



EMTECH EMERGING TECHNOLOGIES INSTITUTE  
PYTHON FOR DATA SCIENCE

## Proyecto Final de Fundamentos de Programación con Python

Rodrigo González Cáceres

13 de febrero de 2022. Monterrey, N.L.

## INDICE

<b>1. Introducción.....</b>	<b>3</b>
<b>2. Definición del Código.....</b>	<b>3</b>
2.1 Parte 1.....	3
2.2 Parte 2.....	6
2.3 Parte 3.....	7
<b>3. Solución al Problema.....</b>	<b>10</b>
3.1 Parte 1.....	10
3.2 Parte 2.....	12
3.3 Parte 3.....	12
<b>4. Conclusión.....</b>	<b>14</b>
<b>5. Anexos.....</b>	<b>16</b>

## 1. Introducción

LifeStore es una tienda virtual la cual cuenta con una variedad de productos. El Gerente de Ventas identificó que hay un exceso de acumulación de inventario y una reducción de ventas en el último semestre. LifeStore requiere del apoyo del análisis de sus datos para poder identificar la razón de esta problemática, así como obtener propuestas de soluciones que ayuden a la empresa a recuperar sus ventas y disminuir el inventario.

Este proyecto será realizado 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. Con estas operaciones se podrán resolver algunas surgidas tras esta problemática y brindará un fuerte sustento para las propuestas obtenidas. A continuación, se explicará el código utilizado a detalle para la obtención de los resultados, seguido por la solución al problema y finalmente las conclusiones.

## 2. Definición del Código

### 2.1 Parte 1

Primeramente, se importan las listas a utilizar del archivo de reporte de LifeStore. Después, aparece un input para poder iniciar sesión, donde el único usuario activo es “Rgonzalez” con la contraseña “123”. En caso de que el usuario o contraseña sea incorrecto, se arrojará un mensaje diciendo que está equivocado.

```
1  from lifestore_file import lifestore_products
2  from lifestore_file import lifestore_sales
3  from lifestore_file import lifestore_searches
4
5  intentos = 0
6  contraseñaR = "123"
7  usuarios = "Rgonzalez"
8
9  print('Inicio de sesion')
10 for usuario in usuarios:
11     usuario = input('Usuario: ')
12     if usuario != usuarios:
13         print("Usuario no reconocido")
14     else:
15         break
16 for contraseña in contraseñas:
17     contraseña = input('Contraseña: ')
18     if contraseña != contraseñaR:
19         intentos += 1
20         print("Intenta de nuevo")
21         print("Número de intentos: " + str(intentos))
22     else:
23         print("")
24         print(";Bienvenido de vuelta!")
25     break
```

Se creó una variable diferente para cada artículo empezando con Prod1 y terminando en Prod94, así como la variable num1. En el código siguiente se utiliza las funciones “for” y “if” para poder hacer modificaciones en las variables de los productos de conforme al artículo que

se esté analizando en esa línea de la lista. Donde la variable del producto incrementará por 1 unidad con el fin de contar cuantas unidades ocurren y la variable num1 también aumenta una unidad para que en el siguiente “loop” avance en la siguiente línea.

```

79 #Loop para sumar cuantas veces se vendió cada artículo
80 for ventas in lifestore_sales:
81     if lifestore_sales[num1][1] == 1:
82         prod1 += 1
83         num1 += 1
84         continue
85     elif lifestore_sales[num1][1] == 2:
86         prod2 += 1
87         num1 += 1
88         continue
89     elif lifestore_sales[num1][1] == 3:
90         prod3 += 1
91         num1 += 1
92         continue
93     elif lifestore_sales[num1][1] == 4:
94         prod4 += 1
95         num1 += 1
96         continue
97     elif lifestore_sales[num1][1] == 5:
98         prod5 += 1
99         num1 += 1
100        continue

```

Al final de este código, se termina con un “break” para cerrar el código.

```

245         elif lifestore_sales[num1][1] == 94:
246             prod94 += 1
247             num1 += 1
248             continue
249             break

