



ANÁLISIS DE ROTACIÓN DE PRODUCTOS PARA LA TIENDA ”LIFE STORE”

Proyecto Fin de Curso

Salazar Ledesma Héctor Mauricio

Septiembre 2021

Contenido

Introducción.....	2
Definición del código	3
Solución al problema	9
Conclusión	20

Introducción

Desde el principio de los tiempos, los seres humanos tuvimos la necesidad de acumular. Acumulábamos comida para el invierno, materiales que podían ser útiles en la construcción de herramientas y también pieles de animales para nuestra vestimenta, posteriormente con el paso del tiempo tuvimos la necesidad de organizar todo nuestro inventario, para poder así utilizar nuestros recursos de manera óptima y lograr que no se desperdiciara nada o que, en el caso extremo, se desperdiciara la menor cantidad posible. Haciendo la analogía, podemos llevar estos principios de acumulación y organización, al manejo del inventario de una tienda actual, haciendo un análisis de este podemos responder a las preguntas; ¿Cuáles de mis productos son los más buscados?, ¿Los más vendidos?, ¿Y los menos buscados?, ¿Cuáles fueron mis ingresos al final de año?, ¿Y al final del mes? Es así, que, motivado por estas preguntas, presentamos el siguiente reporte.

Definición del código

```
1 from lifestore_file import lifestore_sales, lifestore_searches, lifestore_products
2
3
4 # mas_vendidos = [id_product, cantidad, name]
5 mas_vendidos = []
6 # mas_buscados = [id_product, cantidad, name]
7 mas_buscados = []
8 # Número total de ventas
9 numero_ventas = len(lifestore_sales)
10 #Número total de productos
11 numero_productos = len(lifestore_products)
12 #Número total de búsquedas
13 numero_búsquedas = len(lifestore_searches)
14 #por_categoria = [id_product, cantidad por categoria, name]
15 por_categoria = []
16
17 """
18 Menú
19 #Creamos una lista con los reportes deseados
20 #Imprimimos los tipos que existen para que posteriormente el usuario ingrese
21 #el tipo de reporte deseado
22
23 """
24 reportes = ["Productos mas vendidos", "Productos mas buscados", "Menos vendidos por
25 categoria", "Menos buscados por categoria", "Productos con mejores reseñas", "Productos con
26 peores reseñas", "Total de ingresos mensuales", "Total de ingresos anuales", "Ventas promedio
27 por mes", "Meses con mas ventas"]
28
29 print("Seleccione el tipo de reporte que desea: ")
30 print("Estos pueden ser los que se muestran a continuación: ")
31 for reporte in reportes:
32     print(reporte)
33
34 #Varibale que decide si seguimos imprimiendo reportes o no
35 #"continuar" para continuar imprimiendo los reportes deseados y cualquier
36 #otro valor para no continuar
37 x = "continuar"
38
39 #Definimos un numero de intentos como 3 para que el usuario tenga la oportunidad
40 #de corregir si se ha equivocado
41
42 while x == "continuar":
43
44     for i in range(3):
45         reporte = input("Ingrese a continuación el reporte que desea: ")
46         if reporte in reportes:
47             print("Usted ha seleccionado: ", reporte)
48             break
49         else :
50             print("Por favor ingrese un valor válido")
51             print("Intentos restantes: ", 2-i)
```

```

52         if 2 - i == 0:
53             print("Número de intentos permitidos superado, adiós")
54
55 #Si el valor seleccionado fue "Productos mas vendidos" entonces
56 #buscamos en lifestore_sales hasta encontrar todos los productos iguales
57 #y vamos contabilizando con la variable c cuantas veces de vendió este producto
58 #por último los agregamos a la lista llamada mas_vendidos
59
60     if reporte == "Productos mas vendidos":
61         for j in range(numero_productos):
62             c = 0
63             for i in range(numero_ventas):
64                 if lifestore_sales[i][1] == j:
65                     c += 1
66             mas_vendidos.append([j,c,lifestore_products[j-1][1]])
67
68 #Ordenamos la lista de manera descendente
69
70     lista_ordenada = sorted(mas_vendidos, key = lambda x: x[1], reverse = True)
71     print("Estos son los 15 productos más vendidos y la cantidad de veces que se
72 vendieron")
73
74 #Imprimimos los 15 mas vendidos
75
76     for i in range(15):
77         print(i + 1, ".-", lista_ordenada[i][2])
78         print("Se vendió ", lista_ordenada[i][1], "veces" )
79         print("\n")
80
81 #Si el valor seleccionado fue "Productos mas buscados" ocurre un procesos similar al proceso
82 anterior,
83 #solo que en vez de buscar en lifestore_sales
84 #buscamos en lifestore_searches
85
86     if reporte == "Productos mas buscados":
87         for j in range(numero_productos):
88             c = 0
89             for i in range(numero_búsquedas):
90                 if lifestore_searches[i][1] == j:
91                     c += 1
92             mas_buscados.append([j,c,lifestore_products[j-1][1]])
93
94 #Ordenamos la lista de manera descendente
95
96     lista_ordenada = sorted(mas_buscados, key = lambda x: x[1], reverse = True)
97
98 #Imprimimos los 20 productos mas buscados
99
100     print("Estos son los 20 productos más buscados y la cantidad de veces que se
101 buscaron \n")
102     print("\n")
103     for i in range(20):
104         print(i + 1, ".-", lista_ordenada[i][2])
105         print("Cantidad de veces que se buscó este producto: " , lista_ordenada[i][1])
106         print("\n")
107
108 #Definimos una lista para separar las categorias existentes
109 #en nuestra lista de productos

