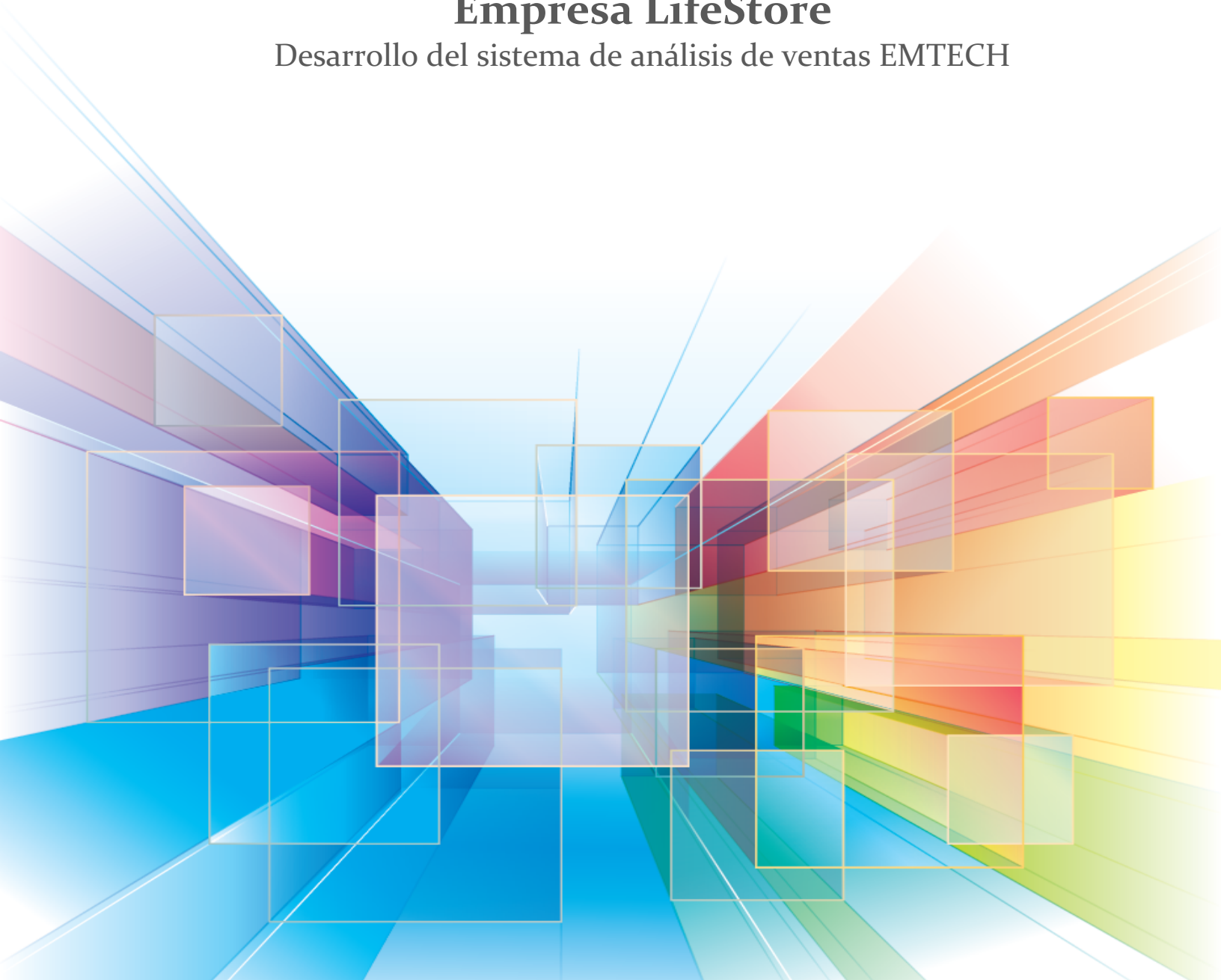


Análisis de Rotación de Productos

Fundamentos de Programación con Python I

Empresa LifeStore

Desarrollo del sistema de análisis de ventas EMTECH



Índice

Introducción	2
Definición del código	2
Inicio de sesión	2
Menú principal de opciones	3
Análisis de ventas	5
Análisis de búsquedas	9
Análisis de reseñas	10
Análisis de ingresos	12
Solución al problema	15
Productos más vendidos	15
Productos con mejores reseñas	16
Productos rezagados	17
Estrategia sugerida	18
Conclusiones	22

Enlace al proyecto en github:

https://github.com/ErickNavarroD/EMTECH_DataScience

Introducción

El análisis de datos en empresas que manejan una variedad amplia de productos es un tema crucial que debe ser examinado con el objetivo de evaluar los productos que se ofrecen al público, su demanda, cómo es que los consumidores lo valoran y las veces que solicitan una devolución. En caso de no hacerlo, la empresa se arriesga a almacenar productos que no son del agrado de los consumidores, generando una tienda no optimizada en la cual hay una ausencia de requerimientos personalizados para cada producto.