```

Para una mejor visualización de resultados, se acomodan como en la siguiente lista.

```

sales2 = [['Producto 1', prod1], ['Producto 2', prod2],
['Producto 3', prod3],
['Producto 4', prod4], ['Producto 5', prod5],
['Producto 6', prod6],
['Producto 7', prod7], ['Producto 8', prod8],
['Producto 10', prod10], ['Producto 11', prod11],
['Producto 12', prod12], ['Producto 13', prod13],
['Producto 17', prod17], ['Producto 18', prod18],
['Producto 21', prod21], ['Producto 22', prod22],

```

Después, se ordena la lista de cierto modo en el que se pueda apreciar tanto de mayor a menor como de menor a mayor los resultados con base en la cantidad de ventas. Se define “segundo” para poder indicar de la lista, que elemento tomar en cuenta para poder ordenar, en este caso se buscaba ser ordenado con base en las ventas. La opción de reverse es para indicar que se ordenara de mayor a menor debido a que por modo predeterminado la función “sort” ordena en orden ascendente. Por último, al momento de imprimir, se indica con [:5] para que solamente muestre los primeros 5 resultados.

```

278 #definir el valor "segundo"
279 def segundo(elem):
280     return elem[1]
281
282 print("")
283 print("Resultado 1.1.1")
284 #ordenar tomando en cuenta el segundo elemento
285 sales2.sort(key=segundo, reverse=True)
286 print("Lista de Productos con sus Respectivas Cantidades
Vendas de Mayor a Menor:")
287 print("")
288 print(sales2[:5])

```

Se utilizó un código idéntico al anterior para poder identificar la cantidad de veces que se buscó cada artículo, haciendo uso de las funciones “for” y “if”. Donde la variable de los productos incrementa conforme en la lista aparecen como buscados y de igual manera la variable num1.

```

#Loop para sumar cuantas veces se buscó cada artículo
for busquedas in lifestore_searches:
    if lifestore_searches[num1][1] == 1:
        prod1 += 1
        num1 += 1
        continue
    elif lifestore_searches[num1][1] == 2:
        prod2 += 1
        num1 += 1
        continue
    elif lifestore_searches[num1][1] == 3:
        prod3 += 1
        num1 += 1
        continue

```

Se acomoda la lista al igual que para el análisis de ventas utilizando la función “sort” con y sin los elementos de “key” y “reverse”.

```

print("Lista de Veces Buscadas:")
print("")
print(busquedas)
busquedas2 = [['Producto 1', prod1], ['Producto 2', prod2],
['Producto 3', prod3],
['Producto 4', prod4], ['Producto 5', prod5],
['Producto 6', prod6],
['Producto 7', prod7], ['Producto 8', prod8],
['Producto 9', prod9],
['Producto 10', prod10], ['Producto 11', prod11],
['Producto 12', prod12], ['Producto 13', prod13],
['Producto 15', prod15],
['Producto 17', prod17], ['Producto 18', prod18],

663 def segundo(elem):
664     return elem[1]
665
666 print("")
667 print("Resultado 1.1.2")
668 busquedas2.sort(key=segundo, reverse=True)
669 print("Lista de Productos con sus Respectivas Cantidades
de Busquedas de Mayor a Menor:")
670 print("")
671 print(busquedas2[:10])

```

Para los resultados por categoría, se creó una serie de loops para poder obtener los resultados de ventas de los 5 productos por categoría con menores ventas. El programa no soportó la cantidad de iteraciones necesarias por lo que se queda a mitad de código. Revisando detalle a detalle el código, la única explicación encontrada fue que el programa no tiene la potencia suficiente para hacer esa cantidad de iteraciones y arrojar los resultados completos.

```
293 categorias = []
294
295 for categoria in lifestore_products:
296     if lifestore_products[num2][0] < 97:
297         cat = lifestore_products[num2][3]
298         categorias.append(cat)
299         num2 += 1
300     else:
301         break
302 categorias = list(dict.fromkeys(categorias))
303
304 sales2.sort(key=segundo)
305 print("Lista de Categorías con sus Respectivas Cantidades")
306 print("Vendas de Menor a Mayor:")
307 print("")
308 num2 = 0
309 num3 = 1
310 for producto in lifestore_products:
311     if lifestore_products[num2][0] == num3:
312         lifestore_products[num2].append(sales[num2])
313         num3 += 1
314     elif num2 <= 96:
315         num2 += 1
```