```

```

110     categorias = []
111
112     #Si el valor seleccionado fue "Menos vendidos por categoria"
113     #imprimimos las categorias disponibles y con base en ese valor
114     #buscamos los menos vendidos
115     if reporte == "Menos vendidos por categoria":
116         print("\n")
117         print("Elija una categoria a buscar")
118         print("\n")
119         for i in range(numero_productos):
120             if lifestore_products[i][3] == lifestore_products[i-1][3]:
121                 continue
122             print(lifestore_products[i][3])
123             categorias.append(lifestore_products[i][3])
124
125     #Definimos un numero de intentos igual a 3 para seleccionar una categoria completa
126
127     for i in range(3):
128         categoria = input("Ingrese a continuación la categoria que desea: ")
129         if categoria in categorias:
130             print("Usted ha seleccionado: ", categoria)
131             print("\n")
132             break
133         else :
134             print("Por favor ingrese un valor válido")
135             print("Intentos restantes: ", 2-i)
136             if 2 - i == 0:
137                 print("Número de intentos permitidos superado, adiós")
138
139     #El proceso es igual que los dos anteriore
140
141     for j in range(numero_productos):
142         if categoria == lifestore_products[j-1][3]:
143             c = 0
144             for i in range(numero_ventas):
145                 if lifestore_sales[i][1] == j:
146                     c += 1
147             por_categoria.append([j,c,lifestore_products[j-1][1]])
148
149     #Ordenamos la lista de manera ascendente
150     lista_ordenada = sorted(por_categoria, key = lambda x: x[1])
151
152     #Imprimimos los 5 productos menos vendidos por categorias
153     print("\n")
154     print("Estos son los 5 productos menos vendidos por la categoria:
155 ", "\'", categoria, "\'", " y la cantidad de veces que se vendieron \n")
156     print("\n")
157     for i in range(5):
158         print(i + 1, ".-", lista_ordenada[i][2])
159         print("Cantidad de veces que se vendió este producto: " , lista_ordenada[i][1])
160         print("\n")
161
162     #Si el valor es "Menos buscados por categoria" hacemos exactamente lo mismo
163     #que el proceso anterior
164     if reporte == "Menos buscados por categoria":
165         print("\n")
166         print("Elige una categoria a buscar: ")
167