Al analizar los datos de ventas y búsquedas de cada producto, es posible generar estrategias de mercado que permitan maximizar las ventas de la empresa y reducir los costos asociados al almacén de los objetos a vender. Así mismo, con base en los hechos observados, se pueden tomar decisiones inteligentes con respecto al cese de venta de un producto dado o al aumento del mismo cuando se observa una tendencia fuerte de oferta o búsqueda.

Por otro lado, los datos pueden analizarse también para observar la tendencia que tienen los consumidores de comprar a lo largo de distintos meses del año; explorar el comportamiento de las ventas promedio mensuales ofrece información muy valiosa para establecer los valores ideales de productos que debe haber en diferentes momentos para satisfacer las necesidades de los clientes sin comprometer las ganancias de la empresa.

La empresa LifeStore carece de un sistema de análisis que, acoplado a la base de datos de ventas de la tienda, nos permita saber cuáles son los productos más y menos vendidos, buscados y satisfactorios. De igual manera, no existe forma de saber la venta mensual promedio a lo largo del año de forma automática; hacerlo de forma manual es sumamente agobiante y requiere una gran cantidad de tiempo que podría aprovecharse en otras actividades. Es por eso que, en este proyecto, programamos un sistema de análisis en python que es capaz de ofrecer al usuario toda la información necesaria para tomar decisiones en la empresa y progresar hacia una tienda con menores pérdidas, más ganancias y, lo más importante, que se adapte a la respuesta del cliente al observar el comportamiento de ventas y búsquedas. De esta manera, lograremos tener una empresa dinámica ajustada a las demandas de nuestros consumidores, aumentando las probabilidades de la misma de crecer a largo plazo y convertirse en la primera opción de los clientes para comprar los productos que requieran.

Definición del código

Inicio de sesión

Con el objetivo de generar un sistema que solamente muestra los datos al personal de la empresa que tenga la autorización de explorar el comportamiento de las ventas de la tienda, generamos un sistema capaz de reconocer un usuario y contraseña que, en caso de ser correctas, desplegará un menú de opciones para solicitar información (**Figura 1**).

```

1436 #%%% Login del usuario
1437 #Establecer usuario y contraseña correcta
1438 usuario_correcto = "EMTECH"
1439 contraseña_correcta = "EMTECH"
1440 #Pedir el usuario y contraseña al usuario
1441 usuario = input("Bienvenido al sistema de análisis EMTECH de Erick Navarro. Por favor, ingrese su usuario: ")
1442 contraseña = input("Ingrese su contraseña: ")
1443
1444 #Repetir mientras la contraseña y el usuario sean incorrectos
1445 import time
1446 import sys
1447 while usuario != usuario_correcto or contraseña != contraseña_correcta:
1448     print("")
1449     print("\nLo sentimos, el usuario o la contraseña que ingresó es incorrecto\n")
1450     #Establecer opción de terminar el programa
1451     salida = input("¿Desea salir del programa? (si/no): ")
1452     #Tomar en cuenta cuanto no dicen si o no
1453     while salida != "no" and salida != "si":
1454         salida = input("\nEntrada incorrecta, por favor escriba si o no: ")
1455     if salida == "si":
1456         print("\nMuchas gracias por utilizar el sistema de análisis EMTECH. ¡Hasta luego!")
1457         time.sleep(3)
1458         sys.exit()
1459         break
1460     #Pedir el usuario y contraseña de nuevo
1461     usuario = input("\nIngrese su usuario nuevamente: ")
1462     contraseña = input("Ingrese la contraseña: ")
1463
1464 print("Inicio de sesión exitoso")
1465

```

Figura 1. Inicio de sesión.

