

Santander - EMTECH (Emerging Technologies Institute)

Proyecto 1 - Introducción a Python - Tienda Virtual LifeStore

Pérez Ruiz Daniel Michell
Domingo 06 de septiembre 2020.

Resumen

En el siguiente proyecto se pone en práctica las bases de programación de Python, se realiza el estudio de la tienda virtual llamada **LifeStore**, la cual maneja artículos de carácter electrónico y se realiza un análisis de datos sobre su inventario, donde se obtienen los productos más vendidos, los más rezagados, los más y menos buscados, igual se realiza un análisis por categoría y se describe las ventas realizadas por mes, así como los ingresos obtenidos.

Índice

1. Introducción	1
2. Definición del Código	2
3. Solución al problema	16
4. Conclusión.	16

1. Introducción

No es un secreto que una gran cantidad de compañías hoy en día sumen a sus filas a científicos de datos para poder obtener un aprovechamiento de los datos que reciben. Cada día a cada instante estamos bombardeados por diferentes datos, la hora en que despertamos, las cosas que compramos, las cosas que comemos, al dar permisos a alguna aplicación sobre el uso de nuestros datos como son nombre, edad, gustos, etc.

El poder aprovechar estos datos de la mejor manera para un objetivo específico es lo que hace a un científico de datos, vital actualmente. En este proyecto se trabaja con una tienda virtual llamada **LifeStore**, la cual requiere el apoyo de científicos de datos para poder aprovechar de mejor manera una cantidad grande de datos que ha ido recibiendo.

A nosotros como futuros científicos de datos se nos pide realizar dicho análisis, donde se obtenga la información de los productos más vendidos, así como sus productos más rezagados. Esto a través de las ventas que se han realizado en sus productos, así como en sus búsquedas en la página de la tienda.

Una vez completado ese aspecto se nos pide obtener las categorías de los productos y nuevamente hacer un listado con las mayores ventas y las mayores búsquedas realizadas por los usuarios.

La página recibe reseñas de sus productos, por lo cual también se implementa un listado con los productos con mejores reseñas y con peores, para así tener en cuenta la razón de la cual hay productos con mayores ventas.

Finalmente como en cualquier inventario, se nos pide obtener el listado con las ventas realizadas cada mes, así como los ingresos mensuales, con esto se obtiene finalmente las ventas realizadas en el año, algo a aclarar es que como el año aún no está concluido, habrá meses en los que aún no se hallan realizado ventas ni obtenido ingresos.

2. Definición del Código

El código comienza solicitando un número de usuario y una contraseña, para esto se utiliza un input y utilizamos el método `getpass()` para que la contraseña no se pueda observar, así se muestre de una forma más profesional.

```
1 import lifestore_file as lf          # Importamos los datos
2 import getpass, os                  # Para poner una contraseña y limpiar la
   pantalla
3
4 #Buscamos mayores ventas, creamos una lista vacía donde se guarden las mayores
   ventas
5 #lifestore_searches = [id_search, id product]
6 #lifestore_sales = [id_sale, id-product, score (from 1 to 5), date, refund (1 for
   true or 0 to false)]
7 #lifestore_products = [id-product, name, price, category, stock (en existencia)]
8
9 input('Escribe tu nombre de usuario: ')      #Ingresa el LOGIN
10 password = getpass.getpass('Contraseña: ***** ')
11 os.system ("cls")
```

Se da un bienvenida a la base de datos de **LifeStore** y se mencionan los apartados que se mostrarán: productos más vendidos, más buscados, así como su respectiva categoría, los productos con mejores reseñas y los ingresos y ventas percibidos por mes.

```
1
2 #Mensaje de bienvenida
3 print('''Bienvenido a la base de datos de LifeStore, debido a un analisis de los
   productos hemos tenido que hacer un ajuste en nuestra configuracion. \n
4   A continuacion se desplegarán 3 apartados:\n
5   Primero se muestran los productos mas vendidos y los productos rezagados, esto
   a traves de un listado con las mayores ventas y mayores busquedas:
6   Tambien se muestra por categoria
7   Segundo se muestran los productos por resenha, los productos con mejores
   resenhas y con peores.
8   Finalmente se muestran los ingresos totales y las ventas promedio mensual.
9   Estas listo?
10   ''')
11
12
13
14 input('Se mostrará a continuación el primer punto, los productos más vendidos y más
   buscados, presiona Enter:')
15
16 os.system ("cls")
```