```

```

168     for i in range(numero_productos):
169         if lifestore_products[i][3] == lifestore_products[i-1][3]:
170             continue
171         print(lifestore_products[i][3])
172         categorias.append(lifestore_products[i][3])
173 #Definimos un numero de intentos permitidos igual a 3
174     for i in range(3):
175         categoria = input("Ingrese a continuación la categoria que desea: ")
176
177         if categoria in categorias:
178             print("Usted ha seleccionado: ", categoria)
179             break
180         else :
181             print("Por favor ingrese un valor válido")
182             print("Intentos restantes: ", 2-i)
183             if 2 - i == 0:
184                 print("Número de intentos permitidos superado, adiós")
185
186     for j in range(numero_productos):
187         if categoria == lifestore_products[j-1][3]:
188             c = 0
189             for i in range(numero_búsquedas):
190                 if lifestore_searches[i][1] == j:
191                     c += 1
192             por_categoria.append([j,c,lifestore_products[j-1][1]])
193
194 #Ordenamos la lista de manera ascendente
195
196     lista_ordenada = sorted(por_categoria, key = lambda x: x[1])
197     print("\n")
198     print("Estos son los 20 productos menos buscados por la categoria: ", "\n",
199 categoria, "\n", " y la cantidad de veces que se buscaron \n")
200
201     for i in range(20):
202         print(i + 1 , ".-", lista_ordenada[i][2])
203         print("Cantidad de veces que se buscó este producto: " , lista_ordenada[i][1])
204         print("\n")
205
206 #Creamos una lista donde llamada calificaciones donde vamos guardando
207 #las calificaciones promedio para cada producto su identificador, su nombre
208 #y las veces que fueron devueltas
209     calificaciones = []
210 #Lista ordenada de manera descendente con respecto a las calificaciones
211     calificaciones_ordenadas = []
212
213     for j in range(1,numero_productos + 1):
214         c = 0
215         suma = 0
216         dev = 0
217         for i in range(numero_ventas):
218             if lifestore_sales[i][1] == j:
219                 c += 1
220                 suma += lifestore_sales[i][2]
221                 dev += lifestore_sales[i][4]
222
223         if c ==0:
224             continue
225         else:

```

```

226         calificaciones.append([j,suma / c,lifestore_products[j-1][1],dev])
227 #Ordenamos las calificaciones con respecto a las calificaciones obtenidas de manera
228 #descendente
229     calificaciones_ordenadas = sorted(calificaciones, key = lambda x: x[1], reverse = True)
230
231 #Si se selecciona el valor de "Productos con mejores reseñas" imprimimos
232 #los productos con mejores reseñas
233     if reporte == "Productos con mejores reseñas":
234
235         print("Estos son los 10 productos mejor calificados y su calificación promedio")
236         print("\n")
237
238         for i in range(10):
239             print(i + 1, ".-", calificaciones_ordenadas[i][2])
240             print("Calificación promedio: ", calificaciones_ordenadas[i][1] )
241             print("Cantidad de devoluciones: ", calificaciones_ordenadas[i][3])
242             print("\n")
243
244 # Si se selecciona el valor de "Productos con peores reseñas
245 #Ordenamos de manera ascendente e imprimimos los productos con peores reseñas
246     if reporte == "Productos con peores reseñas":
247
248         peor_calificados = sorted(calificaciones, key = lambda x: x[1])
249
250         print("Estos son los 10 productos peor calificados y la cantidad de veces que se
251 devolvieron ")
252         print("\n")
253
254         for i in range(10):
255             print(i + 1, ".-", peor_calificados[i][2])
256             print("Calificación promedio: ", peor_calificados[i][1] )
257             print("Cantidad de devoluciones: ", peor_calificados[i][3])
258             print("\n")
259
260 #Lista que contiene los meses del año, su numero en el calendario y sus días
261
262     meses = [ ["Enero", "01", 31], ["Febrero", "02", 29], ["Marzo", "03", 31]
263             , ["Abril", "04", 30], ["Mayo", "05", 31], ["Junio", "06", 30]
264             , ["Julio", "07", 31], ["Agosto", "08", 31], ["Septiembre", "09", 30]
265             , ["Octubre", "10", 31], ["Noviembre", "11", 30], ["Diciembre", "12", 31]]
266
267 #Lista que contiene la cantidad de ventas para cada mes
268
269     cantidad_meses = []
270
271     for j in range(len(meses)):
272         c = 0
273         for i in range(len(lifestore_sales)):
274             fecha = lifestore_sales[i][3]
275             mes = fecha[3:5]
276             if mes == meses[j][1]:
277                 c += 1
278             cantidad_meses.append([meses[j][0],c, c / meses[j][2]])
279
280 # Si se selecciona el valor de "Meses con mas ventas" ordenamos
281 # de manera descendente e imprimimos los meses con mas ventas y la cantidad
282 # de ventas que se obtuvieron
283