El fragmento de código para esto se encuentra en la **Figura 1**. Como se puede observar, para motivos de demostración, el usuario correcto y la contraseña correcta en este caso se establecieron como “EMTECH”. Al iniciar la sesión, se le pedirá al usuario su nombre de usuario y contraseña, los cuales se compararán con los correctos (línea 1447). Mientras esto sea incorrecto, se le dirá al usuario que el nombre o contraseña son incorrectos; por motivos de seguridad no se le especificará cuál de ambos lo fue. Posteriormente, se le ofrecerá la opción de salir del programa (línea 1451). En caso de que no escriba “si” o “no”, se le dirá al usuario que la entrada no es válida, hasta que la respuesta sea una de las dos opciones señaladas (línea 1453). Finalmente, si el usuario elige salir, el sistema se despedirá y el programa terminará. En caso de elegir “no”, se le pedirá al usuario nuevamente la contraseña y el nombre de usuario (línea 1461) hasta que estas sean correctas o la persona decida salir del programa. Finalmente, en caso de ingresar el nombre de usuario y contraseña correcta, se indicará que el inicio de sesión es exitoso y se procederá con las demás partes del programa (línea 1464).

Menú principal de opciones

Posteriormente, el programa procede a desplegar un menú de opciones para que el usuario elija la información que desea ver (**Figura 2**).

```

1466  %% Menú principal de opciones
1467  #Funcion para comprobar que la opcion ingresada se encuentre dentro de las opciones posibles del menu
1468  def comprobar_opcion(opcion, inicio, fin):
1469      while str(opcion) not in [str(i) for i in range(inicio,fin+1)]:
1470          opcion = input("\nEntrada incorrecta, por favor indique uno de los números indicados anteriormente: ")
1471      return(opcion)
1472
1473  #Preguntar al cliente si quiere hacer algo más o salir
1474  def regresar_menu():
1475      regresar_menu = input("""\nDesea realizar alguna otra solicitud?:
1476          1: sí
1477          2: no
1478          """)
1479      #Comprobar las opciones
1480      regresar_menu = comprobar_opcion(regresar_menu,1,2)
1481      if regresar_menu == "1":
1482          menu_principal()
1483      if regresar_menu == "2":
1484          print("""\nMuchas gracias por utilizar el sistema de análisis EMTECH.
1485              ¡Hasta luego!""")
1486          time.sleep(3)
1487          quit()
1488
1489  def menu_principal():
1490      opcion_informacion = input("""\n    BIENVENIDO AL MENÚ PRINCIPAL DEL SISTEMA DE ANÁLISIS EMTECH
1491      Por favor ingrese el número que indique la información que desea obtener:
1492          1: Información de productos vendidos
1493          2: Información de la búsqueda de los productos
1494          3: Información de la reseña de los productos
1495          4: Información de ingresos de la empresa
1496          5: salir del programa
1497          """)
1498      opcion_informacion = comprobar_opcion(opcion_informacion,1,5)

```

Figura 2. Menú principal y funciones básicas usadas frecuentemente.

Lo primero que podemos observar las únicas tres funciones que fueron creadas en el programa con el único objetivo de ahorrar líneas de código, pues son fragmentos que se estarán utilizando constantemente a lo largo del sistema de análisis (Nota del curso: además, fue un fragmento que ya había hecho e implementado cuando mencionaron que evitáramos hacer funciones; como consideré que son muy sencillas y utilizan la misma lógica de lo visto en el curso decidí dejarlas, pues solamente son invocadas para realizar lo mismo).

La primera función, **comprobar_opción**, tiene el fin de comprobar que las opciones ingresadas por el usuario sean válidas. Para facilitar esto, de aquí en adelante, en el programa se dejan de solicitar palabras y se le pide al usuario que ingrese el número correspondiente a lo que desea ver. En esta función se ingresa el input del usuario (opción), el primer número que es válido en nuestro menú de opciones (inicio), y el último de ellos (fin). Posteriormente, se compara si el carácter ingresado por el usuario se encuentra en el rango de caracteres de los números de inicio a fin (línea 1469). Mientras el usuario ingrese cualquier cosa que no se encuentre dentro de ese conjunto de caracteres permitidos, se le pedirá al usuario que ingresa de nuevo su opción (línea 1470). Debido a que se le pide constantemente al usuario ingresar números a lo largo del programa, decidí hacer el código aquí e invocar ese fragmento cada que desee verificar la opción.

La segunda función, **regresar_menu**, le pregunta al usuario si desea hacer alguna otra operación; en caso de que así sea, vuelve a correr el programa principal donde se encuentran todas las opciones (**menu_principal**), en caso contrario, cierra el programa. Esta función se invoca al final de cada una de las opciones terminales del programa para dar la opción de regresar al menú principal.

