

PROYECTO 1.

INTRODUCCIÓN A PYTHON

Institución: EMTECH

Estudiante: Admin Arenas Rodríguez

Fecha de entrega: 1 de septiembre del 2020



Índice

Introducción:.....	3
Código:	4
Solución al problema:	14
.....	17
.....	17
Conclusión:.....	18

Introducción:

Objetivo.

El objetivo de este trabajo es poner en práctica las bases de programación en Python aprendidas en la Parte 1: Fundamentos de Programación con Python. Aplicadas al análisis y clasificación de datos mediante la creación de programas de entrada de usuario y validaciones, uso y definición de variables y listas, operadores lógicos y condicionales para la clasificación de información.

Descripción del caso.

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.

Consigna.

Derivado de la situación, la Gerencia de Ventas solicita realizar un análisis de la rotación de productos identificando los siguientes elementos:

- 1) Productos más vendidos y productos rezagados a partir del análisis de las categorías con menores ventas y categorías con menores búsquedas.
- 2) Productos por reseña en el servicio a partir del análisis de categorías con mayores ventas y categorías con mayores búsquedas.
- 3) Sugerir una estrategia de productos a retirar del mercado, así como sugerencia de cómo reducir la acumulación de inventario considerando los datos de ingresos y ventas mensuales.

Código:

```
1. print("""Nota: un usuario válido es aquel que empieza con la palabra \"lifestore\"
   todo en minúscula, seguido de 3 números. Para tener las credenciales necesarias, los
   últimos 3 números juntos deben ser mayores a 100, por ejemplo \"lifestore101\" """)
2. decision=0 #variable que nos ayudará a saber si el usuario quiere seguir intentando
   entrar al sistema o ya no
3.
4. #Bucle while donde se interactúa con el usuario y nos ayuda a que el mismo decida si
   quiere continuar o salir
5. #####
   #####
6. while decision == 0:
7.     Usuario_Administrador = input("Para ingresar escribe un usuario-administrador
   válido, con las credenciales necesarias: ") #Variable donde se guardará el Usuario-
   Administrador y se muestra el mensaje de bienvenida
8.     print(" ")
9.     contador=-1 #Contador que nos ayudará a ingresar a los caracteres de la variable
   Usuario_Administrador
10.    contador2=0 #Contador que nos ayudará a saber si el usuario está bien escrito
11.
12.    #Ciclo for para validar el usuario
13.    #####
    #####
14.    for i in "lifestore123":
15.        contador+=1
16.        if len(Usuario_Administrador)!=12:#condición en la que se verifica que el
   usuario-administrador cuente con 12 caracteres
17.            print("Error, tu usuario-administrador debe contener 9 letras, seguido
   de 3 números")
18.            break
19.            if contador>8:#condición para validar que el usuario esté bien escrito en
   la parte final
20.                if (Usuario_Administrador[contador] in
   ["0","1","2","3","4","5","6","7","8","9"]):
21.                    contador2+=1
22.                    continue
23.                else:
24.                    print("Usuario-administrador inválido, error en los últimos 3
   dígitos")
25.                    contador2=0 #Inicializamos contador2 en cero para que no muestre
   que no se tienen las credenciales necesarias, pues ese no es el error del usuario
26.                    break
27.                    if Usuario_Administrador[contador] != i:#condición para validar que el
   usuario esté bien en los primeros 8 caracteres
28.                        if contador > 8:
29.                            continue
30.                            print("Error, el usuario-administrador contiene mayúsculas o está mal
   escrito")
31.                            break
32.
   #####
   #####
33.
34.    #If para validar las credenciales
35.    #####
   #####
36.    if contador2==3 and int(Usuario_Administrador[9:12])>100:
```

```

37.         print("Bienvenido al sistema de Ventas de Lifestore :D ")
38.         #Una vez dentro del sistema, se genera una lista con todos los años en los
que haya habido ventas
39.         años=[int(lifestore_sales[0][3][6:10])]# primero se guarda el año de la
primer venta registrada en la lista lifestore_ventas
40.         for producto in lifestore_sales:# se genera un for para revisar todas las
ventas en la lista lifestore_ventas
41.             nuevo_año=0 #variable que nos ayudará a saber si el año ya se tiene o
aún no se captura en la nueva lista años
42.             for año in años: #for que ingresará a todos los elementos de la lista
años
43.                 if int(producto[3][6:10])==año: #condición que hará que pase al
siguiente elemento si es que ya lo tenemos en la lista años
44.                     continue
45.                 else:
46.                     nuevo_año+=1 # si no se cumple la condición se sumará 1 a la
variable nuevo_año
47.             if (nuevo_año+1)>len(años): # en caso de que el elemento no esté en la
lista año, nuevo_año+1 deberá ser más grande que len(años)
48.                 años.append(int(producto[3][6:10])) #si eso pasa agregamos el año
a la lista años
49.                 años.sort()#ordenamos la lista años
50.                 meses=["enero", "febrero", "marzo", "abril", "mayo", "junio", "julio",
"agosto", "septiembre", "octubre", "noviembre", "diciembre"]
51.                 #creamos una lista con todos los meses del año para una mejor visualización
más adelante
52.                 while decision==0: # mientras la variable decision sea igual a cero el
usuario estará dentro del sistema
53.                     ejecucion_usuario=input("""¿Qué quieres hacer?
1) Ver productos más vendidos y productos rezagados
2) Ver listado de 20 productos con mejor reseña y los 20 con peor reseña
3) Ver total de ingresos y ventas promedio mensuales, total anual y meses con más
ventas en el año
4)Salir""")#Se presentan las opciones anteriores al usuario para que pueda elegir
54.                     if ejecucion_usuario=="1": #Si el usuario contesta 1 se realizarán los
siguientes listados
55.
print("""#####
###
#
#                               #
#           Top 50 de productos con mayores ventas   por mes           #
#                               #
#####
[número de ventas, año, mes, [id_product, name, price, category, stock]]""")
56.                 for año in años: #este for hará que se haga un listado para todos
los años en los que haya habido ventas
57.                     for mes in range(1,13): #for para que haya un listado para
todos los meses del año
58.                         ventas_top_50=[] #en esta lista se guardarán los top de
ventas para todos los meses y por eso que aquí se inicializa
59.                         for producto in lifestore_products: #este for es para ir
checando las ventas, producto por producto
60.                             ventas=0 #variable donde se guardarán el número de
ventas para todos los productos, es por eso que en este paso se inicializa en cero
61.                             for venta in lifestore_sales:#este for es para
recorrer todas las ventas en la lista lifestore
62.                                 if venta[1]==producto[0] and
mes==int(venta[3][3:5]) and año==int(venta[3][6:10]):#si se cumplen las 3
condiciones se sumará 1 a las ventas del respectivo producto en el año y mes
respectivo
63.                                     ventas+=1