Una vez dada una bienvenida se comienza el código serio, se realizan listas de las ventas de los productos así como de sus búsquedas, esto a partir de ciclos for y un condicional if.

```

1 #####
2
3 greater_sales = []                                #Lista vacia que almacenará las mayores
   ventas
4 for i in range(1,len(lf.lifestore_products)+1):
5
6 #Lo buscamos por su tamaño por qué nos devolvera el número de listas , mas no los
   elementos de cada lista , se empieza en 1, por qué no tenemos índice 0
7
8
9 contador_greater_sales = 0                        #Un contador que anhadiremos a nuestra
   lista
10 for j in lf.lifestore_sales:                      #Iteramos sobre la lista de ventas
11     if i == j[1]:
12
13     #Si el producto en la lista de ventas es igual al producto en la lista de
       productos agregamos 1 al contador
14
15     contador_greater_sales +=1
16 greater_sales.append([i,contador_greater_sales])  #Guardamos el producto y el
   contador
17
18
19 #####
20
21 #Realizamos lo mismo con las búsquedas
22
23 greater_searches = []
24 for i in range(1,len(lf.lifestore_products)+1):
25     contador_greater_searches = 0
26     for j in lf.lifestore_searches:
27         if i == j[1]:
28             contador_greater_searches +=1
29 greater_searches.append([i,contador_greater_searches])
30
31 #print(greater_sales) #Este es más de apoyo para mi para visualizar lo realizado
32 #print(greater_searches)

```

Después se crean dos pequeñas funciones para utilizar Sort(), para que se acomoden en orden descendente, así obtenemos las mayores ventas y los más buscados.

```

1
2     #
   #####
3
4
5 #Ahora vamos a acomodar en orden descendente las listas obtenidas , realizamos una
   pequeña función para hacerlo , sólo para acomodar el sort
6 def Sort_Sales(element):
7     return element[1]
8
9 greater_sales.sort(key = Sort_Sales , reverse =True)    #La lista se acomoda en
   orden descendente
10 #print(greater_sales)
11 worst_sales = greater_sales.copy()                    #Realizamos una copia para las
   peores ventas

```

```

12 worst_sales.sort(key = Sort_Sales) #Acomodamos en orden ascendente
13
14 #
15 #####
16 #Realizamos lo mismo con las búsquedas
17 def Sort_Searches(element):
18     return element[1]
19 greater_searches.sort(key = Sort_Searches , reverse =True)
20 #print(greater_searches)
21 worst_searches = greater_searches.copy()
22 worst_searches.sort(key = Sort_Searches)
23

```

Se imprimen los resultados de los más vendidos y los más buscados (así como los menos vendidos y los menos buscados), se imprimen menos de 50, para no acaparar en datos.

```

1 #
2 #####
3 #Imprimimos los resultados obtenidos
4 #Vamos a cambiar esto: Mostramos los primeros 10 más vendidos , ya que se pierde
5   detalle al mostrar tantos
6 input('Los 10 productos más vendidos son: (presiona Enter) \n')
7 print(' ***** LOS MÁS VENDIDOS *****\n \n')
8
9 for i in range(len(greater_sales[0:10])): #Mostramos los 10 primeros
10     print(lf.lifstore_products[greater_sales[i][0]][1] ,'\n') #Para eso
11     mandamos a llamar a la lista de productos
12
13
14 #### Realizamos lo mismo para los menos vendidos
15 input('Los menos vendidos son: (presiona Enter) \n')
16 print(' ***** LOS MENOS VENDIDOS ***** \n \n')
17 #print('Los menos vendidos son:')
18 for i in range(len(greater_sales[0:10])):
19     print(lf.lifstore_products[worst_sales[i][0]][1] ,'\n')
20
21 #
22 #####
23
24 input('Los más buscados son: (presiona Enter) \n')
25 print(' ***** LOS MÁS BUSCADOS *****\n \n')
26 for i in range(len(worst_searches[0:10])):
27     print(lf.lifstore_products[greater_searches[i][0]][1] ,'\n')
28
29 input('Los menos buscados son: (presiona Enter) \n')
30 print(' ***** LOS MENOS BUSCADOS *****\n \n')
31 for i in range(len(worst_searches[0:10])):
32     print(lf.lifstore_products[worst_searches[i][0]][1] ,'\n')
33
34 #
35 #####

```

Una vez finalizado este proceso se empieza a trabajar con las categorías, hacemos una lista de todas las categorías, realizamos for para eliminar las repeticiones y quedarnos sólo con una de cada categoría.

```

1 #Dado que no siempre nuestra base de datos será tan pequeña, es conveniente
  resivar cuantos tipos de categorías hay en lifestore_file
2 #Y con eso hacer una lista con los valores no repetidos de las categorías.
3 #Vamos a realizar una lista con todas las categorías y una más con las categorías
  sin repetición
4 category = []
5 category_unique = []
6
7 for i in range(1, len(lf.lifestore_products)):      #Hacemos un for sobre el
  tamaño de los productos para revisar a que categoría pertenece
8 #print(lf.lifestore_products[i][3])
9 category.append(lf.lifestore_products[i][3])        #Agregamos las categorías
10
11 for i in category:                                  #Nos quedamos con las categorías únicas
12     if i not in category_unique:
13         category_unique.append(i)
14
15 #print(category_unique)