## 2.2 Parte 2

Se utilizó un doble “if” para poder crear un “loop” en el cual la primera condición es si el artículo se devolvió o no. En caso de que sí se devuelva, si activa el siguiente “if”, el cual ahora tiene la condición de que id del artículo se esta utilizando. Una vez que estas dos condiciones se activaron, se suma una unidad al respectivo producto y a num1 para pasar al siguiente.

```
803 #Loop para sumar cuantas veces se vendió cada artículo
804 for devoluciones in lifestore_sales:
805     if lifestore_sales[num1][4] == 1:
806         if lifestore_sales[num1][1] == 1:
807             prod1 += 1
808             num1 += 1
809             continue
810         elif lifestore_sales[num1][1] == 2:
811             prod2 += 1
812             num1 += 1
813             continue
814         elif lifestore_sales[num1][1] == 3:
815             prod3 += 1
816             num1 += 1
817             continue
```

En caso de que no se haya devuelto el artículo, se pasa a la siguiente condición la cual es “else” para que la única acción que realice sea pasar a la siguiente línea de la lista.

```

974     else:
975         num1 += 1
976         continue
977         break

```

Finalmente se acomodan y ordenan los resultados.

```

1036 print(devoluciones)
1037 devoluciones2 = [['Producto 1', prod1], ['Producto 2',
1038 prod2], ['Producto 3', prod3],
1039 ['Producto 4', prod4], ['Producto 5', prod5],
1040 ['Producto 6', prod6],
1041 ['Producto 7', prod7], ['Producto 8', prod8],
1042 ['Producto 10', prod10], ['Producto 11', prod11],
1043 ['Producto 12', prod12], ['Producto 13', prod13],
1044 ['Producto 17', prod17], ['Producto 18', prod18],

```

Se imprimen los resultados y para el 2.1.2, se elaboró un “loop” para poder excluir al momento de imprimir, los valores que fueran 0, de esta forma solo se consideran valores con devoluciones.

```

929 def segundo(elem):
930     return elem[1]
931
932 print("")
933 print("Resultado 2.1.1")
934 devoluciones2.sort(key=segundo, reverse=True)
935 print("Lista de Productos con sus Respectivas
936 Devoluciones de Mayor a Menor:")
937 print("")
938 print(devoluciones2[:5])
939
940 print("")
941 print("Resultado 2.1.2")
942 devoluciones2.sort(key=segundo)
943 print("Lista de Productos con sus Respectivas
944 Devoluciones de Menor a Mayor:")
945 print("")
946
947 num1 = 0
948 num2 = 0
949 for dev in devoluciones2:
950     if devoluciones2[num1][1] > 0 and num2 <= 4:
951         print(str(devoluciones2[num1][0])+"": "+str
952 (devoluciones2[num1][1]))
953         num1 += 1
954         num2 += 1
955     else:
956         num1 += 1

```

### Parte 3

Para la tercera parte, primero se creó un “loop” con el que se acumula la cantidad de artículos vendidos por cada id.

```

1610 num2 = 0
1611 #Multiplicacion de cantidad de productos vendidos por su
    respectivo precio
1612 for ventas2 in lifestore_products:
1613     if lifestore_products[num2][0] == 1:
1614         prod1 = prod1*lifestore_products[num2][2]
1615         num2 += 1
1616         continue
1617     elif lifestore_products[num2][0] == 2:
1618         prod2 = prod2*lifestore_products[num2][2]
1619         num2 += 1
1620         continue

```

Después, se utiliza un “loop” para multiplicar la cantidad de productos vendidos por su respectivo precio.

```

1778 num2 = 0
1779 #Multiplicacion de cantidad de productos vendidos por su
    respectivo precio
1780 for ventas2 in lifestore_products:
1781     if lifestore_products[num2][0] == 1:
1782         prod1 = prod1*lifestore_products[num2][2]
1783         num2 += 1
1784         continue
1785     elif lifestore_products[num2][0] == 2:
1786         prod2 = prod2*lifestore_products[num2][2]
1787         num2 += 1
1788         continue
1789     elif lifestore_products[num2][0] == 3:
1790         prod3 = prod3*lifestore_products[num2][2]
1791         num2 += 1
1792         continue