```

```

64.         if ventas==0: # Si las ventas son iguales a cero no se
            incluyen en el listado
65.             continue
66.         else:
67.             ventas_top_50.append([ventas,año,mes,producto])#en
            esta línea guardamos las ventas totales en el mes y año respectivo con la
            información del producto
68.         if ventas_top_50==[]: #si la lista está vacía se imprime
            el siguiente renglón
69.             print("\nEn ",meses[mes-1]," del año " , año," no hay
            ventas")
70.         else:#aquí se imprime la información del top del año y mes
            respectivo
71.             print("\nEn el mes de ",meses[mes-1]," del año " ,
            año,""" el top 50 es:\n""")
72.
73.             for i in range(len(ventas_top_50)):#antes de imprimir
            el top, se ordenan los productos respecto a las ventas de mayor a menor
74.                 for j in range(len(ventas_top_50)):
75.                     if ventas_top_50[i] > ventas_top_50[j]:
76.                         auxiliar = ventas_top_50[i]
77.                         ventas_top_50[i] = ventas_top_50[j]
78.                         ventas_top_50[j] = auxiliar
79.             ventas_top_50=ventas_top_50[0:50]#nos quedamos solo
            con los primeros 50
80.             for i in ventas_top_50:#imprimimos el listado
81.                 print(i)
82.
            print("""\n#####
            #####
            #
            #                               #
            #               Los 100 productos con mayores búsquedas                #
            #                               #
            #####
            [número de búsquedas, [id_product, name, price, category, stock]]\n""")
83.             top_100_búsquedas=[] #en esta lista se guardarán los top de
            búsquedas
84.             for producto in lifestore_products:#for para ingresar a todos los
            productos en la lista lifestore_products
85.                 busquedas=0 #en esta variable se guardarán el total de
            búsquedas por producto, es por eso que aquí se inicializa
86.                 for busqueda in lifestore_searches:#for para ingresar
            a todas las búsquedas de lifestore_searches
87.                     if busqueda[1]==producto[0]:#si el id_product del
            respectivo producto en el que se encuentre el primero for es igual al elemento
88.                         #de la lista lifestore_searches en el que esté
            el segundo for entonces se suma una búsqueda y así sucesivamente
89.                         busquedas+=1
90.                 if busquedas==0:# si no hubo búsquedas no se contará
            en el listado
91.                     continue
92.                 else: #se guarda la información obtenida sobre las
            búsquedas del respectivo producto
93.                     top_100_búsquedas.append([busquedas,producto])
94.                     for i in range(len(top_100_búsquedas)): #se ordenan los productos
            respecto a las búsquedas de mayor a menor
95.                         for j in range(len(top_100_búsquedas)):
96.                             if top_100_búsquedas[i] >
            top_100_búsquedas[j]:
97.                                 auxiliar = top_100_búsquedas[i]

```

```

98.                                     top_100_búsquedas[i] =
    top_100_búsquedas[j]
99.                                     top_100_búsquedas[j] = auxiliar
100.                                top_100_búsquedas=top_100_búsquedas[0:100]#nos quedamos solo con
    el top 100 de las búsquedas
101.
102.                                for i in top_100_búsquedas:#imprimimos el listado
103.                                    print(i)
104.
105.
    print("""\n#####
    #####
    #
    #                               Top 50 de productos con menores ventas por categoría                               #
    #
    #####
    [ventas,[id_product, name, price, category, stock]] """)
106.                                categorias=[lifestore_products[0][3]]#se obtienen las categorías
    existentes en el listado lifestore_products, de manera parecida a lo que se hizo con
    los años
107.                                for producto in lifestore_products:
108.                                    nuevo_categoria=0
109.                                    for categoria in categorias:
110.                                        if producto[3]==categoria:
111.                                            continue
112.                                        else:
113.                                            nuevo_categoria+=1
114.                                    if (nuevo_categoria+1)>len(categorias):
115.                                        categorias.append(producto[3])
116.                                for categoria in categorias:#obtenemos el top 50 de menores ventas
    por categoría de la misma manera como se hizo se hizo en el top 50 por mes y año
117.                                    top_50_cat=[]
118.                                    for producto in lifestore_products:
119.                                        ventas_producto=0
120.                                        for venta in lifestore_sales:
121.                                            if producto[0]==venta[1] and producto[3]==categoria:
122.                                                ventas_producto+=1
123.                                        if producto[3]==categoria:
124.                                            top_50_cat.append([ventas_producto, producto])
125.                                    for i in range(len(top_50_cat)):#se ordena cada top de menor a
    mayor respecto a las ventas
126.                                        for j in range(len(top_50_cat)):
127.                                            if top_50_cat[i] < top_50_cat[j]:
128.                                                auxiliar = top_50_cat[j]
129.                                                top_50_cat[j] = top_50_cat[i]
130.                                                top_50_cat[i] = auxiliar
131.                                    top_50_cat=top_50_cat[0:50]#nos quedamos con los 50 con
    menores ventas
132.                                print("\nEl top 50 de productos con menores ventas de la
    categoría ",categoria, " es:\n")#imprimimos el listado
133.                                for i in top_50_cat:
134.                                    print(i)
135.
136.
    print("""\n#####
    #####
    #
    #                               Los 100 productos con menores búsquedas                               #
    #
    #####
    [número de búsquedas, [id_product, name, price, category, stock]]\n""")

