EMTECH
Emerging Technologies Institute

Santander

PROYECTO 1

Introducción a Python

REPORTE FINAL

RAUL ANTONIO ALVAREZ CRUZ

GPO. 3

Tutor: Javier Ramirez

Índice

Introducción

El presente documento es el producto final de la primera etapa del proyecto sobre los fundamentos de Python. Se presentará el trabajo integrador, 394 líneas de código en donde, bajo la consigna de utilizar los principales operadores lógicos se resolvieron algunos problemas para la empresa LifeStore.

Algunos de los problemas resueltos a través del uso de ciencia de datos con Python fueron la identificación de los productos más vendidos, los menos vendidos, los de mejor y peor calificación, los ingresos mensuales, su promedio y el ingreso neto anual.

Con base en esa información fue posible hacer un análisis que sea de utilidad a la empresa sobre sus productos y emitir recomendaciones para optimizaciones en la empresa, según la información que arrojaron los números. Se explicará previamente en qué consiste, a grandes rasgos el código y finalmente se harán las recomendaciones pertinentes.

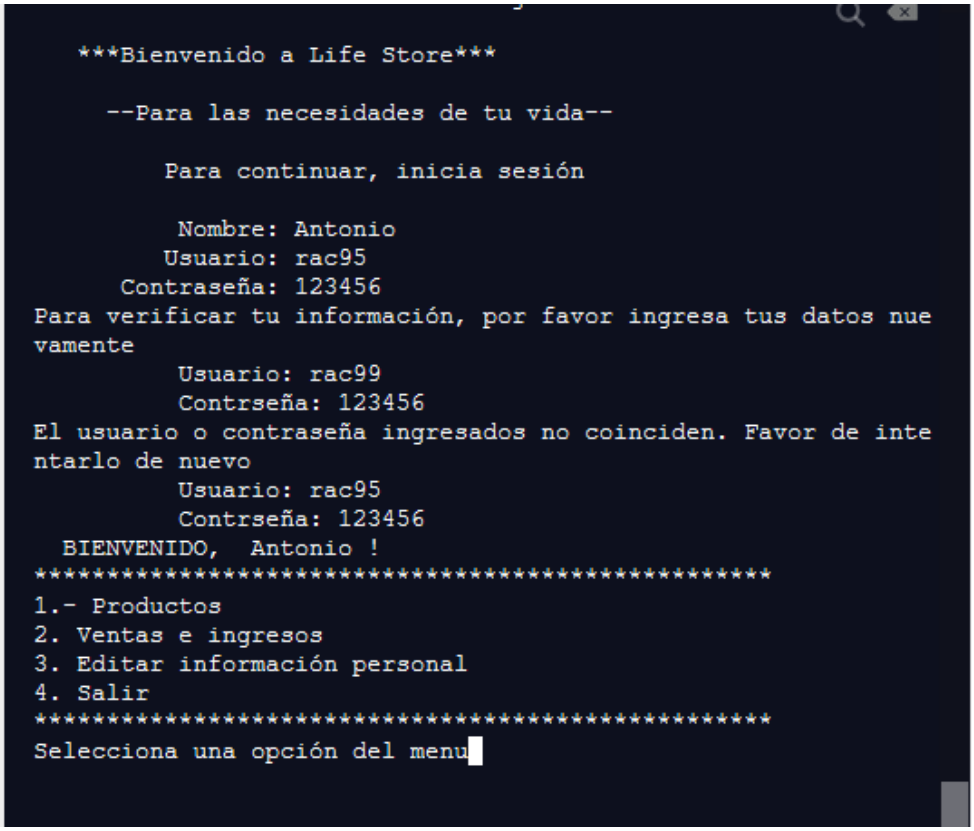
Definición de código

```python
from StoreData import lifestore_products, lifestore_sales, lifestore_searches

print("\n   ***Bienvenido a Life Store***\n")
print("     --Para las necesidades de tu vida--\n")

print("         Para continuar, inicia sesión\n")

nombre = input("          Nombre: ")
usuario = input("         Usuario: ")
contrasena = input("      Contraseña: ")

print("Para verificar tu información, por favor ingresa tus datos nuevamente")

while True:
    verif_usuario = input("          Usuario: ")
    verif_contrasena = input("          Contrseña: ")
    if verif_usuario == usuario and verif_contrasena == contrasena:
        print("  BIENVENIDO, ", nombre,"!")
        break
    else:
        print("El usuario o contraseña ingresados no coinciden. Favor de intentarlo de nuevo")

```

```
<built-in method count of list object at 0x7f8d44f8f680>

   ***Bienvenido a Life Store***

     --Para las necesidades de tu vida--

         Para continuar, inicia sesión

          Nombre: Antonio
         Usuario: rac95
      Contraseña: 123456
Para verificar tu información, por favor ingresa tus datos nue
vamente
          Usuario:
```

En la primera parte del código se importa la información del archivo principal con la información sobre los productos, las ventas y las búsquedas. El código continúa e imprime un mensaje de bienvenida y luego pide un registro de información para iniciar la sesión y poder proceder. Este acceso funciona como una especie de verificación a dos pasos, donde el usuario se registra con nombre usuario y contraseña y luego tiene que confirmar su información para poder proceder al menú principal.

Si la información proporcionada durante el proceso de verificación coincide con los datos de la primera parte, entonces el usuario tendrá acceso a ver las listas a través del menú principal, de otra manera el sistema informará del error y continuará hasta que la información coincida.