```

Se imprimen los resultados.

```

1951 print("")
1952 print("Ventas Anuales por Producto en $:")
1953 print("")
1954 print("Producto 1: $" + str(prod1))
1955 print("Producto 2: $" + str(prod2))
1956 print("Producto 3: $" + str(prod3))
1957 print("Producto 4: $" + str(prod4))

```

Se realizó una sumatoria de todas las ventas para tener una variable total con las ventas netas totales. Después de obtener ese valor, se divide entre 12 para obtener un promedio de venta mensual.

```

1997 Total = prod1 + prod2 + prod3 + prod4 + prod5 + prod6 +
    prod7 + prod8 + prod10 + prod11 + prod12 + prod13 + prod17
    + prod18 + prod21 + prod22 + prod25 + prod28 + prod29 +
    prod31 + prod33 + prod40 + prod42 + prod44 + prod45 +
    prod46 + prod47 + prod48 + prod49 + prod50 + prod51 +
    prod52 + prod54 + prod57 + prod60 + prod66 + prod67 +
    prod74 + prod74 + prod84 + prod85 + prod89 + prod94
1998
1999 print("")
2000 print("Total Ventas Anuales: $" + str(Total))
2001 Promedio_Mensual = Total/12
2002 print("")
2003 print("Promedio Mensual Anual: $" + str(Promedio_Mensual))

```

Se creó una variable de “valor” la cual tomará el valor del precio de venta de cada artículo y se utilizó la función “if” y “for” para poder asignar ese valor dependiendo del artículo al que le pertenezca.



```

2046 valor94 = 0
2047
2048 num0 = 0
2049
2050 for precio in lifestore_products:
2051     if lifestore_products[num0][0] == 1:
2052         valor1 = lifestore_products[num0][2]
2053         num0 += 1
2054         continue
2055     elif lifestore_products[num0][0] == 2:
2056         valor2 = lifestore_products[num0][2]
2057         num0 += 1
2058         continue

```

Se definió variables para cada mes y valor de cantidad de cada mes.

```

2263 print("Producto 94: " + str(valor94))
2264
2265
2266 num3 = 0
2267 Enero = 0
2268 Febrero = 0
2269 Marzo = 0
2270 Abril = 0
2271 Mayo = 0
2272 Junio = 0
2273 Julio = 0
2274 Agosto = 0
2275 Septiembre = 0
2276 Octubre = 0
2277 Noviembre = 0
2278 Diciembre = 0
2279
2280 CENE = 0
2281 CFEB = 0
2282 CMAR = 0
2283 CABR = 0
2284 CMAY = 0
2285 CJUN = 0

```

Se creo un “loop” don doble “if” donde la primera condición identifica que mes se esta utilizando, después agrega un valor a la variable de cantidad de cada mes. La siguiente condición es para agregar el respectivo valor su respectivo mes, el cual fue obtenido del “loop” de valores.

```

2296 #Agregar el respectivo valor a cada mes
2297 for mes in lifestore_sales:
2298     if lifestore_sales[num3][3][4] == "1":
2299         CENE += 1
2300         if lifestore_sales[num3][1] == 1:
2301             Enero += valor1
2302             num3 += 1
2303             continue
2304         elif lifestore_sales[num3][1] == 2:
2305             Enero += valor2
2306             num3 += 1
2307             continue

```

Finalmente, se imprime el valor de cada mes, la cantidad de ocurrencias de cada mes y se obtiene un promedio de cada mes dividiendo el valor total de venta vendida entre la cantidad de ventas que tuvo cada mes.

```

4372     print(CNOV)
4373     print(CDIC)
4374
4375     print("")
4376     print("Promedio de Venta por Mes:")
4377     print("")
4378     print("Enero: "+str(Enero/CENE))
4379     print("Febrero: "+str(Febrero/CFEB))
4380     print("Marzo: "+str(Marzo/CMAR))
4381     print("Abril: "+str(Abril/CABR))
4382     print("Mayo: "+str(Mayo/CMAY))
4383     print("Junio: "+str(Junio/CJUN))
4384     print("Julio: "+str(Julio/CJUL))
4385     print("Agosto: " +str(Agosto/CAGO))
4386     print("Septiembre: "+str(Septiembre/CSEP))

