

# CASO 1. INTRODUCCIÓN A PYTHON

*LifeStore*

**Nombre del alumno:**

Agustina Guadalupe Arellanes Salazar

**Nombre del tutor:**

Edgar Egurrola

**Fecha:**

22/06/22

## **Índice**

<b>Índice</b>	<b>2</b>
<b>Introducción</b>	<b>3</b>
<b>Definicion de codigo</b>	<b>4</b>
<b>Importación del lifestore_file</b>	<b>4</b>
<b>Login</b>	<b>4</b>
<b>Reporte de ventas mayores</b>	<b>5</b>
<b>Reporte de ventas menores</b>	<b>6</b>
<b>Solución al problema</b>	<b>7</b>
<b>Conclusión</b>	<b>8</b>

## Introducción

LifeStore es una tienda virtual que maneja una amplia gama de artículos, recientemente, la Gerencia de ventas, se percató que la empresa tiene una importante acumulación de inventario. Asimismo, se ha identificado una reducción en las búsquedas de un grupo importante de productos, lo que ha redundado en una disminución sustancial de sus ventas del último trimestre.

En el presente documento, se da a conocer el programa realizado el cual tiene la finalidad de mostrar un reporte de nivel básico, tomando en cuenta las validaciones de usuario necesarias para cuidar la seguridad de la información de la empresa, es decir, que no cualquier persona tenga acceso a la misma, a continuación se describe el flujo de lo que el sistema debe hacer de manera sencilla:

### **Login de usuario-administrador:**

- Al ejecutar el programa, solicita al usuario sus datos de acceso (nombre de usuario y contraseña) y permitir visualizar el reporte siempre que los datos sean correctos.
- Definir dentro del código el siguiente usuario que deberá tener permisos de administrador:
  - nombre de usuario: **emtech**
  - contraseña: **caso1**

### **Productos más vendidos y productos rezagados:**

- Generar un listado de los 5 productos con mayores ventas y uno con los 10 productos con mayor búsquedas.
- Por categoría, generar un listado con los 5 productos

## Definición de código

A continuación, se describe el código del sistema de generación de reporte en la empresa LifeStore:

### Importación del lifestore\_file

Para evitar robustez, se trabajó con el archivo con la información proporcionada por la tienda en la misma carpeta del proyecto, a su vez este fue importada desde el main para poder utilizar los elementos de las listas correspondientes.

```
#importar documento
import lifestore_file
```

### Login

Antes que nada, se definen las variables que almacenan la información del usuario administrador para poder generar los reportes de LifeStore, a su vez se declaran las variables de entrada (Userxtxt y Passtxt) , es decir, aquellas donde se almacenará la información de usuario y contraseña que el usuario teclee en consola.

```
#Definir usuario Administrador
User='emtech'
password='caso1'

#Capturar usuario y contraseña
print('=====LOGIN=====')
Userxtxt=input("Nombre de usuario: ")
Passtxt=input("Contraseña: ")
```

Se valida que el usuario y contraseña sean correctos, en caso que sea correcto, entonces mostrará los reportes, si no mostrará el siguiente mensaje”¡Usuario no permitido!, Intente de nuevo”.

```
if Userxtxt==User and Passtxt==password:
    print("=====REPORTES LIFESTORE=====")
```

Contraseña: caso1

REPORTES LIFESTORE

```
raith Stealth
```

```
print("¡Usuario no permitido!, Intente de nuevo")
```

Contraseña: 582q

¡Usuario no permitido!, Intente de nuevo

El mismo procedimiento se aplica para el reporte de los 10 productos con mejor ventas, solamente se valida que el tamaño de la lista temporal sea 10, para detener el proceso.

```

print('\n')
print("-----5 productos con mayores ventas-----")
#Variables de las listas auxiliares
IDList=[]
NombreLista=[]

#buscar los primeros 5 productos con valor de venta 5
for i in range(len(lifestore_file.lifestore_sales)):
    if lifestore_file.lifestore_sales[i][2] == 5:
        IDList.append(lifestore_file.lifestore_sales[i][1])
        if len(IDList) == 5:
            break;
#print(IDList)
#Buscar el nombre del id almacenado en la tmp IDList en la lista de los productos
for i in range(len(IDList)):
    for x in range(len(lifestore_file.lifestore_products)):
        if IDList[i] == lifestore_file.lifestore_products[x][0]:
            NombreLista=(lifestore_file.lifestore_products[x][1])
    print("-Producto "+str(i+1)+": "+ NombreLista)#Imprimir lista