Finalmente, **menu_principal** es una función que posee dentro todo lo que se va a hacer en el programa y que se invoca una vez nada más al final del script para que de inicio. La única razón por la cual la hice función, además de la nota explicada anteriormente, es para que se pudiera regresar a ella cada vez que quisiera, creando iteraciones en ella misma sin tener que copiar y pegar todo el código de nuevo. Después de dar la bienvenida, se establecen 5 opciones; cuatro de ellas son para obtener información de la tienda y la última es para salir del programa. Las primeras cuatro se explicarán a detalle en el reporte, y la última únicamente termina el script. Tras comprobar que el número que el usuario ingrese sea válido, se procede a hacer el análisis indicado.

Análisis de ventas

En caso de elegir la opción 1 en el menú principal (**Figura 2**), se realizará un análisis del número de ventas de cada producto por categoría. Para esto, el primer paso es generar una lista que posea al frecuencia de venta de cada uno de los productos; esto se realiza de la línea 1508 a 1518, donde por cada venta se evalúa si no está en la lista de frecuencias, agrgándola a esta con un 1 y, en caso de que ya esté, se le suma 1 a la frecuencia del mismo. Posteriormente, se ordena esta lista de mayor a menor frecuencia de venta (líneas 1523-1534). Finalmente, se le añade a la lista el nombre, categoría y lugar de ventas de cada producto. Esto se hace buscando el ID del producto en la lista de **lifestore_products**; en caso de hacer match, se le añade a la lista el nombre y categoría, lo cual se usará posteriormente para desplegar los datos (línea 1535-1544). Así, al final de esto se termina con una lista que posee una lista en cada elemento con el ID del producto, su frecuencia de ventas, nombre, categoría y lugar en venta (líneas 1546-1547, **Figura 3**).

```

1500 #Opcion productos vendidos
1501 if opcion_informacion == "1":
1502     print("\nElegiste información de productos vendidos")
1503     #Hacer lista mas a menos vendidos
1504     conteos_ventas = [] #inicializar lista que va a tener [id_product, numero de ventas]
1505     ids_ventas = [] #Inicializar una lista que me va a servir para saber cuando un producto ya se haya contado o no
1506
1507     #Generar lista unicamente del ID de los productos
1508     for venta in lifestore_sales:
1509         #Cuando no este en la lista
1510         if venta[1] not in ids_ventas:
1511             ids_ventas.append(venta[1])
1512             conteos_ventas.append([venta[1],1])
1513         #Cuando el ID ya esté en la lista de ventas (ya se registro)
1514         elif venta[1] in ids_ventas:
1515             #buscar el ID que le corresponde en la lista de conteos
1516             for id in conteos_ventas:
1517                 if venta[1] == id[0]:
1518                     id[1]+=1
1519     #Comprobar que el total de conteos sea igual a la longitud de la lista de ventas
1520     #x = 0
1521     #for i in conteos_ventas:
1522     #    x += i[1]
1523     #Ordenar la lista por frecuencia de ventas
1524     conteos_ventas_ordenados = []
1525     while conteos_ventas:
1526         maximo = conteos_ventas[0][1]
1527         lista_actual = conteos_ventas[0]
1528         for producto in conteos_ventas:
1529             if producto[1] > maximo:
1530                 maximo = producto[1]
1531                 lista_actual = producto
1532         conteos_ventas_ordenados.append(lista_actual)
1533         conteos_ventas.remove(lista_actual)
1534     del conteos_ventas #Eliminar lista desordenada que quedó vacía
1535     #Añadir a la lista categoría, nombre del producto, y lugar en la lista.
1536     for producto in conteos_ventas_ordenados:
1537         for id in lifestore_products:
1538             if producto[0] == id[0]:
1539                 producto.append(id[1]) #añadir nombre
1540                 producto.append(id[3]) #añadir categoría
1541                 break
1542         continue
1543     for lugar in range(0,len(conteos_ventas_ordenados)):
1544         conteos_ventas_ordenados[lugar].append(lugar+1)
1545
1546     Producto final:
1547     conteos_ventas_ordenados = [id_product, NumeroDeVentas, Nombre, categoría, LugarEnVentas]
1548

```

Figura 3. Generación de frecuencia de ventas por producto en una lista ordenada.