```

### 3. Solución al Problema

#### 3.1 Parte 1.

Para el primer punto, se obtiene los siguientes resultados en cuanto a la venta de los artículos.

```

Resultado 1.1.1
Lista de Productos con sus Respectivas Cantidades Vendidas de M
ayor a Menor:

[['Producto 54', 50], ['Producto 3', 42], ['Producto 5', 20], [
'Producto 42', 18], ['Producto 57', 15]]

Resultado 1.2.1
Lista de Categorías con sus Respectivas Cantidades Vendidas de
Menor a Mayor:

Programa no tiene suficiente capacidad como para correr todo el
código, se queda atorado a la mitad de la corrida
procesadores
Producto 1 / Vendidos: 2
Producto 2 / Vendidos: 13
Producto 3 / Vendidos: 42
Producto 4 / Vendidos: 13
Producto 5 / Vendidos: 20
Producto 6 / Vendidos: 3
Producto 7 / Vendidos: 7
Producto 8 / Vendidos: 4
Producto 9 / Vendidos: 1
tarjetas de video
Producto 10 / Vendidos: 3
Producto 11 / Vendidos: 9
Producto 12 / Vendidos: 1
Producto 13 / Vendidos: 0
Producto 14 / Vendidos: 0

```

```

Producto 15 / Vendidos: 0
Producto 16 / Vendidos: 1
Producto 17 / Vendidos: 5
Producto 18 / Vendidos: 0
Producto 19 / Vendidos: 0
Producto 20 / Vendidos: 2
Producto 21 / Vendidos: 1
Producto 22 / Vendidos: 0
Producto 23 / Vendidos: 0
Producto 24 / Vendidos: 2
Producto 25 / Vendidos: 0
Producto 26 / Vendidos: 0
Producto 27 / Vendidos: 1
Producto 28 / Vendidos: 14
tarjetas madre
Producto 29 / Vendidos: 0
Producto 30 / Vendidos: 6
Producto 31 / Vendidos: 0
Producto 32 / Vendidos: 2
Producto 33 / Vendidos: 0
Producto 34 / Vendidos: 0
Producto 35 / Vendidos: 0
Producto 36 / Vendidos: 0
Producto 37 / Vendidos: 0
Producto 38 / Vendidos: 0
Producto 39 / Vendidos: 1
Producto 40 / Vendidos: 0
Producto 41 / Vendidos: 18
Producto 42 / Vendidos: 0
Producto 43 / Vendidos: 6
Producto 44 / Vendidos: 1
Producto 45 / Vendidos: 1
Producto 46 / Vendidos: 11
tarjetas madre
Producto 47 / Vendidos: 9
Producto 48 / Vendidos: 3
Se pone limitante a 49 porque programa no tiene capacidad de co
rrer más líneas

```

Se aprecia que el artículo más vendido es el Producto 54 con 50 ventas, seguido por el Producto 3 con 42 ventas, el Producto 5 con 20 ventas, el Producto 42 con 18 y el Producto 57 con 15. Los productos con menos ventas son muchos igualados con una unidad, estos son el Producto 10, 13, 17, 22, 28, 40, 45, 46, 50, 60, 66, 67, 84, 89, y 94.

En cuanto a las categorías, el programa no soportó la cantidad de iteraciones necesarias por lo que se queda a mitad de código. Revisando detalle a detalle el código, la única explicación encontrada fue que el programa no tiene la potencia suficiente para hacer esa cantidad de iteraciones y arrojar los resultados completos. Por lo que se aprecia solo los resultados hasta la categoría de “tarjeta madre”

Como resultado de las cantidades, los productos que más búsquedas tuvieron fueron los productos 54 con 263 búsquedas, producto 57 con 107, producto 29 con 60 y el producto 3 con 55.

```
Resultado 1.1.2
Lista de Productos con sus Respektivas Cantidades de Busquedas
de Mayor a Menor:

[['Producto 54', 263], ['Producto 57', 107], ['Producto 29', 60],
 ['Producto 3', 55], ['Producto 4', 41], ['Producto 85', 35],
 ['Producto 67', 32], ['Producto 7', 31], ['Producto 5', 30], ['Producto 47', 30]]
```