```



```

284     if reporte == "Meses con mas ventas":
285         lista_ordenada = sorted(cantidad_meses, key = lambda x: x[1], reverse = True)
286
287         for i in range(len(lista_ordenada)):
288             print(lista_ordenada[i][0], "con un total de ventas de ", lista_ordenada[i][1])
289
290 # Si el valor obtenido fue "Ventas promedio por mes,
291 # imprimimos el mes y su promedio de ventas por mes
292
293     if reporte == "Ventas promedio por mes":
294         for i in range(len(cantidad_meses)):
295             print(cantidad_meses[i][0], "con un promedio de: ", cantidad_meses[i][2],
296 "ventas por dia.")
297
298 #Lista que contabiliza los ingresos por cada mes obtenido
299
300     ingresos_por_mes = []
301
302     for m in range(len(meses)):
303         suma = 0
304         for j in range(numero_productos):
305             c = 0
306             for i in range(numero_ventas):
307                 fecha = lifestore_sales[i][3]
308                 num = fecha[3:5]
309                 if meses[m][1] == num:
310                     if lifestore_sales[i][1] == j and lifestore_sales[i][4] == 0:
311                         c += 1
312                         suma += c * lifestore_products[j - 1][2]
313
314             ingresos_por_mes.append(suma)
315
316 #Si se selecciona el valor de "Total de ingresos mensuales" imprimimos
317 #los ingresos mensuales obtenidos
318     if reporte == "Total de ingresos mensuales":
319         for m in range(12):
320             print("El total de ingresos para el mes de ", meses[m][0], "es de
321 ", ingresos_por_mes[m])
322
323 #Si se selecciona el valor de #Total de ingresos anuales
324 #imprimimos los ingresos anuales
325
326     if reporte == "Total de ingresos anuales":
327         suma = sum(ingresos_por_mes)
328
329         print("Los ingresos para el año 2020 fueron de", suma)
330
331     print("Para continuar, escriba: continuar, para salir pulse cualquier tecla")
332     x = input()

```

Solución al problema

Productos más vendidos, más buscados, menos vendidos y menos buscados.

Con el programa anterior logramos obtener los productos más vendidos en la tienda Life Store, así como también los más buscados, estos, los presentamos a continuación.

15 productos más vendidos:

1.- SSD Kingston A400, 120GB, SATA III, 2.5", 7mm

Se vendió 50 veces

2.- Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth

Se vendió 42 veces

3.- Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)

Se vendió 20 veces

4.- Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD

Se vendió 18 veces

5.- SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm

Se vendió 15 veces

6.- Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD

Se vendió 14 veces

7.- Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth

Se vendió 13 veces

8.- Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire

Se vendió 13 veces

9.- SSD XPG SX8200 Pro, 256GB, PCI Express, M.2

Se vendió 11 veces

10.- Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0

Se vendió 9 veces

11.- SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2

Se vendió 9 veces

12.- Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)

Se vendió 7 veces

13.- Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD

Se vendió 6 veces

14.- Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AMD B450, 64GB DDR4 para AMD

Se vendió 6 veces

15.- Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030, 2GB 64-bit GDDR5, PCI Express x16 3.0

Se vendió 5 veces.

20 productos más buscados.