Posteriormente, se necesita separar estas ventas por categorías, en caso de que el usuario desee ver la información en cada una de ellas. Para hacer esto, se genera una lista con las categorías de los productos vendidos (línea 1550-1554). Posteriormente, se genera otra lista donde cada elemento contiene otra lista de las ventas de esa categoría dada en específico (líneas 1555-1563). Esto se logra haciendo un loop for en donde, por cada categoría contenida en la lista previamente hecha (línea 1550-1554), se compara si cada elemento vendido pertenece a esta; en caso de que así sea, se añade a su lista correspondiente de ventas_por_categoria (explicado en líneas 1564-1570). Finalmente, ya que están hechas las listas ordenadas con la información necesaria, se le pregunta al usuario si desea ver los productos más vendidos, menos vendidos o regresar al menú principal (líneas 1572-1577). Tras verificar que el input sea correcto (línea 1579), se procede a hacer lo indicado (Figura 4).


```

1549     ###Generar listas separadas por categorías
1550     #Hacer lista con las categorías únicas vendidas
1551     categorias_unicas_ventas = []
1552     for pro_producto: list_ventas_ordenados:
1553         if producto[3] not in categorias_unicas_ventas:
1554             categorias_unicas_ventas.append(producto[3])
1555     #Hacer una lista de ventas por categoría
1556     ventas_por_categoria = []
1557     for numero_categoria in range(0,len(categorias_unicas_ventas)):
1558         for producto_vendido in conteos_ventas_ordenados:
1559             if producto_vendido[3] == categorias_unicas_ventas[numero_categoria]:#Añadir el producto cuando ya esta la categoria
1560                 if numero_categoria >= len(ventas_por_categoria):
1561                     ventas_por_categoria.append([producto_vendido])
1562                 else:
1563                     ventas_por_categoria[numero_categoria].append(producto_vendido)
1564     """Producto final:
1565     categorias_ventas_unicas contiene el nombre de cada categoría vendida
1566     ventas_por_categoria es una lista que agrupa ventas de una misma categoría, donde cada elemento es una lista de id del producto, nombre,
1567     categoría y lugar en ventas.
1568     Su posición corresponde a cada categoría de categorias_ventas_unicas.
1569     categorias_ventas_unicas = [categoría]
1570     ventas_por_categoria = [lista de productos vendidos dentro de una categoría]
1571     """
1572     #Elegir la información que se necesita
1573     opcion_secundaria = input("""\nElige la información que deseas ver:
1574     1: Lista de productos más vendidos
1575     2: Lista de productos menos vendidos
1576     3: Regresar al menú principal
1577     """)
1578     #Comprobar que la opción sea correcta
1579     opcion_secundaria = comprobar_opcion(opcion_secundaria,1,3)

```

Figura 4. Creación de lista con ventas por categoría ordenadas y pregunta al usuario de información requerida.

En caso de que el usuario elija ver los productos más vendidos, se le pregunta al usuario qué categoría quiere ver, o si quiere saber esta información de todas las categorías. Para esto, en un loop for, cada categoría vendida se imprime en pantalla junto con su posición en la lista previamente hecha (líneas 1583-1585) Después el número siguiente se imprime, indicando que esa opción es la de todas las categorías (línea 1586), y el número siguiente brinda la opción de regresar al menú principal (línea 1587). Tras validar que el número ingresado se encuentre dentro del rango 1: (número de categorías vendidas +2), se procede a hacer lo indicado.

Lo primero que se hace es evaluar si la elección del usuario corresponde a una de las categorías de los productos (línea 1591); en caso de ser así, se imprimen los 50 productos más vendidos en caso de que esa categoría posea más de 50 elementos (líneas 1593-1597). En el caso de que posean menos de 50 elementos, se indicará al usuario que hay menos de 50 productos de esa categoría, y se desplegarán todos los elementos contenidos ordenados de mayor a menor venta (líneas 1598-1602). Al finalizar ambos procedimientos, se pregunta al usuario si quiere realizar otra operación y regresar al menú principal o no (**Figura 5**).

