



EmTech Emerging Technologies Institute

Fundamentos de Programación con Python

“PROYECTO FINAL”

2da semana
Diciembre 2020

Integrantes:

Leobardo Julian Quiroga Lechuga

Tutor: Jaime Saúl Alonso Sánchez

Fecha de entrega: 11 de Diciembre de 2021

ÍNDICE

INTRODUCCIÓN.....	3
DEFINICIÓN DEL CÓDIGO	3
SOLUCIÓN AL PROBLEMA.....	6
CONCLUSIÓN	7

ÍNDICE DE FIGURAS

Ilustración 1: Log In del usuario (simple)	3
Ilustración 2: Filtro de Búsquedas	4
Ilustración 3: Ventas por mes de los mejores productos	4
Ilustración 4: Lista de mejores y peores reseñas	5
Ilustración 5: Ventas por mes (p.1)	5
Ilustración 6: Ventas por mes (p.2)	6
Ilustración 7: Ganancias por mes.....	6
Ilustración 8: Ganancias totales.....	6

INTRODUCCIÓN

Como cierre de bloque se realizará el análisis de datos mediante la programación en el lenguaje de Python. En el cual se presenta una situación de la cual LifeStore es una tienda virtual que maneja una amplia gama de artículos, en la cual se cuenta con una amplia cantidad de artículos en almacén, los cuales no generan ganancias. Además que se detectaron una disminución de búsquedas en ciertas categorías, haciendo proporcional una pérdida de ventas.

El proyecto consiste en generar un código que muestre los datos mensualmente para su análisis individual conforme a lo solicitado por la empresa, con el fin de radicalizar en una solución óptima en base a los datos que se recaudaran y mostrarán.

DEFINICIÓN DEL CÓDIGO

```
1  #Importamos las listas del archivo "lifestore_file" para no codificar debajo de la base de datos
2  from lifestore_file import lifestore_products, lifestore_searches, lifestore_sales
3  from collections import defaultdict
4
5  #Lo primero es realizar un login sencillo
6  #Los datos del usuario ya están precargados en la base de datos
7  USUARIO = 'Liontári'
8  CONTRASENA = 'Pok3m0n98'
9
10 #La cadena a continuación son condiciones a cumplir en dado caso que el usuario vierta mal sus datos
11 username = input('Ingrese su nombre de usuario:\n > ')
12 password = input('Ingrese la contraseña:\n > ')
13 if username == USUARIO:
14     if password == CONTRASENA:
15         print("Buen día! Bienvenido al programa, Jimmy")
16     #Comparando la base de datos con lo que el usuario ingrese
17     else:
18         print("Contraseña errónea")
19 else:
20     print("El usuario no existe")
21 #En dado caso de ser erróneo, se regresa un mensaje correspondiente y se detiene el programa
```

Ilustración 1: Log In del usuario (simple)

```
#En este bloque se busca el # de productos buscados
#Se suman a una lista vacía
mayores_búsquedas = []
for search in lifestore_searches:
    id_search=search[0]
    prod_search=search[1]
    mayores_búsquedas.append(prod_search)
def leaders(xs, top=15):
    counts = defaultdict(int)
    for x in xs:
        counts[x] += 1
    return sorted(counts.items(), reverse=True, key=lambda tup: tup[1][:top])
#Finamente se hace un conteo de las veces que se encuentra cada "ID" en la lista
zs = list(mayores_búsquedas)
print("Mayores búsquedas")
print("(Id_Producto, # Búsquedas)")
print(leaders(zs)[0:10])
print("Menores búsquedas")
print("(Id_Producto, # Búsquedas)")
print(leaders(zs)[-6:-1])
```

Ilustración 2: Filtro de Búsquedas

```
#Se solicita al usuario seleccionar un mes para mostrarle resultados del mes
num_mes=int(input("Inserta el número de mes: "))
meses=["/01/", "/02/", "/03/", "/04/", "/05/", "/06/", "/07/", "/08/", "/09/", "/10/", "/11/", "/12/"]
ventas_mes = []
#Se crea una lista vacía que se llenará repetitivamente con el ID
#en base al número de veces que aparece en el mes
for venta in lifestore_sales:
    fecha_venta = venta[3]
    id_venta=venta[1]
    if meses[num_mes-1] in fecha_venta:
        ventas_mes.append(id_venta)
#Los ID se cuentan y se suman respectivamente, para mostrar el TOP al final del bloque
def leaders(xs, top=10):
    counts = defaultdict(int)
    for x in xs:
        counts[x] += 1
    return sorted(counts.items(), reverse=True, key=lambda tup: tup[1][:top])
xs = list(ventas_mes)
print("(Id_Producto, # Ventas en el mes)")
print(leaders(xs)[0:5])
```