```

```

print("-----10 productos con mayores ventas-----")
#Variables de las listas auxiliares
IDList10=[]
NombreLista10=[]

#buscar los primeros 10 productos con valor de venta 5
for i in range(len(lifestore_file.lifestore_sales)):
    if lifestore_file.lifestore_sales[i][2] == 5:
        IDList10.append(lifestore_file.lifestore_sales[i][1])
        if len(IDList10) == 10:
            break;
#print(IDList)
#Buscar el nombre del id almacenado en la tmp IDList en la lista delos productos
for i in range(len(IDList10)):
    for x in range(len(lifestore_file.lifestore_products)):
        if IDList10[i] == lifestore_file.lifestore_products[x][0]:
            NombreLista10=(lifestore_file.lifestore_products[x][1])
    print("-Producto "+str(i+1)+": "+ NombreLista10)#Imprimir lista

```

## Reporte de ventas menores

Para la generación de reportes con mayores ventas, el sistema primero busca en la lista `lifestore_sales` los primeros 5 productos con score de 1, es decir, la más baja y almacena su id en una lista temporal que se utilizará más tarde para buscar el nombre del producto; cuando el tamaño de la lista es igual a 5, para. Esto debido a que el primer reporte muestra únicamente los primeros 5 productos con mayores ventas; en seguida se consulta en la tabla `IDLista` (la temporal) el Id del producto para buscar el nombre del producto en la lista `lifestore_products` y va imprimiendo el nombre para mostrar el reporte.

```

print("-----5 productos con menores ventas-----")
#Variables de las listas auxiliares
IDListmenor=[]
NombreListamenor=[]

#buscar los primeros 5 productos con valor de venta 1
for i in range(len(lifystore_file.lifystore_sales)):
    if lifystore_file.lifystore_sales[i][2] == 1:
        IDListmenor.append(lifystore_file.lifystore_sales[i][1])
        if len(IDListmenor) == 5:
            break;
#print(IDListmenor)
#print(IDList)
#Buscar el nombre del id almacenado en la tmp IDList en la lista de los productos
for i in range(len(IDListmenor)):
    for x in range(len(lifystore_file.lifystore_products)):
        if IDListmenor[i] == lifystore_file.lifystore_products[x][0]:
            NombreListamenor=(lifystore_file.lifystore_products[x][1])
    print("-Producto "+str(i+1)+": "+ NombreListamenor)#Imprimir lista

```

## Solución al problema

Para la solución del problema presentado se realizó un análisis de los productos más vendidos y productos rezagados a partir del análisis de las categorías con menores ventas y categorías con menores búsquedas.

De la misma forma, también se validó la seguridad de la información de la tienda, es decir, se limitaron los permisos de acceso mediante un login.

Dado el análisis y los reportes generados mediante el sistema, se propone retirar del mercado los siguientes productos, por mencionar alguno :

```

-Producto 1: Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0
-Producto 2: Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
-Producto 3: Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
-Producto 4: Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
-Producto 5: Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
-Producto 6: Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
-Producto 7: Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
-Producto 8: Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel

```

En justificación a la anterior es debido a que son aquellos productos que menor venta tienen y por ende, solamente acumulan stock ya que tarda mucho en venderse o bien, no son los mejores productos.

## **Conclusión**

Para finalizar con el proyecto, se llegó a la conclusión que es necesario implementar herramientas que ayuden a visualizar está información de manera más intuitiva y a su vez tener opciones que le permitan al usuario interactuar con el sistema, es decir, permitir al usuario decir que hacer en cuanto a la generación de reportes se requiera.

Haciendo énfasis en el lenguaje de programación utilizado Python, es un gran lenguaje de programación, fácil de aprender, tomando en cuenta que su sintaxis es simple y concreta, lo cual permite que la interpretación del código sea elemental.