DATOS SOLICITADOS AUTOMÁTICAMENTE DENTRO DE LA LISTA. ESTOS
DATOS YA ESTÁN ORDENADOS DEL MAYOR AL MENOR BUSCADOS
```

```
while total_buscados:
    maxi = total_buscados[0][2]
    lista_maxi = total_buscados[0]
    for totbusq in total_buscados:
        if totbusq[2] > maxi: #INDICA EL ORDEN DE MAYOR A MENOR
            maxi = totbusq[2]
            lista_maxi = totbusq
    grupos_ordenados_busqueda_maxi.append(lista_maxi) #agrega los
datos a la lista
    total_buscados.remove(lista_maxi) #va eliminando de la lista
para que no busque lo que ya encontró antes
```

```
#ORDENAR LOS PRODUCTOS DEL MENOS AL MENOS VENDIDO (OPCIÓN 2)
grupos_ordenados_venta_mini = [] #LISTA VACÍA PARA IR INGRESANO
LOS DATOS SOLICITADOS AUTOMÁTICAMENTE DENTRO DE LA LISTA. ESTOS
DATOS YA ESTÁN ORDENADOS DEL MENOR AL MAYOR VENDIDOS
```

```
while total_ventas:
    mini = total_ventas[0][2]
    lista_mini = total_ventas[0]
    for totavent in total_ventas:
        if totavent[2] < mini: #INDICA EL ORDEN DE MENOR A MAYOR
            mini = totavent[2]
```

```
    lista_mini = totavent
    grupos_ordenados_venta_mini.append(lista_mini) #agrega los datos
a la lista
    total_ventas.remove(lista_mini) #va eliminando de la lista para
que no busque lo que ya encontró antes
```

```
#ORDENAR LOS PRODUCTOS DEL MENOS AL MENOS BUSCADO (OPCIÓN 2)
grupos_ordenados_busqueda_mini = [] #LISTA VACÍA PARA IR INGRESANO
LOS DATOS SOLICITADOS AUTOMÁTICAMENTE DENTRO DE LA LISTA. ESTOS
DATOS YA ESTÁN ORDENADOS DEL MENOR AL MAYOR BUSCADOS
```

```
while total_buscados:
    minim = total_buscados[0][2]
    lista_minim = total_buscados[0]
    for totbusqu in total_buscados:
        if totbusqu[2] < minim: #INDICA EL ORDEN DE MENOR A MAYOR
            minim = totbusqu[2]
            lista_minim = totbusqu
    grupos_ordenados_busqueda_mini.append(lista_minim) #agrega los
datos a la lista
    total_buscados.remove(lista_minim) #va eliminando de la lista
para que no busque lo que ya encontró antes
```

```
#CONTAR LAS MEJORES RESEÑAS (OPCIÓN 3)
contaresena = 0 #SIRVE PARA EVITAR BUCLES INFINITOS
total_resenas = [] #SE VAN AGREGANDO LOS DATOS SOLICITADOS EN LA
LISTA VACÍA. SE VAN CONTANDO LOS PRODUCTOS PARA POSTERIORMENTE
HACER LA LISTA
```

```
for producto in lifestore_products:
    for resena in lifestore_sales:
        if producto[0] == resena[2]:
            contaresena += 1 #va sumando cada uno
            formato_bonito = [producto[0],producto[2],contaresena]
            total_resenas.append(formato_bonito)
        contaresena = 0 #resetea el contador para evitar bucles
infinitos
```

```
#ORDENAR LAS RESEÑAS DE MAYOR A MENOR (OPCIÓN 3)
grupos_ordenados_resena_maxi = [] #LISTA VACÍA PARA IR INGRESANO
LOS DATOS SOLICITADOS AUTOMÁTICAMENTE DENTRO DE LA LISTA. ESTOS
DATOS YA ESTÁN ORDENADOS DEL MAYOR AL MENOR PUNTAJE EN LAS RESEÑAS
```

```
while total_resenas:
    maximo = total_resenas[0][2]
```

```

lista_maximo = total_resenas[0]
for totrese in total_resenas:
    if totrese[2] > maximo: #INDICA EL ORDEN DE MAYOR A MENOR
        maximo = totrese[2]
        lista_maximo = totrese
    grupos_ordenados_resena_maxi.append(lista_maximo) #agrega los
datos a la lista
    total_resenas.remove(lista_maximo) #va eliminando de la lista
para que no busque lo que ya encontró antes

#ORDENAR LAS RESEÑAS DE MAYOR A MENOR
grupos_ordenados_resena_minim = [] #LISTA VACÍA PARA IR INGRESANO
LOS DATOS SOLICITADOS AUTOMÁTICAMENTE DENTRO DE LA LISTA. ESTOS
DATOS YA ESTÁN ORDENADOS DEL MENOR AL MAYOR PUNTAJE EN LAS RESEÑAS

while total_resenas:
    minimo = total_resenas[0][2]
    lista_minimo = total_resenas[0]
    for totresen in total_resenas:
        if totresen[2] < minimo: #INDICA EL ORDEN DE MENOR A MAYOR
            minimo = totresen[2]
            lista_minimo = totresen
        grupos_ordenados_resena_minim.append(lista_minimo) #agrega los
datos a la lista
        total_resenas.remove(lista_minimo) #va eliminando de la lista
para que no busque lo que ya encontró antes