```

Hacemos una lista donde identificamos los productos que se encuentran en dicha categoría esto a partir de ciclos for y un condicional if, este proceso se realiza varias veces a lo largo del programa:

```

1
2     #
3     #####
4 #Identificamos los elementos que se encuentran en la misma categoría
5 category_same = []                                #Creamos una lista donde guardaremos los
  elementos que se encuentran en la misma categoría
6 for i in range(len(category_unique)):              #Recorremos las categorías únicas
7     lista_category_same = []                        #En vez de ir enumerando como anteriormente
  lo hacíamos, vamos ir creando listas de listas
8     for j in lf.lifestore_products:                #Recorremos los productos
9         if category_unique[i] == j[3]:             #Si nuestra categoría concuerda con el
  categoría del producto guardamos
10         lista_category_same.append(j[0])
11     category_same.append(lista_category_same)       #La guardamos en la lista
  category_same
12 #print(category_same)
13
14     #
15     #####

```

Analizamos cuantas ventas tuvo cada categoría, así como cuantas búsquedas, esto como ya se mencionó anteriormente a partir de ciclos for y un if.

```

1 #####
2
3
4 #Vamos a ver cuantos ventas tuvo cada categoria
5
6 category_sales=[]                                #Vamos a ver cuantas ventas tuvo cada categor
  ía

```

```

7  for i in category_same :                #Recorremos la lista anteriormente creada
8      lista_category_sales =[]            #Nuevamente crearemos una lista de listas
9      for j in range(len(i)):            #De igual forma se crea una lista vacia
10     para cada ciclo
11         for k in greater_sales:          #Recorremos la lista de las ventas
12             if i[j]==k[0]:              #Si los indices concuerdan se guarda en
13             ventas por categoría
14             lista_category_sales.append(k)
15             category_sales.append(lista_category_sales)    #La guardamos en la lista
16             category_sales
17
18     #
19     #####
20 category_searches=[]                    #se hace la misma operacion para la seccion
21 de busquedas
22 for i in category_same:
23     lista_category_searches =[]
24     for j in range(len(i)):
25         for k in greater_searches:
26             if i[j]==k[0]:
27                 lista_category_searches.append(k)
28                 category_searches.append(lista_category_searches)
29
30     #
31     #####

```

Realizamos listas de cada categoría, de las cuales obtendremos las mejores ventas, así como las mejores búsquedas y también su contraparte. Las acomodamos con sort nuevamente, realizamos una copia, la cual se utiliza para la contraparte.

```

1      #
2      #####
3
4  input('Se mostrará a continuación los productos más vendidos por categoría, sus
5      ventas y cuantas busquedas se realizaron , presiona Enter: \n')
6
7  os.system ("cls")
8
9  input('Listo para ver las ventas y busquedas por categoría? (Presiona Enter): \n')
10
11
12     #
13     #####
14 #Hacemos una lista para cada categoría, mostraremos los productos más vendidos y
15     menos vendidos por caterogoría
16 procesadores_sales = category_sales[0]
17 #print(procesadores_sales)
18 tarjetas_de_video_sales = category_sales[1]

```

```

18 tarjetas_madre_sales = category_sales[2]
19 discos_duros_sales = category_sales[3]
20 memorias_usb_sales = category_sales[4]
21 pantallas_sales = category_sales[5]
22 bocinas_sales = category_sales[6]
23 audifonos_sales = category_sales[7]
24
25 procesadores_searches = category_searches[0]
26 tarjetas_de_video_searches = category_searches[1]
27 tarjetas_madre_searches = category_searches[2]
28 discos_duros_searches = category_searches[3]
29 memorias_usb_searches = category_searches[4]
30 pantallas_searches = category_searches[5]
31 bocinas_searches = category_searches[6]
32 audifonos_searches = category_searches[7]
33
34
35 #
36 #####
37 #Realizamos una función sort, en la cual se acomoda de forma descendente cada
38 #categoría, además se crean copias
39 #para aplicarlas posteriormente en las peores ventas o búsquedas
40
41 def Sort(element):
42     return element[1]
43
44 procesadores_sales.sort(key = Sort, reverse =True)
45 procesadores_searches.sort(key = Sort, reverse =True)
46 procesadores_sales_worst = procesadores_sales.copy()
47 procesadores_sales_worst.sort(key = Sort)
48 procesadores_searches_worst = procesadores_searches.copy()
49 procesadores_searches_worst.sort(key = Sort)
50
51 tarjetas_de_video_sales.sort(key = Sort, reverse =True)
52 tarjetas_de_video_searches.sort(key = Sort, reverse =True)
53 tarjetas_de_video_sales_worst = tarjetas_de_video_sales.copy()
54 tarjetas_de_video_sales_worst.sort(key = Sort)
55 tarjetas_de_video_searches_worst = tarjetas_de_video_searches.copy()
56 tarjetas_de_video_searches_worst.sort(key = Sort)
57
58 tarjetas_madre_sales.sort(key = Sort, reverse =True)
59 tarjetas_madre_searches.sort(key = Sort, reverse =True)
60 tarjetas_madre_sales_worst = tarjetas_madre_sales.copy()
61 tarjetas_madre_sales_worst.sort(key = Sort)
62 tarjetas_madre_searches_worst = tarjetas_madre_searches.copy()
63 tarjetas_madre_searches_worst.sort(key = Sort)
64
65 discos_duros_sales.sort(key = Sort, reverse =True)
66 discos_duros_searches.sort(key = Sort, reverse =True)
67 discos_duros_sales_worst = discos_duros_sales.copy()
68 discos_duros_sales_worst.sort(key = Sort)
69 discos_duros_searches_worst = discos_duros_searches.copy()
70 discos_duros_searches_worst.sort(key = Sort)