```

```

137.
138.         top_100_busquedas=[]#Se realiza un proceso igual al que se realizó
           cuando se obtuvo el top 100 de mayores búsquedas
139.         #sin embargo el ordenamiento esta vez será de
           menor a mayore respecto a las búsquedas
140.         for producto in lifestore_products:
141.             ventas=0
142.             for venta in lifestore_searches:
143.                 if venta[1]==producto[0]:
144.                     ventas+=1
145.
146.             top_100_busquedas.append([ventas,producto])
147.         for i in range(len(top_100_busquedas)):
148.             for j in range(len(top_100_busquedas)):
149.                 if top_100_busquedas[i] <
top_100_busquedas[j]:
150.                     auxiliar = top_100_busquedas[j]
151.                     top_100_busquedas[j] =
top_100_busquedas[i]
152.                     top_100_busquedas[i] = auxiliar
153.         top_100_busquedas=top_100_busquedas[0:100]
154.
155.         for i in top_100_busquedas:
156.             print(i)
157.
158.         print("""\n#####
#####
#
#                               #
#               Número de ventas por categoría               #
#                               #
#####\n""")
159.         for categoria in categorias:# for para recorrer todos los
           elementos de la lista categorias
160.             ventas_total=0#se inicializa en cero la variable para contar
           el total de ventas de cada categoría
161.             for producto in lifestore_products:#for para reocrer todos
           los productos de la lista lifestore_products
162.                 for sale in lifestore_sales:#for para recorrer
           todas las ventas en la lista lifestore_sales
163.                     if sale[1]==producto[0] and
           producto[3]==categoria:#condición para saber si esa venta se contará para la
           respectiva categoría
164.                         ventas_total+=1
165.             print("\nEl número de ventas para la categoría ",categoria, "
           es: ",ventas_total)#imprimimos la información obtenida
166.
167.         print("""\n#####
#####
#
#                               #
#               Número de búsquedas por categoría               #
#                               #
#####\n""")
168.         for categoria in categorias:# for para recorrer todos los
           elementos de la lista categorías
169.             busqueda_total=0#se inicializa en cero la variable para contar
           el total de búsquedas de cada categoría
170.             for producto in lifestore_products:#for para recorrer todos
           los productos de la lista lifestore_products

```



```

171.                 for busqueda in lifestore_searches:#for para
                    recorrer todas las búsquedas en la lista lifestore_searches
172.                 if busqueda[1]==producto[0] and
                    producto[3]==categoria:#condición para saber si esa búsqueda se contará para la
                    respectiva categoría
173.                     busqueda_total+=1
174.                     print("\nEl número de búsquedas para la categoría ",categoria,
                        " son: ",busqueda_total)#imprimimos información obtenida
175.
176.     print("""\n#####
        #####
        #
        #                               #
        #                               #
        #                               #
        #####\n""")
177.         for categoria in categorias:#se realiza el mismo proceso que en el
            bloque anterior, solo que ahora solo se sumará el stock registrado en la lista
            lifestore_products
178.             stock_total=0
179.             for producto in lifestore_products:
180.                 if producto[3]==categoria:
181.                     stock_total+=producto[4]
182.                 print("\nEl inventario para la categoría ",categoria, " es de:
                    ",stock_total)
183.
184.
185.         elif ejecucion_usuario=="2":
186.
187.
188.     print("""\n#####
        #####
        #
        #                               #
        #                               #
        #                               #
        #####
        [promedio reseñas, ventas, [id_product, name, price, category, stock]]""")
189.         print("Nota: el \"promedio reseñas\" se calculó sumando todas las
            reseñas en la lista lifestore_sales del respectivo producto y se dividió entre el
            número de ventas del mismo\n")
190.         top_20_reseña=[]#en esta lista se guardaran los productos con su
            respectivo valor de promedio reseñas
191.         for producto in lifestore_products: #for para ingresar a todos los
            productos de la lista lifestore_products
192.             ventas=0 #variable donde se guardarán todas las ventas
            de cada producto
193.             reseña=0 #variable para guardar la suma de las reseñas
194.             for venta in lifestore_sales:#for para ingresar a
            todas las ventas de la lista lifestore_sales
195.                 if venta[1]==producto[0]: #si el id_producto es
                    igual al de la venta en la lista lifestore_sales se suma una venta y la respectiva
                    reseña
196.                     ventas+=1
197.                     reseña+= venta[2]
198.                 if ventas==0:# si las ventas son iguales a cero, el
                    promedio reseña será igual a cero
199.                     top_20_reseña.append([0,producto])
200.                 else:
201.                     top_20_reseña.append([reseña/ventas,
                        ventas,producto])#se ingresa la información del producto así como su promedio reseña