#TOTAL DE INGRESOS Y VENTAS ANUALES (OPCIÓN 4)
total_ingre = [] #SE VAN AGREGANDO LOS DATOS SOLICITADOS EN LA
LISTA VACÍA. SE VAN CONTANDO LOS PRODUCTOS PARA POSTERIORMENTE
HACER LA LISTA

for ingresos in lifestore_sales:
    lista_ingresos = ingresos[3][3:5]
    total_ingre.append(lista_ingresos) #agrega los datos a la lista
    total_ingre.remove(lista_ingresos) #va eliminando de la lista
para que no busque lo que ya encontró antes

#ORDENAR SEGÚN LOS MESES (OPCIÓN 4)
grupos_ordenados_ingreso = [] #LISTA VACÍA PARA IR INGRESANO LOS
DATOS SOLICITADOS AUTOMÁTICAMENTE DENTRO DE LA LISTA. ESTOS DATOS
YA ESTÁN ORDENADOS DEL MENOR AL MAYOR SEGÚN EL MES EN QUE SE
REALIZÓ

while total_ingre:

```



```

for totingre in total_ingre:
    if totingre[2] < minimo: #INDICA EL ORDEN DE MENOR A MAYOR
        minimo = totingre[2]
        lista_minimos = totingre
    grupos_ordenados_ingreso.append(lista_minimos) #agrega los datos
a la lista
total_ingre.remove(lista_minimos) #va eliminando de la lista
para que no busque lo que ya encontró antes

```

```

#IMPRIME LOS RESULTADOS EN LISTA
if opción = 1 #INDICA "REALIZA ESTA OPCIÓN SI Y SÓLO SI, SE
SELECCIONÓ LA OPCIÓN 1", por eso se agregó la opción con el número
en el menú principal

```

```

    for indice in range(0,50): #SÓLO IMPRIME LOS PRIMEROS 50
PRODUCTOS DE LA LISTA DE MÁXIMOS VENDIDOS Y BUSCADOS

```

```

        print("El producto:
\n",grupos_ordenados_venta_maxim[indice][1],"\n","se
vendió",grupos_ordenados_venta_maxim[indice][2],"veces")
        print("El producto:
\n",grupos_ordenados_busqueda_maxi[indice][1],"\n","se
buscó",grupos_ordenados_busqueda_maxi[indice][2],"veces")

```

```

if opción = 2 #INDICA "REALIZA ESTA OPCIÓN SI Y SÓLO SI, SE
SELECCIONÓ LA OPCIÓN 2", por eso se agregó la opción con el número
en el menú principal

```

```

    for indice in range(0,50): #SÓLO IMPRIME LOS PRIMEROS 50
PRODUCTOS DE LA LISTA DE MÍNIMOS VENDIDOS Y BUSCADOS

```

```

        print("El producto:
\n",grupos_ordenados_venta_mini[indice][1],"\n","se
vendió",grupos_ordenados_venta_mini[indice][2],"veces")
        print("El producto:
\n",grupos_ordenados_busqueda_mini[indice][1],"\n","se
buscó",grupos_ordenados_busqueda_mini[indice][2],"veces")

```

```

if opción = 3 #INDICA "REALIZA ESTA OPCIÓN SI Y SÓLO SI, SE
SELECCIONÓ LA OPCIÓN 3", por eso se agregó la opción con el número
en el menú principal

```

```

    for indice in range(0,20): #SÓLO IMPRIME LOS PRIMEROS 20
PRODUCTOS DE LA LISTA DE MEJORES Y PEORES RESEÑAS

```

```

        print("El producto:
\n",grupos_ordenados_resena_maxi[indice][1],"\n","se
vendió",grupos_ordenados_resena_maxi[indice][2],"veces")

```

```
print("El producto:  
\n",grupos_ordenados_resena_minim[indice][1],"\n","se  
buscó",grupos_ordenados_resena_minim[indice][2],"veces")  
  
if opción = 4 #INDICA "REALIZA ESTA OPCIÓN SI Y SÓLO SI, SE  
SELECCIONÓ LA OPCIÓN 4", por eso se agregó la opción con el número  
en el menú principal. IMPRIME LOS PRODUCTOS SEGÚN EL MES  
for ventas in grupos_ordenados_ingreso:  
    print(grupos_ordenados_ingreso)
```

3. SOLUCIÓN AL PROBLEMA

El propósito de este reporte fue detallar y resaltar el análisis realizado de rotación de los productos con la finalidad de dar una conclusión y una posible solución al grave problema que enfrenta la tienda. Sin embargo, este no se pudo llevar a cabo, por lo que no se pudieron obtener los análisis esperados y por lo tanto se sugiere un nuevo análisis para determinar cuáles podrían ser los siguientes pasos por seguir para brindar una solución eficaz al problema.

4. CONCLUSIONES

Se sugiere un nuevo análisis de los datos puesto que los datos obtenidos no fueron suficientes y por lo tanto no pueden ser concluyentes para desarrollar un plan de acción que brinde una real solución al problema que enfrenta la tienda. No obstante, se sugiere la implementación de una plataforma que, con base en las búsquedas del usuario, haga una sugerencia de productos similares al que está solicitando el usuario, pero únicamente de los productos que se encuentran en la lista de rezagados. De igual manera, se sugiere la eliminación de los productos que tienen malas reseñas puesto que al ser un producto que el cliente considera de mala calidad, no se venderá.