```

```

71 memorias_usb_sales.sort(key = Sort, reverse =True)
72 memorias_usb_searches.sort(key = Sort, reverse =True)
73 memorias_usb_sales_worst = memorias_usb_sales.copy()
74 memorias_usb_sales_worst.sort(key = Sort)
75 memorias_usb_searches_worst = memorias_usb_searches.copy()
76 memorias_usb_searches_worst.sort(key = Sort)
77
78 pantallas_sales.sort(key = Sort, reverse =True)
79 pantallas_searches.sort(key = Sort, reverse =True)
80 pantallas_sales_worst = pantallas_sales.copy()
81 pantallas_sales_worst.sort(key = Sort)
82 pantallas_searches_worst = pantallas_searches.copy()
83 pantallas_searches_worst.sort(key = Sort)
84
85 bocinas_sales.sort(key = Sort, reverse =True)
86 bocinas_searches.sort(key = Sort, reverse =True)
87 bocinas_sales_worst = bocinas_sales.copy()
88 bocinas_sales_worst.sort(key = Sort)
89 bocinas_searches_worst = bocinas_searches.copy()
90 bocinas_searches_worst.sort(key = Sort)
91
92 audifonos_sales.sort(key = Sort, reverse =True)
93 audifonos_searches.sort(key = Sort, reverse =True)
94 audifonos_sales_worst = audifonos_sales.copy()
95 audifonos_sales_worst.sort(key = Sort)
96 audifonos_searches_worst = audifonos_searches.copy()
97 audifonos_searches_worst.sort(key = Sort)
98
99
100 #
#####

```

Una vez obtenidas las variables anteriores, se imprimen las mejores y peores, esto a través de un ciclo for en la lista de las ventas y los productos.

```

1 #
#####
2
3
4 #print(procesadores)
5 print('Procesadores, se muestran los 2 más vendidos y los 2 menos vendidos \n')
6 print('Las mejores ventas por categoria en procesadores fueron: \n')
7 for i in range(len(procesadores_sales[0:2])):
8     print('El producto ', lf.lifstore_products[procesadores_sales[i][0]-1][1], 'tuvo '
9         ,procesadores_sales[i][1], 'ventas \n')
9 print('Las peores ventas por categoria en procesadores fueron: \n')
10 for i in range(len(procesadores_sales_worst[0:2])):
11     print('El producto ', lf.lifstore_products[procesadores_sales_worst[i][0]-1][1],
12         'tuvo',procesadores_sales_worst[i][1], 'ventas \n')
12
13 input('***** Presiona Enter para continuar *****
14 \n')
14
15 print('Tarjetas de vídeo, se muestran los 2 más vendidos y los 2 menos vendidos \n')
16 print('Las mejores ventas por categoria en tarjetas de vídeo fueron: \n')