Ilustración 3: Ventas por mes de los mejores productos

```

66 score_mejores=[]
67 score_peores=[]
68 #Creamos 2 listas vacías respecto a las reseñas
69 for venta in lifestore_sales:
70     score=venta[2]
71     id_venta=venta[1]
72     if score == 5:
73         score_mejores.append(id_venta)
74     else:
75         score == 1
76         score_peores.append(id_venta)
77 #Condicionamos la búsqueda en base al mejor y peor reseña (5 y 1) para agregar la ID a la lista vacía
78 #Se cuenta cada una de las listas para mostrar los ID de los productos con mejor y peor reseña
79 def leaders(xs, top=10):
80     counts = defaultdict(int)
81     for x in xs:
82         counts[x] += 1
83     return sorted(counts.items(), reverse=True, key=lambda tup: tup[1][:top])
84 xs = list(score_mejores)
85 print("Id_Producto, Veces con score de 5")
86 print(leaders(xs)[0:5])
87 def leaders(xs, top=10):
88     counts = defaultdict(int)
89     for x in xs:
90         counts[x] += 1
91     return sorted(counts.items(), reverse=True, key=lambda tup: tup[1][:top])
92 xs = list(score_peores)
93 print("Id_Producto, Veces con score de 1")
94 print(leaders(xs)[0:5])

```

Ilustración 4: Lista de mejores y peores reseñas

```

96 #Creación de una lista vacía a la cual se le añadirán las ventas
97 #Condicionada si el producto tuvo reembolso o no
98 ventas = []
99 for sale in lifestore_sales:
100     refund = sale[4]
101     if refund == 1:
102         continue
103     else:
104         ventas.append(sale)
105
106 #Creación de una lista con los meses a buscar
107 #Se añade una lista vacía dentro de la lista principal para segregar los valores obtenidos
108 meses = [
109     '/01/', '/02/', '/03/', '/04/', '/05/', '/06/',
110     '/07/', '/08/', '/09/', '/10/', '/11/', '/12/'
111 ]
112 ventas_por_mes = []
113 for mes in meses:
114     lista_vacia = []
115     ventas_por_mes.append(lista_vacia)

```

Ilustración 5: Ventas por mes (p.1)

```

117 #Se buscan los datos de la venta
118 for venta in ventas:
119     id_venta = venta[0]
120     fecha = venta[3]
121 #Se clasifican los datos obtenidos y guardados en los meses buscados
122     contador_de_mes = 0
123     for mes in meses:
124         if mes in fecha:
125             ventas_por_mes[contador_de_mes].append(id_venta)
126             continue
127         contador_de_mes = contador_de_mes + 1
128 #Se realiza un contador de las ventas buscadas en cada mes, y se cuenta el tamaño de la lista
129     contador_de_mes = 0
130     for venta_mensual in ventas_por_mes:
131         print(f'En el mes de {meses[contador_de_mes]} hubo {len(venta_mensual)} ventas!')
132         contador_de_mes = contador_de_mes + 1
133

```

Ilustración 6: Ventas por mes (p.2)

```

#Creación de la lista vacía de ganancias
#De la cuál se irá buscando el ID de la venta para relacionarla con el precio de cada producto
#Se suman las ganancias por cada ID encontrado en el mes
gancias_mensuales = []
for venta_mensual in ventas_por_mes:
    ganancia_del_mes = 0
    for id_venta in venta_mensual:
        indice_de_venta = id_venta - 1
        info_de_venta = lifestore_sales[indice_de_venta]
        id_prod = info_de_venta[1]
        indice_de_prod = id_prod - 1
        info_del_prod = lifestore_products[indice_de_prod]
        precio_de_prod = info_del_prod[2]
        ganancia_del_mes = ganancia_del_mes + precio_de_prod
    gancias_mensuales.append(ganancia_del_mes)
print("Ganancias por mes: ")
print(gancias_mensuales)

```

Ilustración 7: Ganancias por mes

```

#Se suman los datos de ganancia de cada mes obtenidos previamente con una función
total_anual=sum(gancias_mensuales)
print(f"El total de ganancias en el año fue de ${total_anual}: ")

```

Ilustración 8: Ganancias totales

SOLUCIÓN AL PROBLEMA

Comenzaríamos por remover del almanaque aquellos productos que coincidan con una venta baja, baja puntuación y/o baja reseña, independientemente si el producto tuvo buenas reseñas

o puntuación, pero obtuvo un bajo número de ventas es algo a considerar para removerlo de la lista de productos a ofrecer, para remover estos productos se podría hacer una oferta al mercado con un costo reducido, no necesariamente para obtener un margen de ganancia por producto a lo habitual, si no mucho menor, solo con el fin de quedar “tablas” al momento de venderlos y así reducir inventario para no volver a comprar/fabricar dichos productos.

CONCLUSIÓN

Fue un reto realizar la programación de este proyecto, disfrute los obstáculos y el aprendizaje que deja. Es un proyecto que tiene un gran margen de mejora, sin duda alguna. Pero logré comprender de una mejor manera la lógica detrás de un código y las diferentes maneras que hay para resolver un mismo punto.

Logré cumplir los objetivos del proyecto, de la mejor manera que se me ocurrió. Es algo “rústico” la manera que expresé los resultados. Espero mejoras en lo que hago.