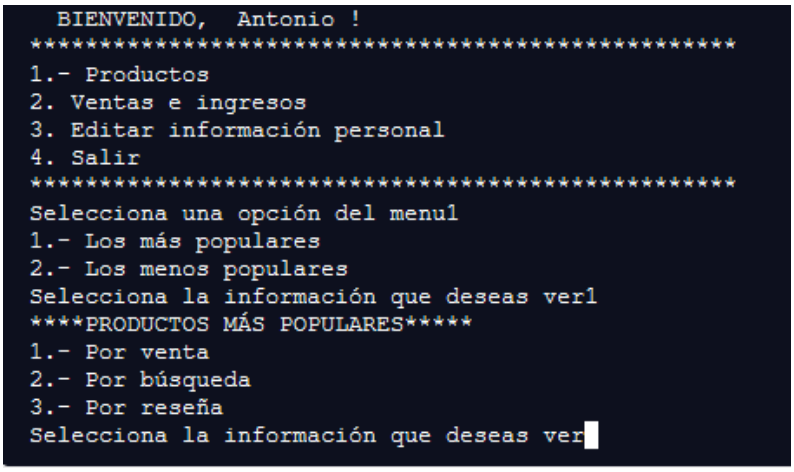



```
("**************************************************")
print("1.- Productos")
print("2. Ventas e ingresos")
print("3. Editar información personal")
print("4. Salir")
print
("**************************************************")

sales = []
while True:
    menu_op = int(input("Selecciona una opción del menu")
    )
    if menu_op < 0 or menu_op > 5:
      print("Opcion invalida. Intenta de nuevo.")
    else:
      break


if menu_op == 1:
    print("1.- Los más populares")
    print("2.- Los menos populares")

    while True:
        prod_op = int(input("Selecciona la información
        que deseas ver"))
        if prod_op == 1:
            print("****PRODUCTOS MÁS POPULARES*****")
```

El menú principal imprime, entonces un mensaje de bienvenida con el nombre del usuario y muestra las opciones del menú para ver los datos de los (1) productos, sobre las (2) finanzas de la compañía. Incluso da la opción de poder cambiar (3) el usuario o contraseña de la cuenta y la opción para (4) salir, después de lo cual se le pide al usuario ingresar la opción que desea ver, si el usuario escribe algún número fuera del rango permitido, le seguirá preguntando hasta que escoja una acción válida del menú.

Sub Menú – Productos

Si el usuario selecciona la opción de Productos (tecleando el número 1), podrá ver el menú de arreglo de los productos, la primera opción que el usuario tiene es según los más o menos populares. Después de haber seleccionado el orden o importancia de los productos de ese menú, el usuario escoge la información específica que quiere ver de éstos, en ambos casos son "*información por venta*", *"información por búsqueda*" o "*información por reseña*".

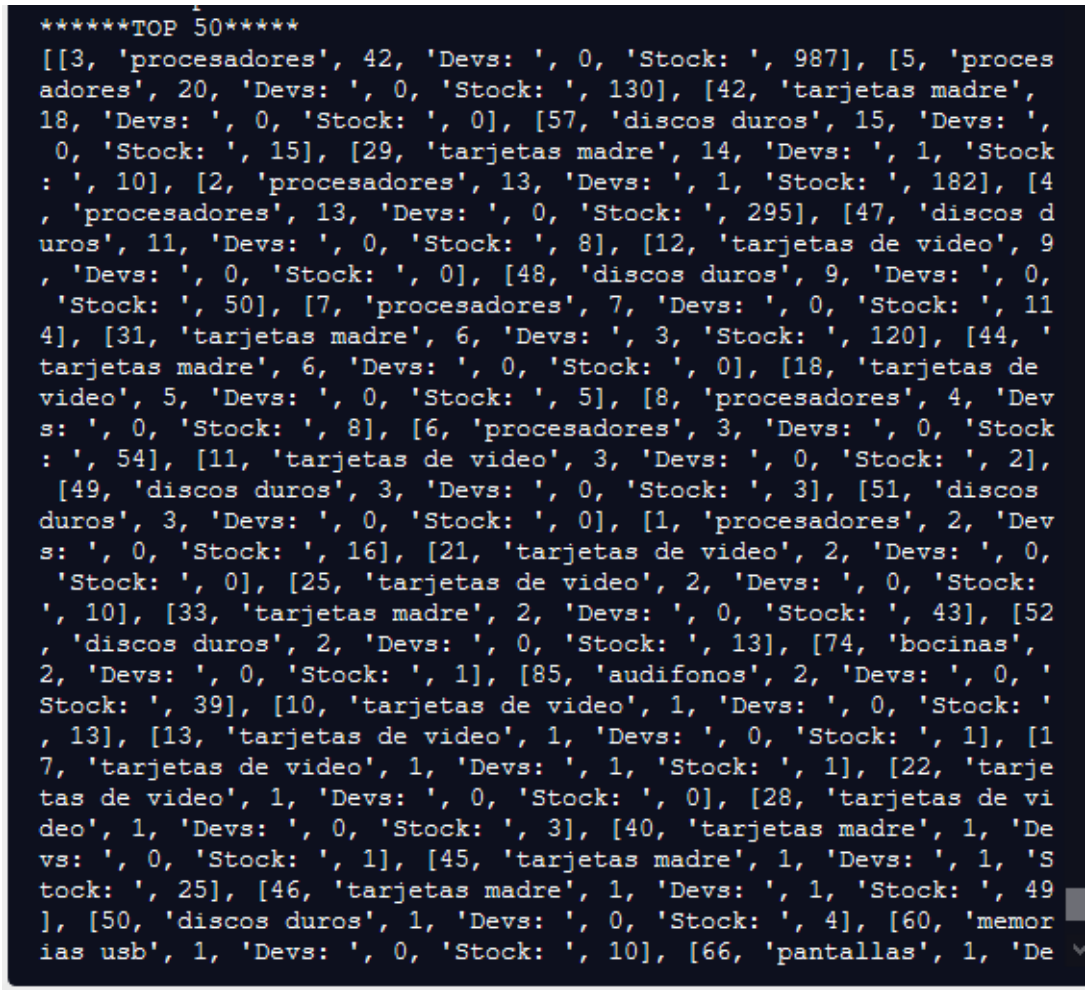


En cualquiera de las informaciones: ventas, búsquedas y reseñas, tanto en los más y los menos vendidos el usuario puede acceder a la información según el ordena que quiera ver. De primera instancia el usuario tiene acceso al TOP 10 de la categoría en cuestión, ya sea los 10 más vendidos, los 10 menos vendidos, las 10 mejores reseñas, etc. Si el usuario quisiera ver más información puede solicitar el TOP 50 o ver la lista entera, según lo desee.