```



```

17 for i in range(len(tarjetas_de_video_sales[0:2])):
18     print('El producto ', lf.lifestore_products[tarjetas_de_video_sales[i][0]-1][1], '
        tuvo', tarjetas_de_video_sales[i][1], 'ventas \n')
19 print('Las peores ventas por categoria en tarjetas de video fueron: \n')
20 for i in range(len(tarjetas_de_video_sales_worst[0:2])):
21     print('El producto ', lf.lifestore_products[tarjetas_de_video_sales_worst[i
        ][0]-1][1], 'tuvo', tarjetas_de_video_sales_worst[i][1], 'ventas \n')
22
23 input('***** Presiona Enter para continuar *****
        \n')
24
25 print('Tarjetas madre, se muestran los 2 más vendidos y los 2 menos vendidos \n')
26 print('Las mejores ventas por categoria en tarjetas madre fueron: \n')
27 for i in range(len(tarjetas_madre_sales[0:2])):
28     print('El producto ', lf.lifestore_products[tarjetas_madre_sales[i][0]-1][1], '
        tuvo', tarjetas_madre_sales[i][1], 'ventas \n')
29 print('Las peores ventas por categoria en tarjetas madre fueron: \n')
30 for i in range(len(tarjetas_madre_sales_worst[0:2])):
31     print('El producto ', lf.lifestore_products[tarjetas_madre_sales_worst[i
        ][0]-1][1], 'tuvo', tarjetas_madre_sales_worst[i][1], 'ventas \n')
32
33 input('***** Presiona Enter para continuar *****
        \n')
34
35 print('Discos duros, se muestran los 2 más vendidos y los 2 menos vendidos \n')
36 print('Las mejores ventas por categoria en discos duros fueron: \n')
37 for i in range(len(discos_duros_sales[0:2])):
38     print('El producto ', lf.lifestore_products[discos_duros_sales[i][0]-1][1], 'tuvo'
        , discos_duros_sales[i][1], 'ventas \n')
39 print('Las peores ventas por categoria en discos duros fueron: \n')
40 for i in range(len(discos_duros_sales_worst[0:2])):
41     print('El producto ', lf.lifestore_products[discos_duros_sales_worst[i][0]-1][1],
        'tuvo', discos_duros_sales_worst[i][1], 'ventas \n')
42
43 input('***** Presiona Enter para continuar *****
        \n')
44
45 print('Memorias USB, se muestran los 2 más vendidos y los 2 menos vendidos \n')
46 print('Las mejores ventas por categoria en Memorias USB fueron: \n')
47 for i in range(len(memorias_usb_sales[0:2])):
48     print('El producto ', lf.lifestore_products[memorias_usb_sales[i][0]-1][1], 'tuvo'
        , memorias_usb_sales[i][1], 'ventas \n')
49 print('Las peores ventas por categoria en Memorias USB fueron: \n')
50 for i in range(len(memorias_usb_sales_worst[0:2])):
51     print('El producto ', lf.lifestore_products[memorias_usb_sales_worst[i][0]-1][1],
        'tuvo', memorias_usb_sales_worst[i][1], 'ventas \n')
52
53 input('***** Presiona Enter para continuar *****\n
        n')
54
55
56 print('Pantallas, se muestran los 2 más vendidos y los 2 menos vendidos \n')
57 print('Las mejores ventas por categoria en Pantallas fueron: \n')
58 for i in range(len(pantallas_sales[0:2])):
59     print('El producto ', lf.lifestore_products[pantallas_sales[i][0]-1][1], 'tuvo',

```

```

    pantallas_sales[i][1], 'ventas \n')
60 print('Las peores ventas por categoria en Pantallas fueron: \n')
61 for i in range(len(pantallas_sales_worst[0:2])):
62     print('El producto ', lf.lifestore_products[pantallas_sales_worst[i][0]-1][1], '
        tuvo', pantallas_sales_worst[i][1], 'ventas \n')
63
64
65 input('***** Presiona Enter para continuar *****
    \n')
66
67
68 print('Bocinas, se muestran los 2 más vendidos y los 2 menos vendidos \n')
69 print('Las mejores ventas por categoria en bocinas fueron: \n')
70 for i in range(len(bocinas_sales[0:2])):
71     print('El producto ', lf.lifestore_products[bocinas_sales[i][0]-1][1], 'tuvo',
        bocinas_sales[i][1], 'ventas \n')
72 print('Las peores ventas por categoria en Pantallas fueron: \n')
73 for i in range(len(bocinas_sales_worst[0:2])):
74     print('El producto ', lf.lifestore_products[bocinas_sales_worst[i][0]-1][1], 'tuvo
        ', bocinas_sales_worst[i][1], 'ventas \n')
75
76
77
78
79
80 #####
81
82
83 input('Se mostrará a continuación los productos más buscados por categoría,
    presiona Enter: \n')
84
85 os.system ("cls")
86
87 input('Presiona Enter para contunuar y ver las busquedas por categoría: \n')
88
89
90 print('Procesadores, se muestran los 2 más buscados y los 2 menos buscados \n')
91 print('Los más buscados por categoria en procesadores fueron: \n')
92 for i in range(len(procesadores_searches[0:2])):
93     print('El producto ', lf.lifestore_products[procesadores_searches[i][0]-1][1], '
        tuvo', procesadores_searches[i][1], 'busquedas \n')
94 print('Los menos buscados por categoria en procesadores fueron: \n')
95 for i in range(len(procesadores_searches_worst[0:2])):
96     print('El producto ', lf.lifestore_products[procesadores_sales_worst[i][0]-1][1],
        'tuvo', procesadores_searches_worst[i][1], 'busquedas \n')
97
98 input('***** Presiona Enter para continuar *****
    \n')
99
100 print('Tarjetas de vídeo, se muestran los 2 más buscados y los 2 menos buscados \n'
    )
101 print('Los más buscados por categoria en tarjetas de vídeo fueron: \n')
102 for i in range(len(tarjetas_de_video_searches[0:2])):
103     print('El producto ', lf.lifestore_products[tarjetas_de_video_searches[i
        ][0]-1][1], 'tuvo', tarjetas_de_video_searches[i][1], 'busquedas \n')