```

```

202.         for i in range(len(top_20_reseña)): # se ordena el top de mayor a
menor promedio reseña
203.             for j in range(len(top_20_reseña)):
204.                 if top_20_reseña[i] > top_20_reseña[j]:
205.                     auxiliar = top_20_reseña[i]
206.                     top_20_reseña[i] = top_20_reseña[j]
207.                     top_20_reseña[j] = auxiliar
208.         top_20_reseña=top_20_reseña[0:20]# nos quedamos con los mejores 20
209.
210.         for i in top_20_reseña: #imprimimos el listado
211.             if top_20_reseña[0]==0:
212.                 continue
213.             else:
214.                 print(i)
215.
216.         print("""\n#####
#####
#
#
#           20 productos con peores reseñas
#
#
#####
[promedio reseñas, devoluciones, ventas, [id_product, name, price, category, stock]]
Primero se ordena de la peor reseña a la mejor y si hay empate se pondrá primero el
producto con mayor cantidad de devoluciones\n""")
217.         top_20_reseña=[] # se realiza el mismo proceso que para el
anterior listado, solo que al ordenar se ordena de menor a mayor reseña y nos
quedamos con los peores 20, además se toma como factor de desempate el número de
devoluciones
218.         for producto in lifestore_products:
219.             ventas=0
220.             reseña=0
221.             devolucion=0
222.             for venta in lifestore_sales:
223.                 if venta[1]==producto[0]:
224.                     ventas+=1
225.                     reseña+= venta[2]
226.                     devolucion+= venta[4]
227.             if ventas==0:
228.                 continue
229.             else:
230.                 top_20_reseña.append([reseña/ventas, devolucion,
ventas,producto])
231.         for i in range(len(top_20_reseña)):
232.             for j in range(len(top_20_reseña)):
233.                 if top_20_reseña[i] < top_20_reseña[j]:
234.                     auxiliar = top_20_reseña[j]
235.                     top_20_reseña[j] = top_20_reseña[i]
236.                     top_20_reseña[i] = auxiliar
237.                 elif top_20_reseña[i][0] ==
top_20_reseña[j][0]:
238.                     if top_20_reseña[i][1] >
top_20_reseña[j][1]: #si hay empate, ordenamos por número de devoluciones
239.                         auxiliar = top_20_reseña[i]
240.                         top_20_reseña[i] = top_20_reseña[j]
241.                         top_20_reseña[j] = auxiliar
242.         top_20_reseña=top_20_reseña[0:20]
243.         for i in top_20_reseña:
244.             if top_20_reseña[0]==0:
245.                 continue
246.             else:

```

```

247.             print(i)
248.
249.
250.         elif ejecucion_usuario=="3":
251.
252.             print("""\n#####
#####
#
#                               #
#               Total de ingresos y ventas promedio mensuales           #
#                               #
##### """)
253.             for año in años: #for para realizar el procedimiento en todos los
años en los que ha habido ventas
254.                 ventas_totales=0 #variable en la que se guardaran las ventas
en dinero de cada año
255.                 ventas_totales_dinero=0 #variable en la que se guardaran las
ventas de cada año
256.                 for producto in lifestore_products: #con este for ingresaremos
a todos los productos en la lista lifestore_products
257.                     for venta in lifestore_sales: #con este for
ingresaremos a todas las ventas en la lista lifestore_sales
258.                         if venta[1]==producto[0] and
año==int(venta[3][6:10]): #si el id_producto respectivo y el año de venta
corresponden a los que se está verificando, el precio del producto se suma a las
ventas_totales_dinero, así como la venta a ventas_totales
259.                             ventas_totales+=1
260.                             ventas_totales_dinero+=producto[2]
261.                 if ventas_totales==0:
262.                     print("\nEn ", año, " no hubo ventas") # si las ventas son
igual a cero se imprime que en ese año no hubo ventas
263.                 else:
264.                     print("\nPara el año " , año, ""
Ventas totales: $""",ventas_totales_dinero, #si las ventas son diferentes de cero,
se imprime el total de ingresos indicando el año respectivo
"""\nVentas promedio mensuales: $""",ventas_totales_dinero/12, "\nVentas totales:
",ventas_totales,"\nVentas promedio mensuales: ",ventas_totales/12 )#se calcula el
promedio de ventas de los meses del año respectivo
265.
266.
267.
268.             print("""\n#####
#####
#
#                               #
#               Total de ingresos por mes                               #
#                               #
##### """)
269.             ventas_meses_lista=[] #en esta lista se guardaran las ventas de
todo el histórico en la lista lifestore_sales
270.             for año in años: #for para mostrar el total de ventas y promedio
por mes de todos los años en los que hay habido ventas
271.                 for mes in range(1,13): #for para mostrar el total de ventas
de todos los meses en cada año en el que haya habido venta
272.                     ventas_totales=0 # en esta variable se guardarán las ventas
totales del mes respectivo
273.                     ventas_totales_dinero=0 # en esta variable se guardarán las
ventas totales en dinero del mes respectivo
274.                     for producto in lifestore_products: #aquí se ingresa a
todos los productos de la lista lifestore_products para ver que tantos ingresos
dejaron el mes respectivo del año en turno