Esto se logró teniendo la lista general en el orden deseado (ascendente o descendente) y posteriormente, mostrar los primeros *x* elementos de esta lista. El código, el cual se repite en las categorías antes mencionadas se muestra a continuación.

```
88                  total_searches = []
89                  for item in lifestore_products:
90                    for search in lifestore_searches:
91                      if item[0] == search[1]:
92                        contador += 1
93                    freq = [item[0], item[-2],
                      contador]
94                    total_searches.append(freq)
95                    contador = 0
96                    total_searches.sort(key = lambda
                      x: x[2], reverse = True)
97                    for item in freq:
98                      if contador == 0:
99                        total_searches.remove(freq)
100
101                 print("******TOP 10*****")
102                 print(total_searches[1:11])
103                 while True:
104                   ver_mas = input("Presiona '+'
                      para ver el top 50 de ventas o
                      presiona 1 para ver la lista
                      completa")
105                   if ver_mas == '+':
106                     print("******TOP 50*****")
107                     print(total_searches[1:51])
108                   if ver_mas == 1:
109                     print("****TODAS LAS
                        BÚSQUEDAS****")
110                     print(total_searches)
```

```
**********************************************************
Selecciona una opción del menu1
1.- Los más populares
2.- Los menos populares
Selecciona la información que deseas ver1
****PRODUCTOS MÁS POPULARES*****
1.- Por venta
2.- Por búsqueda
3.- Por reseña
Selecciona la información que deseas ver1
*****TOP 10 MÁS VENDIDOS****
[[3, 'procesadores', 42, 'Devs: ', 0, 'Stock: ', 987], [5, 'proces
adores', 20, 'Devs: ', 0, 'Stock: ', 130], [42, 'tarjetas madre',
18, 'Devs: ', 0, 'Stock: ', 0], [57, 'discos duros', 15, 'Devs: ',
 0, 'Stock: ', 15], [29, 'tarjetas madre', 14, 'Devs: ', 1, 'Stock
: ', 10], [2, 'procesadores', 13, 'Devs: ', 1, 'Stock: ', 182], [4
, 'procesadores', 13, 'Devs: ', 0, 'Stock: ', 295], [47, 'discos d
uros', 11, 'Devs: ', 0, 'Stock: ', 8], [12, 'tarjetas de video', 9
, 'Devs: ', 0, 'Stock: ', 0], [48, 'discos duros', 9, 'Devs: ', 0,
 'Stock: ', 50]]
Presiona '+' para ver el top 50 de ventas o presiona 1 para ver la
 lista completa
```



Información por categoría - Productos

En el caso de la categoría de ventas lo que se hizo fue establecer dos contadores en 0 y abrir una lista en blanco, la cual será la que se imprimirá al final. Acto seguido se hace una iteración sobre cada uno de los elementos en la lista de productos, por cada uno de ellos se hace una iteración por las ventas en la matriz de ventas y cada vez que coincida el ID del producto con una venta se suma 1 al contador de ventas. De la misma manera, si durante las iteraciones hay un uno en la última posición se van al segundo contador en donde se lleva el conteo de las devoluciones del producto. Se indicó al programa acomodar la lista de la siguiente manera, que será útil en su lectura:

ID, CATEGORÍA, # DE VENTAS, # DE DEVOLUCIONES, # DE PIEZAS EN STOCK

Al final se marcan los contadores en 0 de nuevo y con base en el índice 2 de la sub-lista se ordenan, ya sea ascendente o descendente según sea el caso. Las categorías de análisis este caso, se escogieron pensado en ver qué tanto se vende un producto, si se devuelve por parte del cliente y el número de piezas en stock, ya que si fuera de los más vendidos, tendría que haber para poder surtir, o en su defecto ver algún rezago que haya en el producto.

main.py

```
52
53              if op_prod == 1:
54                contador = 0
55                devs = 0
56                ventastotales = []
57                least_sales = []
58                for item in lifestore_products:
59                  for venta in lifestore_sales:
60                    if item[0] == venta[1]:
61                      contador += 1
62                      if venta[4] == 1:
63                        devs += 1
64                  freq = [item[0], item[-2], contador, "Devs: ", devs, "Stock: ", item[-1]]
65                  ventastotales.append(freq)
66                  contador = 0
67                  devs = 0
68                  least = []
69                  ventastotales.sort(key = lambda x: x[2], reverse = True)
70
71
```

La misma lógica mostrada arriba se usó para las categorías de búsquedas y de reseñas. Los únicos cambios significativos varían en la información que se imprime con cada una de las listas. En el caso de la categoría ordenada por la información de las búsquedas de los productos, la información que se imprime en la lista es la siguiente:

ID PRODUCTO, CATEGORÍA, # BÚSQUEDAS

En este caso el número de búsquedas fue manejado de manera más independiente a los otros casos, sólo se hizo después un análisis comparativo entre los primeros lugares en ventas y en búsquedas para encontrar si existe alguna relación entre las búsquedas que se hace de un producto y su posterior compra o si son eventos independientes. Así como alguna tendencia en categoría de productos del catálogo.