```

```

104 print('Los menos buscados por categoria en tarjetas de vídeo fueron: \n')
105 for i in range(len(tarjetas_de_video_searches_worst[0:2])):
106     print('El producto ', lf.lifestore_products[tarjetas_de_video_searches_worst[i][0]-1][1], 'tuvo', tarjetas_de_video_searches_worst[i][1], 'busquedas \n')
107
108 input('***** Presiona Enter para continuar *****\n')
109
110
111 print('Tarjetas madre, se muestran los 2 más buscados y los 2 menos buscados \n')
112 print('Los más buscados por categoria en tarjetas madre fueron: \n')
113 for i in range(len(tarjetas_madre_searches[0:2])):
114     print('El producto ', lf.lifestore_products[tarjetas_madre_searches[i][0]-1][1], 'tuvo', tarjetas_madre_searches[i][1], 'busquedas \n')
115 print('Los menos buscados por categoria en tarjetas madre fueron: \n')
116 for i in range(len(tarjetas_madre_searches_worst[0:2])):
117     print('El producto ', lf.lifestore_products[tarjetas_madre_searches_worst[i][0]-1][1], 'tuvo', tarjetas_madre_searches_worst[i][1], 'busquedas \n')
118
119
120 input('***** Presiona Enter para continuar *****\n')
121
122
123 print('Discos duros, se muestran los 2 más vendidos y los 2 menos vendidos \n')
124 print('Las mejores ventas por categoria en discos duros fueron: \n')
125 for i in range(len(discos_duros_searches[0:2])):
126     print('El producto ', lf.lifestore_products[discos_duros_searches[i][0]-1][1], 'tuvo', discos_duros_searches[i][1], 'ventas \n')
127 print('Las peores ventas por categoria en discos duros fueron: \n')
128 for i in range(len(discos_duros_searches_worst[0:2])):
129     print('El producto ', lf.lifestore_products[discos_duros_searches_worst[i][0]-1][1], 'tuvo', discos_duros_searches_worst[i][1], 'ventas \n')
130
131
132
133 input('***** Presiona Enter para continuar *****\n')
134
135
136 print('Memorias USB, se muestran los 2 más vendidos y los 2 menos vendidos \n')
137 print('Las mejores ventas por categoria en Memorias USB fueron: \n')
138 for i in range(len(memorias_usb_searches[0:2])):
139     print('El producto ', lf.lifestore_products[memorias_usb_searches[i][0]-1][1], 'tuvo', memorias_usb_searches[i][1], 'ventas \n')
140 print('Las peores ventas por categoria en Memorias USB fueron: \n')
141 for i in range(len(memorias_usb_searches_worst[0:2])):
142     print('El producto ', lf.lifestore_products[memorias_usb_searches_worst[i][0]-1][1], 'tuvo', memorias_usb_searches_worst[i][1], 'ventas \n')
143
144
145
146 input('***** Presiona Enter para continuar *****\n')
147

```

```

148
149 print('Pantallas , se muestran los 2 más vendidos y los 2 menos vendidos \n')
150 print('Las mejores ventas por categoria en pantallas fueron: \n')
151 for i in range(len(pantallas_searches[0:2])):
152     print('El producto ', lf.lifestore_products[pantallas_searches[i][0]-1][1], 'tuvo '
        , pantallas_searches[i][1], 'ventas \n')
153 print('Las peores ventas por categoria en pantallas fueron: \n')
154 for i in range(len(pantallas_searches_worst[0:2])):
155     print('El producto ', lf.lifestore_products[pantallas_searches_worst[i][0]-1][1],
        'tuvo', pantallas_searches_worst[i][1], 'ventas \n')
156
157
158
159 input('***** Presiona Enter para continuar *****
        \n')
160
161
162 print('Bocinas , se muestran los 2 más vendidos y los 2 menos vendidos \n')
163 print('Las mejores ventas por categoria en bocinas fueron: \n')
164 for i in range(len(bocinas_searches[0:2])):
165     print('El producto ', lf.lifestore_products[bocinas_searches[i][0]-1][1], 'tuvo ',
        bocinas_searches[i][1], 'ventas \n')
166 print('Las peores ventas por categoria en bocinas fueron: \n')
167 for i in range(len(bocinas_searches_worst[0:2])):
168     print('El producto ', lf.lifestore_products[bocinas_searches_worst[i][0]-1][1], '
        tuvo', bocinas_searches_worst[i][1], 'ventas \n')
169
170
171
172 input('***** Presiona Enter para continuar *****
        \n')
173
174
175 print('Audifonos , se muestran los 2 más vendidos y los 2 menos vendidos \n')
176 print('Las mejores ventas por categoria en audifonos fueron: \n')
177 for i in range(len(audifonos_searches[0:2])):
178     print('El producto ', lf.lifestore_products[audifonos_searches[i][0]-1][1], 'tuvo '
        , audifonos_searches[i][1], 'ventas \n')
179 print('Las peores ventas por categoria en audifonos fueron: \n')
180 for i in range(len(audifonos_searches_worst[0:2])):
181     print('El producto ', lf.lifestore_products[audifonos_searches_worst[i][0]-1][1],
        'tuvo', audifonos_searches_worst[i][1], 'ventas \n')
182
183 input('***** Presiona Enter para continuar *****
        \n')
184
185
186 input('Se mostrará a continuación los productos por resenha , las mejores resenhas y
        las peores , presiona Enter: \n')
187
188 os.system ("cls")
189
190 input('Presiona Enter para contunuar y ver las mejores y peores resenhas: \n')
191 #####