### 3.2 Parte 2

Se creó una lista sobre las devoluciones de los productos tanto de mayor a menor como de menor a mayor. El Producto 31 es el producto con más devoluciones con 3, y el Producto 2, Producto 17, Producto 29, y Producto 45 solamente tienen 1 devolución cada uno y son todos los Productos que tuvieron devoluciones. Y de menores, solo un producto adicional tuvo devoluciones, que fue el Producto 46.

```
Resultado 2.1.1
Lista de Productos con sus Respektivas Devoluciones de Mayor a
Menor:

[['Producto 31', 3], ['Producto 2', 1], ['Producto 17', 1], ['Producto 29', 1], ['Producto 45', 1]]

Resultado 2.1.2
Lista de Productos con sus Respektivas Devoluciones de Menor a
Mayor:

Producto 2: 1
Producto 17: 1
Producto 29: 1
Producto 45: 1
Producto 46: 1
```

### 3.3 Parte 3

Se obtuvo las ventas anuales por producto en pesos donde la sumatoria de todas las ventas da un total de \$768,655. El promedio mensual de ventas es de \$64,054.

Ventas Anuales por Producto en \$:

Producto 1: \$6038  
Producto 2: \$54717  
Producto 3: \$129738  
Producto 4: \$28717  
Producto 5: \$35580  
Producto 6: \$35427  
Producto 7: \$59913  
Producto 8: \$21596  
Producto 10: \$889  
Producto 11: \$22197  
Producto 12: \$59571  
Producto 13: \$3989  
Producto 17: \$4199  
Producto 18: \$10995  
Producto 21: \$10318  
Producto 22: \$3429  
Producto 25: \$11058  
Producto 28: \$9579  
Producto 29: \$34986  
Producto 31: \$13374  
Producto 33: \$8538  
Producto 40: \$17439  
Producto 42: \$32022  
Producto 44: \$16554  
Producto 45: \$2869  
Producto 46: \$1539  
Producto 47: \$13299  
Producto 48: \$23031  
Producto 49: \$9417

Producto 50: \$2949  
Producto 51: \$7197  
Producto 52: \$11318  
Producto 54: \$12950  
Producto 57: \$13335  
Producto 60: \$2519  
Producto 66: \$8049  
Producto 67: \$3229  
Producto 74: \$8478  
Producto 84: \$1089  
Producto 85: \$4318  
Producto 89: \$859  
Producto 94: \$2869

Total Ventas Anuales: \$768655

Promedio Mensual Anual: \$64054.583333333336

La sumatoria de venta total fue la siguiente, siendo abril el mes con mayor venta neta con \$193,295, seguido por marzo con \$164,729 y enero con \$124,446. Los meses de octubre, noviembre y diciembre no contaron con ventas en el listado marcado. La excepción es en noviembre que tuvo una venta, sin embargo, no se consideró en el análisis debido a que esa venta está registrada en el año 2019 a diferencia de los demás años, los cuales son registrados a partir de enero del 2020.

```
Ventas Totales por Mes:
Enero: $124446
Febrero: $110139
Marzo: $164729
Abril: $193295
Mayo: $96394
Junio: $36949
Julio: $26949
Agosto: $3077
Septiembre: $4199
Octubre: $0
Noviembre: $0
Diciembre: $0
```

Una vez obtenido estos resultados, se hizo el mismo análisis, pero ahora considerando solamente la cantidad de artículos vendidos al mes. Abril es también en cuanto a cantidad de ventas el más alto, pero a diferencia de ventas netas, enero tuvo mayores ventas que marzo, solo que en precio fue menor.

```
Articulos Vendidos Totales por Mes:
Enero: 54
Febrero: 41
Marzo: 51
Abril: 75
Mayo: 36
Junio: 11
Julio: 11
Agosto: 3
Septiembre: 1
Octubre: 0
Noviembre: 0
Diciembre: 0
```

El promedio de venta por artículo por mes se muestra en la siguiente tabla. Junio fue el mes con promedio por venta más alto con \$3,359, seguido por marzo con \$3,229.98

```
Promedio de Venta por Mes:
Enero: $2304.5555555555557
Febrero: $2686.317073170732
Marzo: $3229.9803921568628
Abril: $2577.2666666666667
Mayo: $2677.6111111111113
Junio: $3359.0
Julio: $2449.909090909091
Agosto: $1025.6666666666667
Septiembre: $4199.0
```

#### 4. Conclusiones

Con el análisis de la primera parte, se puede concluir que hay muchos productos en el catálogo los cuales no generar ingresos significativos o para nada de ingresos. Es importante tomar en consideración estos análisis debido a que los productos con poco movimiento seguramente están generando costos de inventario además de que se mantienen como costo activo con depreciación debido a su baja movilidad. Con base en los resultados obtenidos, se recomienda dar de baja los artículos que tuvieron ventas nulas e iguales o menores a 5 en este periodo de tiempo.