La lista de la información de las reseñas, redondeada a un digito decimal después de haber sido contadas y divididas entre el número de reseñas del producto, muestra la siguiente información:

ID PRODUCTO, CATEGORÍA, # ESTRELLAS, # DE DEVOLUCIONES

Para las reseñas las categorías de análisis fueron pensadas para poder ser evaluadas como productos independientes o por categorías de productos, ya que podría reflejar si el problema –en el caso de los productos con malas reseñas- o los aciertos son algo más general o si es únicamente un producto. Así mismo nos permite ver el número de piezas devueltas del producto, lo que al contrastarlo con la evaluación de la reseña nos permite también tratar de identificar en primera instancia cuál pudo haber sido el problema o si hay algún problema con ciertas piezas específicas o algún problema con el proveedor.

Sub – Menú – Finanzas

La segunda opción del menú principal permite tener acceso a la información financiera de las operaciones. En el menú de Finanzas las primeras opciones que aparecen son la de ingresos mensuales, ingreso anual y devoluciones.

La primera de éstas permite ver las ventas mensuales, cuántos productos se vendieron en cada mes del año. Además de la vista general, el usuario tiene la opción de arreglar la información en orden ascendente, descendente y una tercera opción que le permite ver con detalle cuánto dinero se vendió en cada mes.

```
245             if menu_ventas == 1:
246               meses = ['01', '02', '03', '04', '05', '06',
                  '07', '08', '09', '10', '11', '12']
247               final = []
248               contador = 0
249               for mes in meses:
250                 for sale in lifestore_sales:
251                   if mes == sale[-2][3:5]:
252                     contador += 1
253                 freq = [mes, contador]
254                 final.append(freq)
255                 contador = 0
256               print("****VENTAS MENSUALES***")
257               print(final)
258               print("Para los meses con menores ventas: 1")
259               print("Para los meses con mayores ventas: 2")
260               print("Para ingreso por mes: 3")
261               menu_arreglo = int(input("Selecciona una opción
                  del menu"))
262               if menu_arreglo == 1:
263                 final.sort(key = lambda x:x[1])
264                 print(final[0:4])
265                 opcionmas = input("Presiona 0 para ver la
                    lista completa")
266                 if opcionmas == 0:
267                   print(final)
268               if menu_arreglo == 2:
269                 final.sort(key = lambda x:x[1], reverse = True)
270                 print(final[0:4])
```

Sobre todo durante la sección de finanzas se trabajó con una lista que tiene los números de los meses y un contador en cero para poder hacer las operaciones necesarias, ya sea por unidades vendidas o por capital. En todos los casos después de la impresión general, el usuario tiene la opción de ordenar los resultados según una operación lambda asignada al índice pertinente en cada uno de los casos, ya sea unidades o dinero.

Para la opción 1 del menú de los ingresos mensuales se usaron tres funciones *for* y dos condicionales *if* para poder indicar la suma total vendida en cada uno de los meses. Al igual que se hizo dentro del sub-menú de productos los resultados se iban mandando a una lista en blanco en la cual se iban ordenando los datos para su impresión final. Se imprime el mes y las unidades vendidas durante el mismo y puede ser ordenado por el usuario. Dentro de los detalles (3) el usuario puede ver también a cuánto dinero equivalen esas unidades vendidas y después lo puede ordenar.

```
            print(final)
        if menu_arreglo == 3:
            meses = ['01', '02', '03', '04', '05', '06',
            '07', '08', '09', '10', '11', '12']
            final = []
            contador = 0
            valor = 0
            for mes in meses:
                for sale in lifestore_sales:
                    if mes == sale[-2][3:5]:
                        contador += 1
                        for producto in lifestore_products:
                            if producto[0] == sale[1]:
                                valor += producto[2]
                freq = [mes, contador, '$',valor]
                final.append(freq)
                contador = 0
                valor = 0
            print("****VENTAS POR MES***")
            print(final)
            opcionmas = int(input("Presiona 0 para ordenar
            la lista"))
            if opcionmas == 0:
                final.sort(key = lambda x: x[-1], reverse =
                True)
                print(final)

```

En el apartado para la información anual, se siguió la misma lógica de contadores y mandar la información a una lista en blanco a través de *for* anidado y condicionales. En el apartado anual, además de la suma de todos los valores mensuales, se realizó la operación para el promedio de venta mensual y el promedio de venta diario con base en el resultado del anual. El número fue dividido según el caso. En la sección de detalles se muestran cada de una de las 96 piezas, cuántas unidades vendió y a cuánto dinero equivale, la suma de estos es igual al ingreso anual.

Para la tercera opción dentro del sub-menú de finanzas se agregó la información correspondiente a las devoluciones del producto por parte de los clientes. Se agregó en esta categoría por el impacto que tiene en el ámbito financiero. Esta sección, construida con la misma lógica que las anteriores refleja el impacto económico de las devoluciones, además, por supuesto, del número y tipo de piezas devueltas si se indica al sistema “ver más detalles”, en donde vemos la pieza devuelta, costo unitario, cuántas veces se devolvió y la multiplicación de estos dos últimos factores.

```
if menu_ventas == 3:
    contador = 0
    refunds = 0
    total_refs = []
    totaltotalrefs = []
    for producto in lifestore_products:
        for refund in lifestore_sales:
            if producto[0] == refund[1]:
                contador += 1
                if refund[-1] == 1:
                    refunds += 1
        freq = [producto[0], producto[2], contador,
        refunds]
        total_refs.append(freq)
        freq1 = [producto[0], producto[2], refunds,
        producto[2]*refunds]
        totaltotalrefs.append(freq1)
        contador = 0
        refunds = 0
        total_refs.sort(key = lambda x: x[0], reverse
        = True)
        if freq1[-2] == 0:
            totaltotalrefs.remove(freq1)
    n1 = 0
    for suma in totaltotalrefs:
        n1 += suma[-1]

    print("Total en devoluciones: $",n1)
```

Sub-Menú – Editar Información

Esta sección, correspondiente a la opción 3 del menú principal fue pensada para que el usuario, en casa de que el archivo se pudiera guardar, leer y escribir, pueda editar su información de su cuenta: su nombre, usuario y/o contraseña. En este sub-menú el usuario puede ver la información que tiene actualmente y elige cuál editar.