```

Se comienza a trabajar con las reseñas, esto es un poco más simple que lo anterior, se realizan dos

for, uno sobre la lista de productos y otro sobre la lista de ventas y de ahí se guardan en listas, sabiendo si puntuación y si fueron devueltas o no.

```
1 #####
2 #####
3
4 #Realizamos una lista vacía donde se almacenarán las reseñas
5
6 review = []
7 for i in range(1, len(lf.lifstore_products)+1):    #De manera similar a lo
    realizado anteriormente se realiza con ciclo for
8     review_score = []                            #Lista para la puntuación de la reseña
9     review_refund = []                          #Lista para si fue devuelta o no
10    for j in lf.lifstore_sales:                  #Iteramos sobre las ventas
11        if i==j[1]:                             #Si cumplen la igualdad
12            review_score.append(j[2])            #Esta variable guarda las reseñas para
            cada producto
13            review_refund.append(j[4])           #Guardar si el producto fue regresado o no.
14            #Si el producto no fue comprado, entonces no puede ser
            regresado
15
16 #print(review_score)
17 #print(reviw_refund)
18 review.append([i, review_score, review_refund])
```

Existen espacios vacíos de productos que no han sido comprados, entonces se da un banderazo, para retirarlos a partir de un ciclo while, después se obtiene un promedio de las reseñas para saber cual es mejor.

```
1 #Vamos a retirar los espacios vacíos, hay listas que no contienen nada
2
3 flag =True                                     #Damos un banderazo de salida verdadero
4 while flag:                                   #Mientras sea verdadero
5     flag=False                               #Cambiamos la bandera a falso
6     for i in review:                         #Iteramos sobre las reseñas
7         if not i[1]:                         #Verificamos si la lista se encuentra vacía
8             flag=True
9             #print(i)
10            review.remove(i)                 #Removemos las listas vacías
11 #print(review)
12
13 #Vamos a calcular el promedio de cada reseña, para saber cual fue mejor y luego
    acomdarlo
14 for i in review:
15     if len(i[1])!=0:
16         i[1]=sum(i[1])/len(i[1])           #Función que promedia
17     else:
18         i[1]=0
```

Se acomodan de mejor a peor reseña a partir de una función sort, y a juicio propio, se toma que si la reseña tiene más de 4.6 de puntuación es una de las mejores reseñas.

```
1 #print(review)
2 def SortReview(element):
3     return element[1]
4 review.sort(key = SortReview, reverse =True)    #Ordenamos del mejor al peor
    sacando el promedio de las reseñas
```

```

5 #print(review)
6
7
8 #Decidimos cuales son las mejores y peores reseñas, esto lo decidimos manualmente,
   si pasa de 4.6 es una buena reseña
9 #Guardamos en listas estos datos
10 best_review= []
11 worst_review = []
12 for i in review:           #Iteramos sobre las reseñas
13     if i[1]>4.6:
14         best_review.append(i)
15     else:
16         worst_review.append(i)

```

Se realiza la impresión de las mejores y peores reseñas

```

1 #Realizamos la impresión de las mejores y peores
2
3 input('La mejor reseña: (presiona Enter)  \n')
4 print('***** LA MEJOR RESEÑA ***** \n \n')
5 for i in range(len(best_review[0:10])):
6     print(lf.lifstore_products[best_review[i][0]][1] ,'\n')
7
8 input('La peor reseña: (presiona Enter)  \n')
9 print('***** LA PEOR RESEÑA *****\n \n')
10 for i in range(len(worst_review[0:10])):
11     print(lf.lifstore_products[worst_review[i][0]][1] ,'\n')
12
13 #print(best_review)
14 #print(worst_review)

```

Finalmente pasamos a la lista de los meses del año, donde se obtiene las ventas cada mes, así como el número de ingresos recibidos.

Primero creamos una lista con los meses del año, donde igual almacenaremos las ventas realizadas y los ingresos recibidos. Esto a partir de un ciclo for y un ciclo if, donde al final se realiza una copia para obtener listas de ventas y de ingresos.

```

1
2 #Realizamos una lista donde se almacenan los meses del años con string, igual con
   números
3 #Dejamos dos espacios en ceros, donde se almacenará el número de ventas por mes y
   los ingresos percibidos
4 Meses = [['Enero',1,0,0],['Febrero',2,0,0],['Marzo',3,0,0], ['Abril',4,0,0],['Mayo',
   5         5,0,0],['Junio',6,0,0],
6         ['Julio',7,0,0],['Agosto',8,0,0],['Septiembre',9,0,0], ['Octubre',
   7         10,0,0],['Noviembre',11,0,0],['Diciembre',12,0,0]]
8
9 for i in lf.lifstore_sales:           #Iteramos sobre la lista de ventas
10     mes_indice = int(i[3][3:5])-1      #Se obtiene el índice del mes
11     producto_indice = i[1]-1          #Se obtiene el índice del producto
12     if i[4] == 0:                     #Si el producto fue regresado, no se
13         almacena la venta
14
15     price = lf.lifstore_products[producto_indice][2] #Hacemos la variable
16     precio la cual almacena los el precio, recogido de la lista de productos