En caso de que el usuario haya elegido ver los productos más vendidos en toda la tienda (línea 1604), se muestran en pantalla los 50 primeros productos de la lista ordenada, en caso de que sean más de 50 (líneas 1606-1611), o todos los elementos en caso de que sean menos de 50, indicando al usuario que eso es lo que pasó (líneas 1613-1619). Al finalizar ambos procesos se pregunta al usuario si quiere regresar al menú principal o salir del programa. Finalmente, en caso de que el usuario haya elegido el número (número de categorías +2), se le regresará al menú principal (líneas 1621-1622, **Figura 5**).


```

1581 if opcion_secundaria == "1":
1582     #Menú de categorías
1583     print("\nElige la opción para la que deseas ver los 50 productos más vendidos:")
1584     for numero_categoria in range(1, len(categorias_unicas_ventas)+1):
1585         print(str(numero_categoria) + ": Categoría " + categorias_unicas_ventas[numero_categoria-1] )
1586     print(str(len(categorias_unicas_ventas)+1) + ": Todas las categorías")
1587     print(str(len(categorias_unicas_ventas)+2) + ": Regresar al menú principal")
1588     eleccion = input()
1589     #Validar que lo que te dieron es una entrada válida
1590     eleccion = comprobar_opcion(eleccion, 1, len(categorias_unicas_ventas)+2)
1591     if int(eleccion) <= len(categorias_unicas_ventas): #Si la eleccion esta dentro de la lista de categorías
1592         print("Elegiste categoría " + categorias_unicas_ventas[int(eleccion)-1] + ".")
1593         if len(ventas_por_categoria[int(eleccion)-1]) >= 50:
1594             for numero_producto in range(0,50):
1595                 print("\nProducto "+ ventas_por_categoria[int(eleccion)-1][numero_producto][2]+ " con " + str(ventas_por_categoria[int(eleccion)-1][numero_producto][4])+ " en ventas globales. ID del producto")
1596                 str(ventas_por_categoria[int(eleccion)-1][numero_producto][4])+ " en ventas globales. ID del producto
1597             regresar_menu()
1598         elif len(ventas_por_categoria[int(eleccion)-1]) <= 50:
1599             print("\nLa cantidad de productos en esta opción es menor a 50; desplegando productos más vendidos:")
1600             for numero_producto in range(0, len(ventas_por_categoria[int(eleccion)-1])):
1601                 print("\nProducto "+ ventas_por_categoria[int(eleccion)-1][numero_producto][2]+ " con " + str(ventas_por_categoria[int(eleccion)-1][numero_producto][4])+ " en ventas globales. ID del producto")
1602                 str(ventas_por_categoria[int(eleccion)-1][numero_producto][4])+ " en ventas globales. ID del producto
1603             regresar_menu()
1604     elif int(eleccion) == len(categorias_unicas_ventas)+1 : #Si quiere ver todas las categorías
1605         print("\nElegiste ver los productos con más ventas de todas las categorías.")
1606         if len(conteos_ventas_ordenados) >= 50: #Elegir los primeros 50 si son más de 50 en la lista de productos vendidos
1607             for lugar_en_ventas in range(0,50):
1608                 print("\nProducto número " + str(lugar_en_ventas+1) + " en ventas: " +
1609                     conteos_ventas_ordenados[lugar_en_ventas][2] + ", con " +
1610                     str(conteos_ventas_ordenados[lugar_en_ventas][1])+
1611                     " ventas totales. ID del producto: " + str(conteos_ventas_ordenados[lugar_en_ventas][0]) + ".")
1612             regresar_menu()
1613         elif len(conteos_ventas_ordenados) <= 50:
1614             print("\nLa cantidad de productos en esta opción es menor a 50; desplegando productos más vendidos:")
1615             for lugar_en_ventas in range(0, len(conteos_ventas_ordenados)):
1616                 print("\nProducto número " + str(lugar_en_ventas+1) + " en ventas: " +
1617                     conteos_ventas_ordenados[lugar_en_ventas][2] + ", con " +
1618                     str(conteos_ventas_ordenados[lugar_en_ventas][1])+
1619                     " ventas totales. ID del producto: " + str(conteos_ventas_ordenados[lugar_en_ventas][0]) + ".")
1620             regresar_menu()
1621     elif int(eleccion) == len(categorias_unicas_ventas)+2 : #Si quiere regresar al menu principal
1622         menu_principal()

```

Figura 5. Elección de qué quiere ver de los productos más vendidos.