Para hacer el proceso se sigue un poco la misma lógica que en los pasos de registro. El usuario debe confirmar su información actual, en cualquiera de las tres circunstancias y luego escribe la que desea tener de ese momento en adelante. De tener un sistema que lea y escriba sobre el archivo, el usuario entraría con ese nuevo usuario al sistema. La nueva información sustituye a las variables que se habían definido al inició de la sesión del programa. Si la confirmación de la actualización es llevada a cabo de manera correcta el sistema imprime un mensaje en pantalla notificando que la acción fue realizada con éxito.

```
356  elif menu_op == 3:
357      print("Usuario: ", usuario)
358      print(" 1.- Editar")
359      print("Contraseña: ", contrasena)
360      print("2. Editar")
361      print("Nombre: ", nombre)
362      print("3. Editar")
363
364      while True:
365        nva_info = int(input("Selecciona una opción del
         menú"))
366        if nva_info == 1:
367          old_user = input("Escribe tu actual usuario")
368          nvo_user = input("Escribe tu nuevo usuario")
369          if old_user == usuario:
370            nvo_user = usuario
371          print("Cambio realizado con éxito")
372          break
373        if nva_info == 2:
374          old_pass = input("Escribe tu contraesña actual")
375          new_pass = input("Escribe tu nueva contraseña")
376          if old_pass == contrasena:
377            new_pass = contrasena
378            print("Cambio realizado con éxito")
379            break
380        if nva_info == 3:
381          old_name = input("Escribe tu nombre actual")
382          nvo_name = input("Escribe tu nuevo nombre")
```

Por último, en el menú principal además de las tres ya detalladas anteriormente, existe la opción de salir del programa, en caso de escoger ésta, el sistema imprime un mensaje de despedida antes de terminar.

Solución al problema – Análisis

A continuación, se detalla información relevante encontrada dentro del análisis de la misma.

Sobre las ventas

De manera general, abril fue el mes que más vendió, tanto por unidades como por dinero. Después de eso existe una correlación positiva entre el número de piezas vendidas y los ingresos mensuales, siendo únicamente marzo superior a enero y febrero. De hecho, tan sólo las ventas del mes de abril ($193.295) equivalen a una tercera parte del total vendida en el año ($760.177) hasta el día del corte.

El promedio de ventas mensuales ($63.348) es superior a la mediana. El promedio de ventas diaria ($2.083) resulta parcialmente sesgado debido al alto porcentaje en las ventas de abril y por los meses que faltan en el año. A pesar de ello, meses como agosto, septiembre registran ventas mensuales prácticamente de entre 1.5 – 2 veces el promedio diario registrado de lo que va del año.

Entre el mes de julio y el mes de agosto las ventas caen casi 7 veces, después de haber tenido un desempeño estable y constante, a excepción del periodo entre mayo y junio en donde las ventas caen también alrededor del 250%.

Estas variaciones entre meses pueden explicarse por cuestiones ambientales, de mercado, competencia o por algún cambio interno que haya tenido lugar dentro de la empresa que haya afectado drásticamente el comportamiento del consumidor.

Sobre los productos

De los 10 productos más vendidos, 4 de ellos son procesadores, 2 tarjetas madres, 3 discos duros y una tarjeta de video. Los procesadores son también los artículos más buscados por los clientes, junto con los discos duros.

Existe una correlación entre los artículos más buscados y los artículos más vendidos, ya que 7 de los 10 principales artículos con mayor número de búsquedas son de los más vendidos. Esto sugiere que, en estos artículos, de las categorías antes descritas, más de la mitad de los clientes compran el artículo después de la búsqueda del mismo. Para estos fines, el artículo con más ventas es el Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth.

Más vendidos

1. (3) Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth
2. (5) Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)
3. (42) Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
4. (57) SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm
5. (29) Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
6. (2) Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth
7. (4) Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire
8. (47) SSD XPG SX8200 Pro, 256GB, PCI Express, M.2
9. (12) Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0
10. (48) SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2

Más buscados

1. (57) SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm
2. (29) Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD

3. (3) Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
4. (4) Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire
5. (85) Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul
6. (67) TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro
7. (7) Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)
8. (5) Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)
9. (47) SSD XPG SX8200 Pro, 256GB, PCI Express, M.2
10. (48) SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2

En el otro extremo, los artículos que menos se venden son las tarjetas de video, ya que en lo que va del año no se ha vendido ninguna. A excepción de la Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0, la cual probablemente sea la única que valga la pena mantener en el inventario, ya que incluso este producto se encuentra actualmente agotado. En lugar de comprar las que se encuentran entre las menos vendida de todo el catálogo de la tienda, valdría más la pena invertir en surtir la anteriormente mencionada.

Los productos mejores evaluados en el sistema son los procesadores y las tarjetas de video. La última parte de esta premisa podríamos atribuirla a un sesgo cuantitativo. Al tener pocas ventas, con esa venta bien evaluada es suficiente para poner al producto entre los mejores, mientras que otros más vendidos con más interacciones con el usuario, su desempeño varía más. No obstante, existen otros que, a pesar de un gran número de ventas, mantienen excelentes reseñas por parte de los clientes, varios de ellos incluso con un 5 limpio.