```

```

275.         for venta in lifestore_sales:# se ingresa a todos los
           elementos de la lista lifestore_sales
276.             if venta[1]==producto[0] and
mes==int(venta[3][3:5]) and año==int(venta[3][6:10]):#usamos condiciones para ver
           que la venta haya sido del producto, mes y año en turno
277.                 ventas_totales_dinero+=producto[2] # si es así
           el precio del producto se suma a las ventas_totales_dinero totales del mes
           respectivo del año en turno y de la misma manera se suma la venta a  ventas_totales
278.                 ventas_totales+=1
279.                 ventas_meses_lista.append([ventas_totales,
           ventas_totales/30, mes, año])#una vez que se obtuvo toda la información para ese mes
           se guarda la información en la lista ventas_meses_lista
280.                 if ventas_totales==0:# si las ventas son iguales a cero en
           el mes en turno se muestra un mensaje avisando sobre eso
281.                     print("\nEn ",meses[mes-1]," del año " , año," no hay
           ventas")
282.                 else:# si no, se muestra información recabada
283.                     print("\nEn el mes de ",meses[mes-1]," del año " ,
           año,"")
284. Ingresos: $""",ventas_totales_dinero, "\nVentas totales: ",ventas_totales)
285.
286.     print("""\n#####
           #####
           #
           #                               #
           #                               #
           #                               #
           #####\n""")
287.         for i in range(len(ventas_meses_lista)):#ordenamos de mayor a
           menor las datos de la lista ventas_meses_lista obtenida en el bloque anterior
288.             for j in range(len(ventas_meses_lista)):
289.                 if ventas_meses_lista[i] >
           ventas_meses_lista[j]:
290.                     auxiliar = ventas_meses_lista[i]
291.                     ventas_meses_lista[i] =
           ventas_meses_lista[j]
292.                     ventas_meses_lista[j] = auxiliar
293.         for año in años:# se muestra para cada año que haya habido ventas
           el top 3 de meses con más ventas
294.             lugar=0# variable que nos ayudará a saber si ya se ingresó el
           top 3
295.             print("\nEn el ",año, " el top 3 de ventas fue")
296.             for mes in ventas_meses_lista: #ingresaremos a todos los
           elementos de la lista ventas_meses_lista
297.                 if lugar ==3: #si la variable lugar es igual a 3 nos
           salimos del for, quiere decir que ya se obtuvieron los primeros 3 lugares del año y
           mes respectivo
298.                     break
299.                 if mes[3]==año:#si el año guardado es igual al año en
           turno se imprimirá en el top
300.                 if mes[0]==0:#si las ventas son cero del mes
           encontrado quiere decir que ya no hay más meses con ventas en ese año
301.                     print(lugar+1, ".- No hay más meses con ventas")
302.                     break
303.                 else: #si esa condición es falsa se imprime la
           información del mes en el año en turno y la variable lugar suma 1
304.                     lugar+=1
305.                     print(lugar, ".- ", meses[mes[2]-1], " con
           ",mes[0], " ventas" )
306.             print("")
307.

```

```

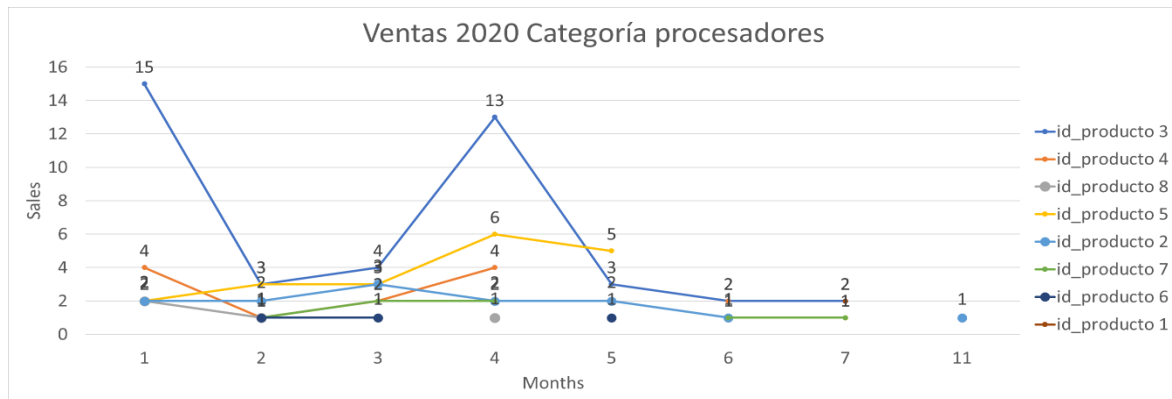
308.
309.
310.
311.         elif ejecucion_usuario=="4":# si el usuario ingresa 4 la variable
           decision toma el valor de uno y nos saca del programa
312.             decision=1
313.         else:# si la variable no es ninguna de las opciones solicitadas le
           pide al usuario que lo vuelva a intentar
314.             print("Respuesta invalida, puede que tu respuesta contenga algún
           espacio. Escribe un número o ingresa 4 si quieres salir")
315.
316.
317.
318.     elif (contador2==3):
319.         print("Lo sentimos, no cuentas con las credenciales suficientes")
320.         contador2 = 0 #mensaje si no cuenta con las credenciales necesarias
321.
           #####
           #####
322.
323.
324.     #En este while se le preguntara al usuario si quiere seguir intentando o
           dejarlo de hacer
325.
           #####
326.     while contador2 == 0:
327.         decision_usuario=input(";Quieres seguir intentando? sí/no: ")
328.         if decision_usuario == "sí" or decision_usuario == "Sí" or decision_usuario
           == "SÍ" or decision_usuario == "sÍ":
329.             contador2=1
330.         elif decision_usuario == "no" or decision_usuario == "No" or
           decision_usuario == "NO" or decision_usuario == "nO":
331.             decision=1
332.             contador2=1
333.         else:
334.             print("Respuesta no valida (debes usar acentos)")
335.             contador2=0
336.
           #####
337.
338.
339. #####
           #####