1.- SSD Kingston A400, 120GB, SATA III, 2.5", 7mm

Cantidad de veces que se buscó este producto: 263

2.- SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm

Cantidad de veces que se buscó este producto: 107

3.- Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD

Cantidad de veces que se buscó este producto: 60

4.- Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth

Cantidad de veces que se buscó este producto: 55

5.- Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire

Cantidad de veces que se buscó este producto: 41

6.- Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul

Cantidad de veces que se buscó este producto: 35

7.- TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro

Cantidad de veces que se buscó este producto: 32

8.- Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)

Cantidad de veces que se buscó este producto: 31

9.- Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)

Cantidad de veces que se buscó este producto: 30

10.- SSD XPG SX8200 Pro, 256GB, PCI Express, M.2

Cantidad de veces que se buscó este producto: 30

11.- SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2

Cantidad de veces que se buscó este producto: 27

12.- Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AMD B450, 64GB DDR4 para AMD

Cantidad de veces que se buscó este producto: 25

13.- Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth

Cantidad de veces que se buscó este producto: 24

14.- Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD

Cantidad de veces que se buscó este producto: 23

15.- Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake)

Cantidad de veces que se buscó este producto: 20

16.- Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0

Cantidad de veces que se buscó este producto: 15

17.- Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC, 8GB 128-bit GDDR6, PCI Express 4.0

Cantidad de veces que se buscó este producto: 15

18.- TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Widescreen, Negro

Cantidad de veces que se buscó este producto: 15

19.- Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030, 2GB 64-bit GDDR5, PCI Express x16 3.0

Cantidad de veces que se buscó este producto: 11

20.- SSD Kingston UV500, 480GB, SATA III, mSATA

Cantidad de veces que se buscó este producto: 11

Observamos una relación directamente proporcional entre los productos más buscados y los más vendidos, es decir que, los clientes, entre más buscan los productos, más los compran.

Presentamos a continuación, los productos menos buscados y menos vendidos por categoría, es decir para la categoría “x” presentamos la lista de los menos buscados y los menos vendidos.

Vamos a tomar como ejemplo, la categoría “procesadores”, así que, presentamos los menos vendidos y los menos buscados para esta categoría:

5 productos menos vendidos para la categoría “procesadores”.

1.- Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)

Cantidad de veces que se vendió este producto: 0

2.- Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache

Cantidad de veces que se vendió este producto: 2

3.- Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake)

Cantidad de veces que se vendió este producto: 3

4.- Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake)

Cantidad de veces que se vendió este producto: 4

5.- Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)

Cantidad de veces que se vendió este producto: 7

5 productos menos buscados para la categoría “procesadores”.

1.- Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)

Cantidad de veces que se buscó este producto: 0

2.- Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)

Cantidad de veces que se buscó este producto: 1

3.- Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache

Cantidad de veces que se buscó este producto: 2

4.- Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake)

Cantidad de veces que se buscó este producto: 3

5.- Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake)

Cantidad de veces que se buscó este producto: 4

Notamos que volvemos a observar la misma relación directa, es decir, entre menos se busca un producto, menos se vende, tal es el caso del procesador “Intel Core i3-8100” que se buscó 0 veces y, por ende, se vendió 0 veces.

Observación: Todos los resultados propuestos se pueden demostrar mediante una regresión lineal, pero no nos enfocaremos en eso dados los alcances del presente curso.

Productos mejor calificados y pero calificados por reseña en el servicio.

Productos mejor calificados por reseña en el servicio.

A continuación, mostraremos los 10 productos mejor calificados por reseña en el servicio.

1.- Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache

Calificación promedio: 5.0

Cantidad de devoluciones: 0

2.- Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake)

Calificación promedio: 5.0

Cantidad de devoluciones: 0

3.- Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)

Calificación promedio: 5.0

Cantidad de devoluciones: 0

4.- Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake)

Calificación promedio: 5.0

Cantidad de devoluciones: 0

5.- Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Express 3.0

Calificación promedio: 5.0

Cantidad de devoluciones: 0

6.- Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC, 8GB 128-bit GDDR6, PCI Express 4.0

Calificación promedio: 5.0

Cantidad de devoluciones: 0

7.- Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC, 4GB 128-bit GDDR5, PCI Express x16 3.0

Calificación promedio: 5.0

Cantidad de devoluciones: 0

8.- Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming, 8GB 128-bit GDDR6, PCI Express 4.0