Los productos peores evaluados por los clientes son principalmente tarjetas madre y audífonos. Ninguno de estos productos no figura realmente dentro de los más vendidos, se recomienda hacer un análisis a profundidad sobre esta situación, ya que es inventario detenido que no parece que vaya moverse pronto, además de que las malas reseñas ahuyentan a los clientes en un ciclo que necesitará de una estrategia contundente para deshacerse del material detenido.

Sobre las devoluciones

En lo que va del año, las devoluciones hechas por los clientes han sido el equivalente a $22.261, lo que representa alrededor de un 3% de las ventas totales del año. Considerando algunos niveles de ventas bajos en los últimos meses, este número podría ser significativamente de no ser atendido.

El artículo que más veces ha sido devuelto es la tarjeta madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD, se recomienda realizar una supervisión de calidad para descartar algún problema con el lote, ya que las tres ocurrieron en fechas cercanas.

El mes de septiembre prácticamente podría considerarse en ceros, ya que la única venta realizada hasta la fecha ese mes fue una devolución de una tarjeta de video. Dejando de lado ello, el resto de las 8 devoluciones fueron hechas entre el mes de enero y el mes de mayo, durante los meses con mayor volumen de ventas.

Podríamos inferir que las devoluciones tuvieron que ver con cuestiones del proveedor, más que con la tienda, pues a pesar de las pocas devoluciones que hubo las ventas se mantenían altas por el servicio de la tienda. A partir de mayo inicia el declive en ventas, esto no es debido a la calidad de los productos ni del proveedor, pues las devoluciones ni siquiera suman un 2% de las ventas totales hechas por la tienda. Se recomienda reconsiderar los cambios administrativos que se hayan hecho al interior de la organización.

Conclusión

Sobre el problema

Derivado del análisis realizado en la sección anterior se han emitido algunas recomendaciones hechas según el estudio preliminar hecha con la información proporcionada por parte de la tienda. De manera general y lo de más urgencia resulta ser la revisión de cambios hechos al interior de la organización de la tienda o algún disruptor mayor en el ambiente del mercado que haya alterado las ventas en la manera en que lo hizo para Life Store. Una función de regresión lineal ayudaría a predecir la tendencia para el cierre de año en las ventas de la tienda.

La tienda inició con buenas ventas a pesar de las circunstancias del encierro por coronavirus, sin embargo, ahora que se ha vuelto a la normalidad sus ventas no lo han hecho. Hay productos que son repetitivos en estar mal evaluados y mal vendidos. Se recomienda deshacerse del stock de estos productos, a costa de reducir el catálogo y tener más stock de los más vendidos, ya que hay varios que tienen un stock muy pequeño o incluso agotado.

Sobre el proyecto

Resultó ser la primera vez que utilizo ciencia de datos para resolución de problemas en escenarios de la vida real. Fue todo un reto y aunque enfrenté muchas dificultas durante varias etapas del proyecto, estoy satisfecho con el resultado y ojalá la estructura llegue a resultar útil para alguien.

Es importante mencionar que para un buen análisis es necesario recolectar la mayor cantidad de información posible. Por ejemplo, en el caso de este trabajo la información pudo haber sido ampliada habiendo evaluado las ventas por días de la semana para ver comportamientos en tendencias ahí, creando correlaciones entre valores, etc, sin mencionar la posibilidad de guardar, leer y escribir sobre el propio archivo para crear una lista actualizada de los usuarios y su información.

Cualquier empresa al día de hoy necesita estar midiendo su desempeño día a día. Esto requiere hacerse las preguntas correctas para saber qué evaluar, cómo

medirlo, pero además es necesario tener la tecnología y el conocimiento necesario para poder interpretar la información y aportar datos de valor que ayuden a una mejora.