En caso de que el usuario haya elegido ver la información de los productos menos vendidos, se hace exactamente lo mismo a lo descrito anteriormente, pero usando índices negativos para mostrar la lista de manera inversa (Figura 6).

```

1623 #Opcion 50 productos menos vendidos por categoria
1624 if opcion_secundaria == "2":
1625     print("\nElige la opción para la que deseas ver los 50 productos menos vendidos:")
1626     for numero_categoria in range(1, len(categorias_unicas_ventas)+1):
1627         print(str(numero_categoria) + ": Categoría " + categorias_unicas_ventas[numero_categoria-1] )
1628     print(str(len(categorias_unicas_ventas)+1) + ": Todas las categorías")
1629     print(str(len(categorias_unicas_ventas)+2) + ": Regresar al menú principal")
1630     eleccion = input()
1631     #Validar que lo que te dieron es una entrada válida
1632     eleccion = comprobar_opcion(eleccion, 1, len(categorias_unicas_ventas)+2)
1633
1634     if int(eleccion) <= len(categorias_unicas_ventas): #Si la eleccion esta dentro de la lista de categorias
1635         print("\nElegiste categoría " + categorias_unicas_ventas[int(eleccion)-1] + ".")
1636         if len(ventas_por_categoria[int(eleccion)-1]) >= 50:
1637             for numero_producto in range(1, 51):
1638                 print("\nProducto " + ventas_por_categoria[int(eleccion)][-numero_producto][2] + " con " + str(ventas_p
1639                     str(ventas_por_categoria[int(eleccion)][-numero_producto][4]) + " en ventas globales. ID del pro
1640             regresar_menu()
1641         elif len(ventas_por_categoria[int(eleccion)-1]) <= 50:
1642             print("\nLa cantidad de productos en esta opción es menor a 50; desplegando productos menos vendidos:")
1643             for numero_producto in range(1, len(ventas_por_categoria[int(eleccion)])+1):
1644                 print("\nProducto " + ventas_por_categoria[int(eleccion)][-numero_producto][2] + " con " + str(ventas_p
1645                     str(ventas_por_categoria[int(eleccion)][-numero_producto][4]) + " en ventas globales. ID del pro
1646             regresar_menu()
1647         elif int(eleccion) == len(categorias_unicas_ventas)+1 : #Si quiere ver todas las categorias
1648             print("\nElegiste ver los productos con menos ventas de todas las categorías.")
1649             if len(conteos_ventas_ordenados) >= 50: #Elegir los primeros 50 si son más de 50 en la lista de productos ven
1650                 for lugar_en_ventas in range(1, 51):
1651                     print("\nProducto número " + str(conteos_ventas_ordenados[-lugar_en_ventas][4]) + " en ventas: " +
1652                         conteos_ventas_ordenados[-lugar_en_ventas][2] + ", con " +
1653                         str(conteos_ventas_ordenados[-lugar_en_ventas][1]) +
1654                         " ventas totales. ID del producto: " + str(conteos_ventas_ordenados[-lugar_en_ventas][0]) + ".")
1655                 regresar_menu()
1656             elif len(conteos_ventas_ordenados) <= 50:
1657                 print("\nLa cantidad de productos en esta opción es menor a 50; desplegando productos menos vendidos:")
1658                 for lugar_en_ventas in range(1, len(conteos_ventas_ordenados)+1):
1659                     print("\nProducto número " + str(conteos_ventas_ordenados[-lugar_en_ventas][4]) + " en ventas: " +
1660                         conteos_ventas_ordenados[-lugar_en_ventas][2] + ", con " +
1661                         str(conteos_ventas_ordenados[-lugar_en_ventas][1]) +
1662                         " ventas totales. ID del producto: " + str(conteos_ventas_ordenados[-lugar_en_ventas][0]) + ".")
1663                 regresar_menu()
1664             elif int(eleccion) == len(categorias_unicas_ventas)+2 : #Si quiere regresar al menu principal
1665                 menu_principal()
1666             #Preguntar al cliente si quiere hacer algo más o salir
1667             regresar_menu()
1668     #Opcion regresar al menú principal
1669     if opcion_secundaria == "3":
1670         menu_principal()

```

Figura 6. Despliegue de los resultados menos buscados por categoría o en total.

Análisis de búsquedas

Dado que el problema presentado en la presentación de los productos más y menos buscados por categoría es exactamente el mismo al del análisis de ventas, se siguió exactamente la misma lógica que la presentada anteriormente; lo único que cambió fue el nombre de cada variable y la lista que se usó (en vez de ser la lista de ventas, fue la lista de búsquedas). Puesto que es muy similar todo, se remite al lector a la sección anterior y se omitirá la explicación detallada del código. En caso de querer verlo, esta sección corresponde a las líneas 1672-1839 del código.