```

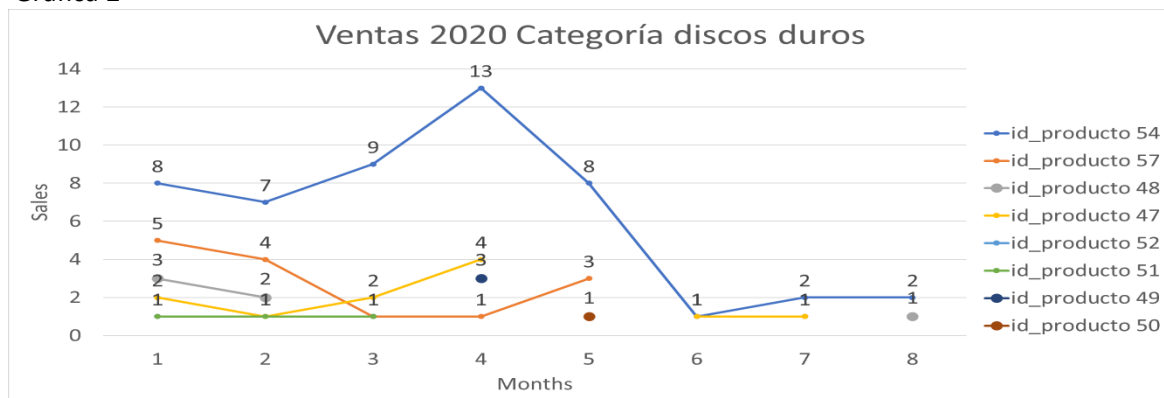
Solución al problema:

De acuerdo con los resultados obtenidos, se realizaron las gráficas de las ventas por mes, de las 3 categorías con mayor número de ventas. En los 3 casos se puede observar que entre los meses de enero y abril se obtuvieron buenos resultados, siendo tanto enero como abril los meses con mayores ventas, sin embargo, después de abril las ventas empezaron a decrecer.