Anexo

Copia del código y liga en GitHub:

```
from StoreData import lifestore_products, lifestore_sales,
lifestore_searches

print("\n   ***Bienvenido a Life Store***\n")
print("     --Para las necesidades de tu vida--\n")

print("         Para continuar, inicia sesión\n")

nombre = input("          Nombre: ")
usuario = input("         Usuario: ")
contrasena = input("      Contraseña: ")

print("Para verificar tu información, por favor ingresa tus datos
nuevamente")

while True:
    verif_usuario = input("          Usuario: ")
    verif_contrasena = input("          Contrseña: ")
    if verif_usuario == usuario and verif_contrasena == contrasena:
        print("  BIENVENIDO, ", nombre,"!")
        break
    else:
        print("El usuario o contraseña ingresados no coinciden. Favor de
intentarlo de nuevo")

print("**************************************************")
print("1.- Productos")
print("2. Finanzas")
print("3. Editar información personal")
print("4. Salir")
print("**************************************************")

sales = []
while True:
    menu_op = int(input("Selecciona una opción del menu"))
    if menu_op < 0 or menu_op > 5:
```

```
      print("Opcion invalida. Intenta de nuevo.")
    else:
      break


if menu_op == 1:
    print("1.- Los más populares")
    print("2.- Los menos populares")

    while True:
        prod_op = int(input("Selecciona la información que deseas ver"))
        if prod_op == 1:
            print("****PRODUCTOS MÁS POPULARES*****")
            print("1.- Por venta")
            print("2.- Por búsqueda")
            print("3.- Por reseña")
            while True:
                op_prod = int(input("Selecciona la información que deseas
ver"))

                if op_prod == 1:
                  contador = 0
                  devs = 0
                  ventastotales = []
                  least_sales = []
                  for item in lifestore_products:
                    for venta in lifestore_sales:
                      if item[0] == venta[1]:
                        contador += 1
                        if venta[4] == 1:
                          devs += 1
                    freq = [item[0], item[-2], contador, "Devs: ", devs,
"Stock: ", item[-1]]
                    ventastotales.append(freq)
                    contador = 0
                    devs = 0
                    least = []
                    ventastotales.sort(key = lambda x: x[2], reverse = True)



                  print("*****TOP 10 MÁS VENDIDOS****")
                  print(ventastotales[1:11])
                  while True:
```

```
                    ver_mas = input("Presiona '+' para ver el top 50 de
ventas o presiona 1 para ver la lista completa")
                    if ver_mas == '+':
                      print("******TOP 50*****")
                      print(ventastotales[1:51])
                    if ver_mas == 1:
                      print("****TODAS LAS VENTAS****")
                      print(ventastotales)



                elif op_prod == 2:
                  contador = 0
                  total_searches = []
                  for item in lifestore_products:
                    for search in lifestore_searches:
                      if item[0] == search[1]:
                        contador += 1
                    freq = [item[0], item[-2], contador]
                    total_searches.append(freq)
                    contador = 0
                    total_searches.sort(key = lambda x: x[2], reverse =
True)
                  print("******TOP 10*****")
                  print(total_searches[1:11])
                  while True:
                    ver_mas = input("Presiona '+' para ver el top 50 de
ventas o presiona 1 para ver la lista completa")
                    if ver_mas == '+':
                      print("******TOP 50*****")
                      print(total_searches[1:51])
                    if ver_mas == 1:
                      print("****TODAS LAS BÚSQUEDAS****")
                      print(total_searches)

                elif op_prod == 3:
                  contador = 0
                  stars = 0
                  devs = 0
                  total_reviews = []
                  for producto in lifestore_products:
                    for review in lifestore_sales:
                      if producto[0] == review[1]:
                        contador += 1
                        stars += review[2]
```

```
                        final_s = stars/contador
                        final_stars = round(final_s, 2)
                        if review[4] == 1:
                          devs += 1
                      freq = [producto[0], producto[-2],final_stars,
"Devoluciones: ", devs]
                      total_reviews.append(freq)
                      contador = 0
                      stars = 0
                      devs = 0
                      total_reviews.sort(key = lambda x: x[2], reverse = True)
                    print("*****TOP 10 MEJORES ARTÍCULOS****")
                    print(total_reviews[1:11])
                    while True:
                      ver_mas = input("Presiona '+' para ver el top 50 de
ventas o presiona 1 para ver la lista completa")
                      if ver_mas == '+':
                        print("******TOP 50*****")
                        print(total_reviews[1:51])
                      if ver_mas == 1:
                        print("****TODAS LAS VENTAS****")
                        print(total_reviews)


                  else:
                    print("Opción inválida. Intenta de nuevo")

          elif prod_op == 2:
              print("****LOS PRODUCTOS MENOS POPULARES*****")
              print("1.- Por venta")
              print("2.- Por búsqueda")
              print("3.- Por reseña")
              while True:
                  op_prod = int(input("Selecciona la información que deseas
ver"))

                  if op_prod == 1:
                    contador = 0
                    devs = 0
                    ventastotales = []
                    for item in lifestore_products:
                      for venta in lifestore_sales:
                        if item[0] == venta[1]:
                          contador += 1
                          if venta[4] == 1:
```

```
                        devs += 1
                    freq = [item[0], item[-2],contador, "Devoluciones: ",
devs, "Stock: ", item[-1]]
                    ventastotales.append(freq)
                    contador = 0
                    devs = 0
                    least = []
                    ventastotales.sort(key = lambda x: x[2])

                  print("****LAS MANZANAS PODRIDAS****")
                  print("**** TOP 10 MENOS VENTAS :( ****")
                  print(ventastotales[1:11])
                  while True:
                    ver_mas = input("Presiona '+' para ver el top 50 de
ventas o presiona 1 para ver la lista completa")
                    if ver_mas == '+':
                      print("******TOP 50 :( *****")
                      print(ventastotales[1:51])
                    if ver_mas == 1:
                      print("****TODAS LAS NO-VENTAS****")
                      print(ventastotales)

                elif op_prod == 2:
                  contador = 0
                  total_searches = []
                  for item in lifestore_products:
                    for search in lifestore_searches:
                      if item[0] == search[1]:
                        contador += 1
                    freq = [item[0], item[-2], contador]