```

```

13     Meses[mes_indice][2] += 1           #Si eso se cumple aumentamos una
    venta
14     Meses[mes_indice][3] += price      #Aumentamos el ingreso
15
16
17
18 #Realizamos copias de la lista Meses
19 sales_months = Meses.copy()
20 takings_months = Meses.copy()

```

Se realizan funciones donde se ordenan los meses donde hubo más ventas y donde hubo más ingresos

```

1
2
3 #Realizamos funciones sort, una donde se acomoda por número de ventas y otra por
    cantidad obtenida en el mes
4 def Sales_months(element):
5     return element[2]
6
7 def Takings_months(element):
8     return element[3]
9
10
11 sales_months.sort(key=Sales_months, reverse = True)
12 takings_months.sort(key = Takings_months, reverse = True)
13 #print(sales_months)
14 #print(takings_months)
15 input('Finalmente se muestra los datos por meses : (presiona Enter) \n')

```

Finalmente se imprimen dichos valores los meses con más ventas y los ingresos por mes, también se realiza un ciclo for sobre los ingresos para obtener el ingreso total.

```

1 print('***** LAS VENTAS POR MESES Y LOS MESES CON MÁS VENTAS FUERON:
    *****\n \n')
2
3
4 for i in range(0,len(sales_months)):
5     print('En el mes de ', sales_months[i][0], ' hubo ', sales_months[i][2], ' ventas
    . \n')
6
7
8
9 print('***** LOS INGRESOS POR MESES Y LOS MESES CON MÁS INGRESOS
    FUERON: *****\n \n')
10 for i in range(0,len(takings_months)):
11     print('En el mes de ', takings_months[i][0], ' hubo ', takings_months[i][3], '
    ingresos. \n')
12
13 print('***** El total de ingresos totales es:
    *****\n \n')
14
15 ingreso_total = 0
16
17 for i in takings_months:
18     ingreso_total += i[3] #Aumentamos el numero de ingresos por mes para obtener el
    total

```

```
19 print('El total de ingresos ha sido:', ingreso_total) #Imprimos el ingreso total del anho
```

3. Solución al problema

- 1.- Se realizó un análisis de los productos de de la tienda virtual **LifeStore**, donde se obtuvo los productos más vendidos. El producto más vendido es *SSD para Servidor Supermicro SSD-DM128-SMCMVN1, 128GB, SATA III, mSATA, 6Gbit/s*, mientras que el menos vendido es *MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0*, el más buscado es *SSD para Servidor Supermicro SSD-DM128-SMCMVN1, 128GB, SATA III, mSATA, 6Gbit/s*, mientras que el menos buscado es *Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0*.

Estos productos sólo reflejan lo más alto y lo más bajo, pero a partir del análisis realizado, también se halló que tanto los discos duro, como los procesadores se encuentran entre los productos más solicitados por los usuarios, mientras que las tarjetas de vídeos y las tarjetas madre son poco solicitadas, por lo que no vale la pena abastecerse de muchas de ellas.

- 2.- De manera similar, se hallo que por categorías, lo más buscado son procesadores y discos duros, mientras que memorias USB y tarjetas de vídeo son las menos solicitadas.
- 3.- También en las reseñas los procesadores llevan la delantera, siendo los que tienen mayor calificación por los usuarios, sorprendentemente las tarjetas de vídeo aunque son poco solicitadas, son bastante eficientes, ya que también se encuentran entre las mejores reseñas. Mientras que las peores reseñas se lo llevan las tarjetas madre, las cuales no son ni las más buscadas, ni las vendidas, entonces se debe plantear retirar varios productos de las mismas.
- 4.- Como último punto se tienen las ventas por meses y los ingresos recibidos, el mes donde mayor hubo ventas en el año es el mes de abril, esto puede deberse debido al inicio de la epidemia y que muchos usuarios requerían de equipos electrónicos para poder trabajar. Este mes tuvo 74 ventas e ingresos de 191066. El total de ingresos en lo que va del año hasta agosto fue 737916.

4. Conclusión.

Una estrategia factible que debe realizar la tienda es retirar productos de tarjetas madre y tarjetas de vídeo, ya que no esos productos no se han vendido en los últimos meses, también debe realizar un pedido de procesadores y discos duros, debido a que estos presentan las mejores reseñas y son muy pedidos y buscados por los usuarios. Esto ayudará al obtener mayores ingresos en la tienda en los meses por venir.