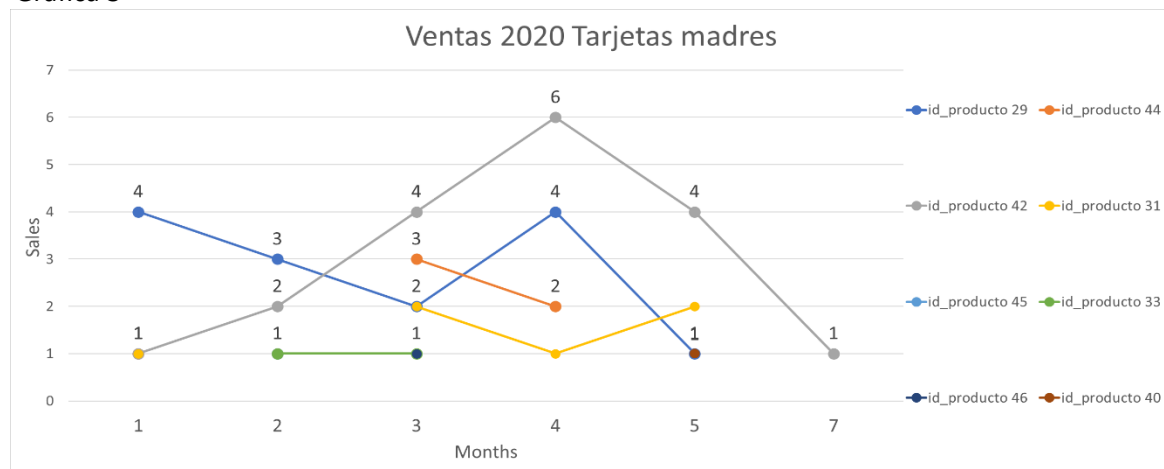
Gráfica 1



Gráfica 2

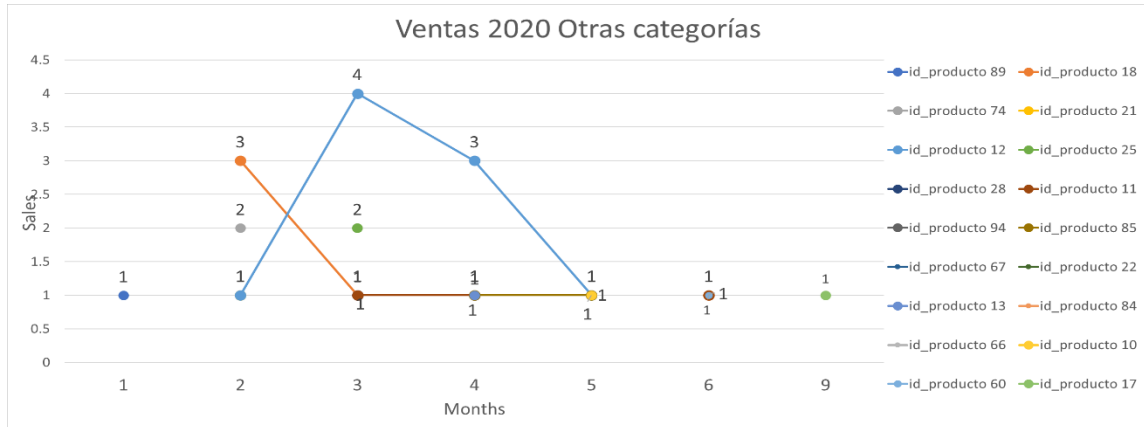


Gráfica 3



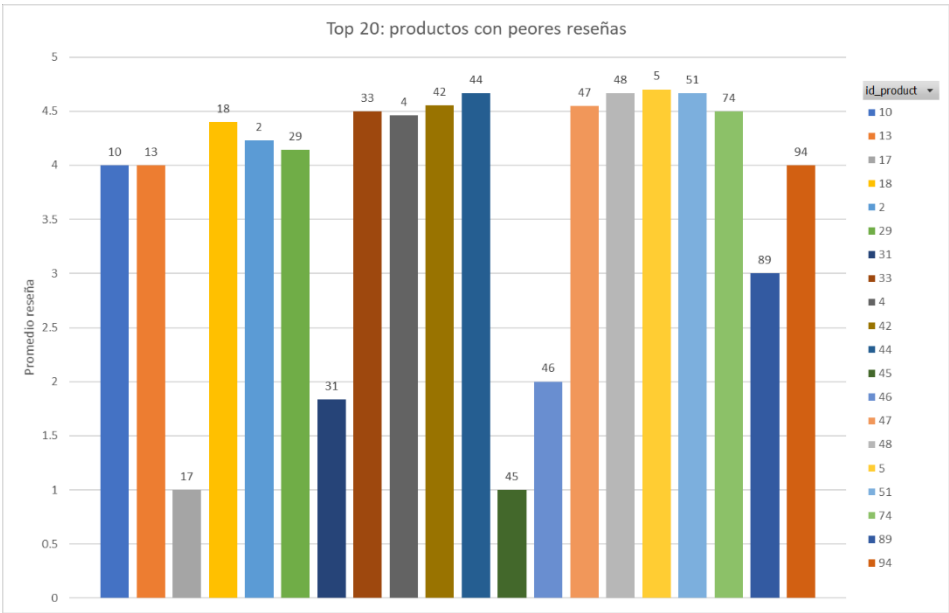
En la gráfica 4, se realiza el mismo análisis para las 5 categorías restantes y se obtiene un comportamiento similar al ya mencionado, con un buen nivel de ventas entre los meses de enero y abril, pero nuevamente hay una reducción significativa en el último trimestre.

Gráfica 4

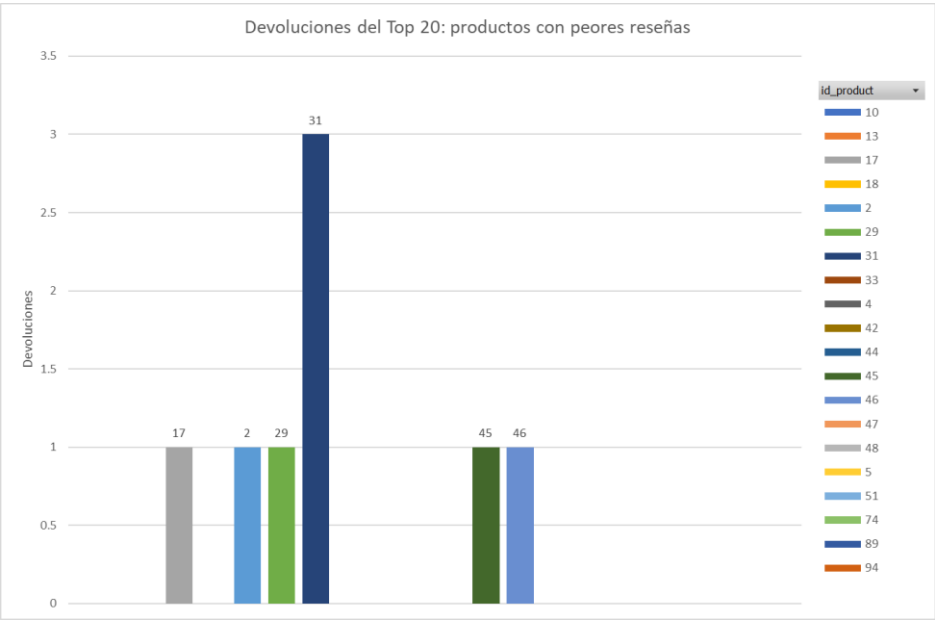


A continuación, en las gráficas 5 y 6 se muestran los 20 productos con peores reseñas y las devoluciones de estos, respectivamente. En ellas se puede notar, comparando con las gráficas anteriores, que aquellos productos con un promedio reseña (definido como la suma de las reseñas del producto entre el total de ventas) menor o igual a 4, difícilmente se vuelven a pedir o se terminan devolviendo.