Calificación promedio: 5.0

Cantidad de devoluciones: 0

9.- Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti, 6GB 192-bit GDDR6, PCI Express x16 3.0

Calificación promedio: 5.0

Cantidad de devoluciones: 0

10.- Tarjeta Madre Gigabyte XL-ATX TRX40 Designare, S-sTRX4, AMD TRX40, 256GB DDR4 para AMD

Calificación promedio: 5.0

Cantidad de devoluciones: 0

Notamos que, aunque el procesador “AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache” es el mejor calificado, (o al menos el que aparece al inicio) este, aparece en la lista de los productos menos buscados y por ende, en los menos vendidos, lo cual podría sugerirnos una problemática de difusión de productos, ya que, este, según los clientes que lo adquirieron, es un producto muy bueno.

Productos peor calificados por reseña en el servicio

1.- Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0

Calificación promedio: 1.0

Cantidad de devoluciones: 1

2.- Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel

Calificación promedio: 1.0

Cantidad de devoluciones: 1

3.- Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD

Calificación promedio: 1.8333333333333333

Cantidad de devoluciones: 3

4.- Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel

Calificación promedio: 2.0

Cantidad de devoluciones: 1

5.- Cougar Audífonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro.

Calificación promedio: 3.0

Cantidad de devoluciones: 0

6.- MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0

Calificación promedio: 4.0

Cantidad de devoluciones: 0

7.- Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix, 4GB 128-bit GDDR5, PCI Express 3.0

Calificación promedio: 4.0

Cantidad de devoluciones: 0

8.- HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro, Inalámbrico, USB, 3.5mm, Negro

Calificación promedio: 4.0

Cantidad de devoluciones: 0

9.- Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD

Calificación promedio: 4.142857142857143

Cantidad de devoluciones: 1

10.- Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth

Calificación promedio: 4.230769230769231

Cantidad de devoluciones: 1

Aquí notamos que efectivamente, los productos peor calificados ni si quiera figuran en las listas de más vendidos y más buscados. Excepto por “Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD”, pero la devolvieron 3 veces, lo cuál podría indicar que no es un producto muy bueno.

Estrategia para retirar productos del mercado y sugerencia para reducir la acumulación del mercado.

Derivado de los resultados de los puntos anteriores, podríamos sugerir que, a los productos con calificaciones no tan bajas o con buenas calificaciones que no figuren entre los más buscados los promoviéramos más, ya sea mediante publicidad, o mediante descuentos.

También para tratar la problemática de acumulación de inventario, lo dicho anteriormente resultaría en la venta de los productos, reduciendo así el inventario, para los productos que realmente podríamos calificarlos como malos, podemos sugerir que entren en algún programa de ofertas y descuento.

Estas ofertas y descuentos podrían entrar en función a partir del mes de agosto donde podemos ver a continuación que es a partir de este mes donde las ventas se reducen.

Mes	Ventas promedio	Total de ventas	Ingresos mensuales
Enero	1.7	53 \$	499,249.00
Febrero	1.41	41 \$	159,527.00
Marzo	1.64	51 \$	287,580.00
Abril	2.5	75 \$	593,560.00
Mayo	1.16	36 \$	143,795.00
Junio	0.36	11 \$	42,247.00
Julio	0.35	11 \$	33,316.00
Agosto	0.096	3 \$	3,336.00
Septiembre	0.03	1 \$	-
Octubre	0	0 \$	-
Noviembre	0.03	1 \$	-
Diciembre	0	0	
Total anual	0.775342466	283 \$	1,762,610.00

Con lo propuesto podríamos aumentar el promedio de ventas por día y así aumentar los ingresos mensuales para estos meses, logrando así un aumento en el total anual de ventas.

Conclusión

Con el presente análisis logramos sugerir una estrategia para el correcto manejo de la mercancía en una tienda de tecnología, logramos dar respuesta y aprendimos a manejar volúmenes de datos no muy grandes, pero eficientes, también logramos perfeccionar nuestro entendimiento acerca de la programación y, sobre todo, obtuvimos una experiencia más.

Una buena organización, planeación y optimización pueden conducirnos a resultados buenos para todos.