Sin embargo, no es el único punto a tomar en consideración. La cantidad de búsquedas de los artículos influyen mucho también. Con el análisis de las búsquedas, se aprecia que solo cuatro artículos superan las 50 búsquedas y coincide que el artículo 54 fue el producto con mayores ventas y mayores búsquedas. Esto significa que para estos cuatro productos no requieran de una atención tan especial como lo que se debería de tomar en consideración con los demás productos. Se deben de crear estrategias de venta para fomentar el crecimiento de búsquedas y ventas. Por ejemplo, crear promociones de artículos no muy solicitados con los más vendidos, hacer más publicidad de los mismos, poner descuentos o incluso hasta el formato de la página puede influenciar en las ventas de acuerdo a como esté acomodado.

Con el análisis de la segunda parte, se muestra el indicador de devoluciones y reseñas de los artículos. Es muy buen resultado el que en todo ese periodo de tiempo solamente 7 artículos han sido devueltos y todos con una sola devolución a excepción del Producto 31 con 3 devoluciones. Aunque las cantidades son bajas, hay que prestar atención en los motivos del porqué se está devolviendo, sobre todo el Producto 31. Puede deberse a distintos motivos, como el que la calidad no era la esperada, el producto estaba defectuoso/podrido, se mostró por mucha diferencia mejor el producto en la página en línea que lo que es físicamente, entre otras. Por lo que la empresa debe de ajustar su oferta con las expectativas reales del cliente y vender los productos como son e inspeccionar que no traigan consigo algún defecto.

Aunque las devoluciones sean pocas, los resultados pueden ser un poco confusos. Esto se debe a que varios productos tienen reseñas bajas por debajo de 3, de igual manera son la minoría, pero es importante saber por qué están teniendo tan malos resultados. Para los productos que se decida mantener y tengan reseñas bajas, se debe de evaluar a calidad de los productos y cambiarlos o poner más filtros para garantizar la satisfacción del cliente.

Con base en el análisis de la tercera parte, se aprecia que, al hacer un promedio mensual de ventas, se refleja que en promedio se vende alrededor de \$64,000 al mes. Sin embargo, una vez obteniendo las ventas al mes, se aprecia que están muy dispersas las ventas donde hay meses con ventas de 4 dígitos y meses con ventas de 6 dígitos. Por lo que se debe de prestar atención a los meses con menores ventas, por lo que se podría crear una campaña para que los meses que tengan menores ventas, cuenten con más ofertas que promuevan las ventas en la tienda y sea más atractivo para los consumidores.

Se deben de establecer parámetros por parte de la empresa para poder determinar que productos dar de baja, cuanta cantidad de inventario contar para producto y como ofrecerlo al público. Lo sugerido con este análisis y tomando en consideración la comparación con otros productos es eliminar los productos con ventas menores a 5 y que tengan búsquedas menores a 2. Además, eliminar los productos que tengan reseñas con valor de 2 y 1. Y, por último, identificar en que meses es importante crear una campaña para incentivar las ventas, especialmente en los meses desde julio hasta diciembre, los cuales son los meses con bajas ventas. Esto pensando como reacción rápida para la operación del negocio, para futuro análisis

con mayor valor, sería realizar un análisis del canasto de compra, debido a que hay productos que puede que no tengan muy buena reputación, muchas búsquedas o incluso muchas ventas, pero esos productos incitan a comprar otros, ya sea en promoción o simplemente porque estaban mostrados a lado o cualquier otro motivo que ayude a vender.

## **5. Anexos**

1. Liga para archivo en GITHUB

<https://github.com/Rgonzalez247/Emtech>