Gráfica 5

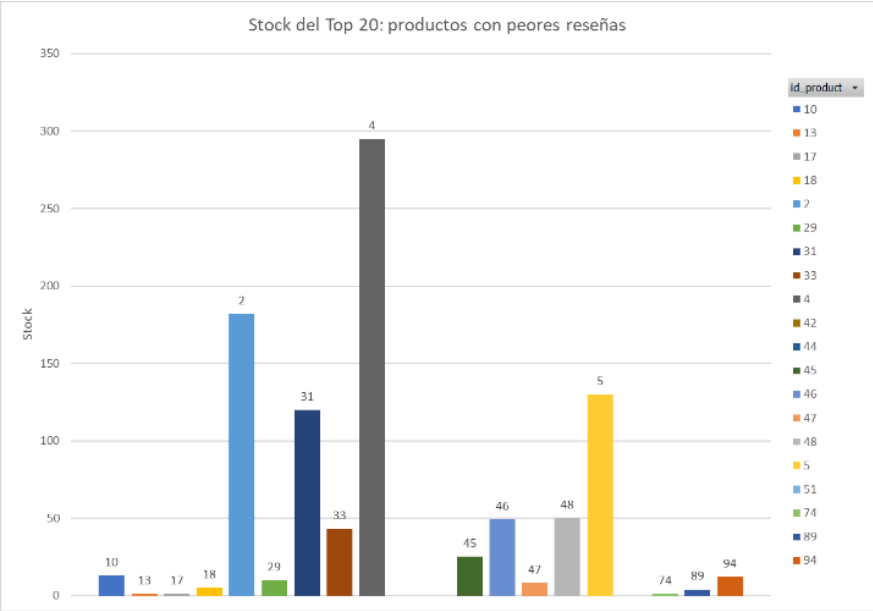


Gráfica 6



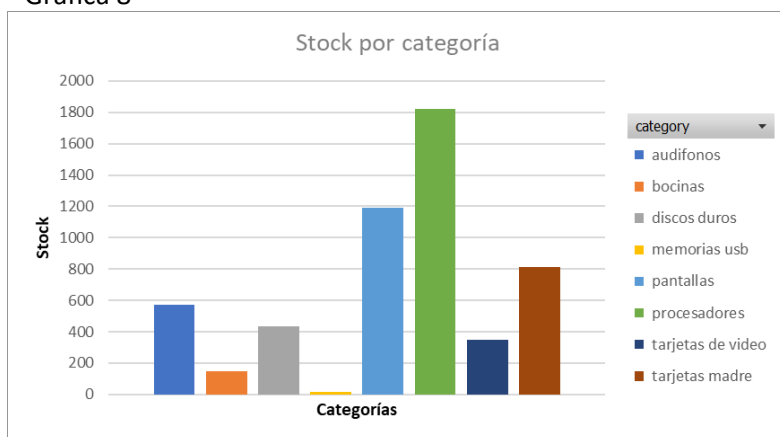
Ahora, si observamos el stock de estos productos en la gráfica 7, nos damos cuenta de que el producto 31 tiene una cantidad importante en el almacén a pesar de tener una mala reseña promedio y haber sido devuelta en 3 de sus 4 compras. Por otra parte, si juntamos el espacio que ocupan los productos 46 y 45 juntos, también es una cantidad importante, y sus reseñas promedio son bajas, además de tener devoluciones en sus únicas ventas.

Gráfica 7

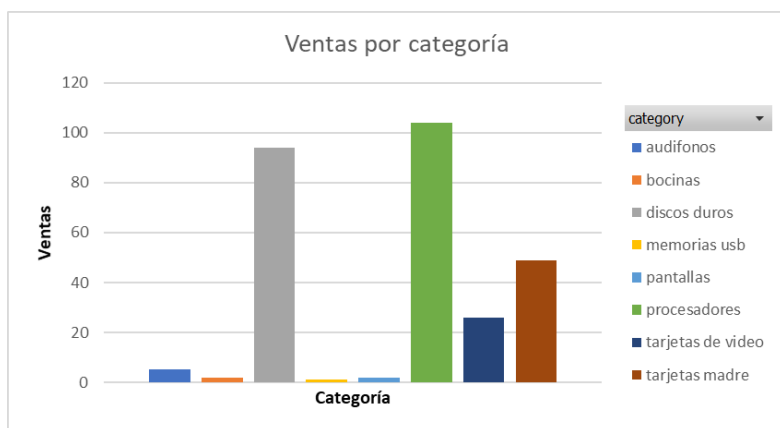


Por último, en los gráficos 8, 9 y 10 podemos observar el análisis del stock, ventas y búsquedas por categoría. En las gráficas 8 y 9 se puede ver el stock y las ventas respectivamente de cada categoría para el año 2020, las cuales concuerdan una con la otra, entre más ventas es mayor el stock, excepto en las categorías discos duros y pantallas, lo cual se puede deber a lo observado en la gráfica 10, que muestra un elevado número de búsquedas para la categoría discos duros y bajo en la búsqueda de pantallas, por lo que este cambio en las búsquedas de los usuarios puede ser el causante de la acumulación de inventario.

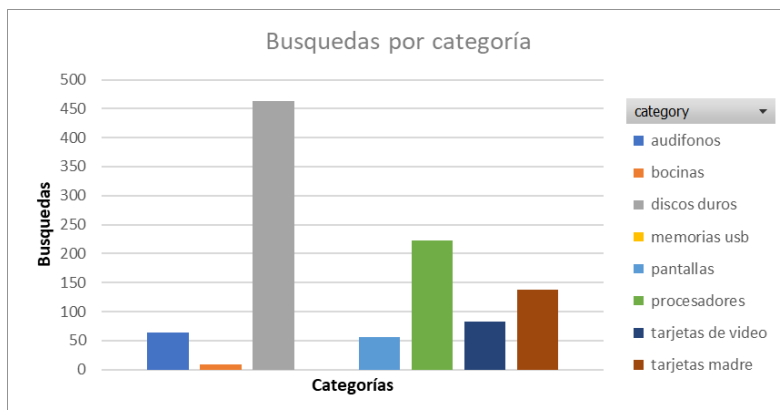
Gráfica 8



Gráfica 9



Gráfica 10



Conclusión:

En conclusión, de acuerdo con el análisis realizado anteriormente, se recomienda disminuir el inventario para los productos de la categoría pantallas y aumentarlo para los productos de la categoría discos duros, pues en el análisis se observó que, tanto en las búsquedas como en las ventas, los usuarios están teniendo preferencia hacia los productos de la categoría discos duros en este año.

De manera más particular se recomienda disminuir el inventario para los artículos con id_producto 31, 46 y 45 pues difícilmente se venderán más de ellos y en caso de ser así, el usuario suele devolverlos. Los 3 productos pertenecen a la categoría de tarjetas madre, por lo que también se recomienda poner atención en la calidad de los productos de esta categoría.

Para finalizar, la baja en las ventas en el tercer trimestre se puede deber al comportamiento estacional que tienen los productos electrónicos, por lo que se recomienda hacer un análisis sobre los mismos con una muestra que abarque mayor tiempo y así, dar un diagnóstico más preciso.