                    total_searches.append(freq)
                    contador = 0
                    total_searches.sort(key = lambda x: x[2])

                  print("***TOP 10 ABANDONADOS :(***")
                  print(total_searches[1:11])
                  while True:
                    ver_mas = input("Presiona '+' para ver el top 50 de
ventas o presiona 1 para ver la lista completa")
                    if ver_mas == '+':
                      print("******TOP 50 MÁS ABANDONADOS*****")
                      print(total_searches[1:51])
                    if ver_mas == 1:
                      print("****TODOS LOS RELEGADOS****")
                      print(total_searches)
```

```
                elif op_prod == 3:
                  contador = 0
                  stars = 0
                  devs = 0
                  total_reviews = []
                  for producto in lifestore_products:
                    for review in lifestore_sales:
                      if producto[0] == review[1]:
                        contador += 1
                        stars += review[2]
                        final_s = stars/contador
                        final_stars = round(final_s, 2)
                        if review[4] == 1:
                          devs += 1
                    freq = [producto[0], producto[-2],final_stars,
"Devoluciones: ", devs]
                    total_reviews.append(freq)
                    contador = 0
                    stars = 0
                    devs = 0
                    total_reviews.sort(key = lambda x: x[2])
                  print("****LOS 10 MENOS QUERIDOS****")
                  print(total_reviews[1:11])
                  while True:
                    ver_mas = input("Presiona '+' para ver el top 50 de
ventas o presiona 1 para ver la lista completa")
                    if ver_mas == '+':
                      print("******TOP 50 PEOR CALIFICADOS*****")
                      print(total_reviews[1:51])
                    if ver_mas == 1:
                      print("****TODAS LAS VENTAS****")
                      print(total_reviews)


        else:
            print("Opción invalida. Intenta de nuevo")

elif menu_op == 2:
    print("****FINANZAS****")
    print("1.- Ventas Mensual")
    print("2.- Ingreso Anual")
    print("3.- Devoluciones")
    while True:
      menu_ventas = int(input("Selecciona la opción que deseas ver"))
```

```
      if menu_ventas == 1:
        meses = ['01', '02', '03', '04', '05', '06', '07', '08', '09', '10',
'11', '12']
        final = []
        contador = 0
        for mes in meses:
          for sale in lifestore_sales:
            if mes == sale[-2][3:5]:
              contador += 1
          freq = [mes, contador]
          final.append(freq)
          contador = 0
        print("****VENTAS MENSUALES***")
        print(final)
        print("Para los meses con menores ventas: 1")
        print("Para los meses con mayores ventas: 2")
        print("Para ingreso por mes: 3")
        menu_arreglo = int(input("Selecciona una opción del menu"))
        if menu_arreglo == 1:
          final.sort(key = lambda x:x[1])
          print(final[0:4])
          opcionmas = input("Presiona 0 para ver la lista completa")
          if opcionmas == 0:
            print(final)
        if menu_arreglo == 2:
          final.sort(key = lambda x:x[1], reverse = True)
          print(final[0:4])
          opcionmas = input("Presiona 0 para ver toda la lista")
          if opcionmas == 0:
            print(final)
        if menu_arreglo == 3:
          meses = ['01', '02', '03', '04', '05', '06', '07', '08', '09',
'10', '11', '12']
          final = []
          contador = 0
          valor = 0
          for mes in meses:
            for sale in lifestore_sales:
               if mes == sale[-2][3:5]:
                contador += 1
                for producto in lifestore_products:
                  if producto[0] == sale[1]:
                    valor += producto[2]
            freq = [mes, contador, '$',valor]
            final.append(freq)
```

```
      contador = 0
      valor = 0
    print("****VENTAS POR MES***")
    print(final)
    opcionmas = int(input("Presiona 0 para ordenar la lista"))
    if opcionmas == 0:
      final.sort(key = lambda x: x[-1], reverse = True)
      print(final)

if menu_ventas == 2:
  contador = 0
  refunds = 0
  total_refs = []
  totaltotalrefs = []
  ingresototalanual = []
  for producto in lifestore_products:
    for refund in lifestore_sales:
      if producto[0] == refund[1]:
        contador += 1
        if refund[-1] == 1:
          refunds += 1
    freq = [producto[0], producto[2], contador, refunds]
    total_refs.append(freq)
    freq1 = [producto[0], producto[2], contador, producto[2]*contador]
    totaltotalrefs.append(freq1)
    freq2 = [producto[2]*contador]
    ingresototalanual.append(freq2)
    contador = 0
    refunds = 0
    total_refs.sort(key = lambda x: x[0], reverse = True)
  sumatotal = sum(sum(x) for x in ingresototalanual)
  print("INGRESO ANUAL: $",sumatotal)
  print("PROMEDIO MENSUAL: $", (round(sumatotal/12)))
  print("PROMEDIO DIARIO: $", (round(sumatotal/365)))
  opcionmas = input("Para ver detalles presiona '+'")
  if opcionmas == "+":
    print(totaltotalrefs)

if menu_ventas == 3:
  contador = 0
  refunds = 0
  total_refs = []
  totaltotalrefs = []
  for producto in lifestore_products:
    for refund in lifestore_sales:
```

```
            if producto[0] == refund[1]:
              contador += 1
              if refund[-1] == 1:
                refunds += 1
          freq = [producto[0], producto[2], contador, refunds]
          total_refs.append(freq)
          freq1 = [producto[0], producto[2], refunds, producto[2]*refunds]
          totaltotalrefs.append(freq1)
          contador = 0
          refunds = 0
          total_refs.sort(key = lambda x: x[0], reverse = True)
          if freq1[-2] == 0:
            totaltotalrefs.remove(freq1)
        n1 = 0
        for suma in totaltotalrefs:
          n1 += suma[-1]

        print("Total en devoluciones: $",n1)
        opcionmas = input("Para más detalles presiona +")
        if opcionmas == "+":
          print(totaltotalrefs)


elif menu_op == 3:
    print("Usuario: ", usuario)
    print(" 1.- Editar")
    print("Contraseña: ", contrasena)
    print("2. Editar")
    print("Nombre: ", nombre)
    print("3. Editar")

    while True:
      nva_info = int(input("Selecciona una opción del menú"))
      if nva_info == 1:
        old_user = input("Escribe tu actual usuario")
        nvo_user = input("Escribe tu nuevo usuario")
        if old_user == usuario:
          nvo_user = usuario
        print("Cambio realizado con éxito")
        break
      if nva_info == 2:
        old_pass = input("Escribe tu contraesña actual")
        new_pass = input("Escribe tu nueva contraseña")
        if old_pass == contrasena:
          new_pass = contrasena
```

```
            print("Cambio realizado con éxito")
            break
        if nva_info == 3:
          old_name = input("Escribe tu nombre actual")
          nvo_name = input("Escribe tu nuevo nombre")
          if old_name == nombre:
            nvo_name = nombre
            print("Cambio realizado con éxito")
            break
        else:
          print("Opción no válida. Intenta de nuevo")


elif menu_op == 4:
    print("Gracias por elegir Life Store")
    print("Hasta pronto")
    exit
else:
  print("La opción ingresada no existe. Intenta de nuevo")
```