En resumen, los pasos que se siguieron, al igual que en la sección pasada, fueron:

1. Creación de lista de frecuencias de búsqueda por producto

2. Ordenamiento de mayor a menor
3. Añadir a cada sublista el nombre y categoría del producto
4. Separar cada elemento por categoría en una lista que almacenara una lista con los productos por categoría en cada posición.
5. Preguntar al usuario qué información quería ver (productos más o menos buscados)
6. Preguntar al usuario si quería ver la información por cada categoría que fue buscada o en total
7. Desplegar los resultados
8. Regresar al menú principal.

Análisis de reseñas

Para realizar el análisis de las reseñas de cada producto, lo primero que se hizo fue hacer una lista con el ID de cada producto vendido en la primera posición y una lista de sus reseñas en la segunda. Para esto, se hizo algo parecido a la lista de frecuencias de los dos incisos anteriores. Para cada venta, se evaluó si su ID estaba ya en la lista de reseñas. En caso de que no, se añadía su ID a la lista de IDs que ya estaban registradas, y a la lista que acumulaba los valores de reseña se le añadía una lista donde el primer elemento era el ID del producto y el segundo una lista con el valor de la reseña. A este último elemento es al que se le añadía cada nueva reseña en caso de que el producto ya estuviera registrado en la lista de IDs (líneas 1842-1858). Posteriormente, para cada producto en esa lista de reseñas, se calculó el promedio al sumar todos los elementos de la lista de la segunda posición dividido entre su longitud (líneas 1859-1861); dicho valor se añadió a la lista de cada producto. Posteriormente, se ordenó la lista de mayor a menor promedio en reseñas (1863-1875) y se le añadió a cada producto el nombre y categoría de la misma manera que se hizo en las secciones anteriores (líneas 1878-1884). Al finalizar esto, se generó una lista donde cada elemento poseía una lista con el ID de cada producto, sus reseñas acumuladas, su reseña promedio, nombre y categoría (líneas 1886-1888, **Figura 7**)

```

1841     if opcion_informacion == "3":
1842         #Hacer lista con las reseñas de cada producto acumuladas
1843         reseñas_acumuladas = [] #inicializar lista que va a tener [id_producto, lista con reseñas del producto]
1844         ids_reseñas = [] #Inicializar una lista que me va a servir para saber cuando un producto ya se haya contado o no
1845         for venta in lifestore_sales:
1846             #Cuando no este en la lista
1847             if venta[1] not in ids_reseñas:
1848                 ids_reseñas.append(venta[1])
1849                 lista_intermedia_inicial = [venta[1],venta[2]] #Hacer una lista que contenga [ID],[Reseña]
1850                 reseñas_acumuladas.append(lista_intermedia_inicial)
1851             #Cuando el ID ya esté en la lista de búsquedas (ya se registro)
1852             elif venta[1] in ids_reseñas:
1853                 #Añadir el valor de la reseña
1854                 for producto in reseñas_acumuladas: #Buscar en reseñas_acumuladas el ID que corresponda con la reseña
1855                     if producto[0] == venta[1]:
1856                         producto[1].append(venta[2]) #Añadir la reseña
1857                         break
1858
1859         #Sacar promedio de reseña de cada producto
1860         for producto in reseñas_acumuladas:
1861             producto.append(sum(producto[1])/len(producto[1]))
1862
1863         #Ordenar la lista con base en el promedio
1864         reseñas_acumuladas_ordenadas = []
1865
1866         while reseñas_acumuladas:
1867             maximo = reseñas_acumuladas[0][2]
1868             lista_actual = reseñas_acumuladas[0]
1869             for producto in reseñas_acumuladas:
1870                 if producto[2] > maximo:
1871                     maximo = producto[2]
1872                     lista_actual = producto
1873             reseñas_acumuladas_ordenadas.append(lista_actual)
1874             reseñas_acumuladas.remove(lista_actual)
1875         del reseñas_acumuladas #Eliminar lista desordenada que quedó vacía
1876
1877
1878         for producto in reseñas_acumuladas_ordenadas:
1879             for id in lifestore_products:
1880                 if producto[0] == id[0]:
1881                     producto.append(id[1]) #añadir nombre
1882                     producto.append(id[3]) #añadir categoría
1883                     break
1884             continue
1885
1886         """Producto final :
1887         Para cada producto vendido:
1888         reseñas_acumuladas_ordenadas = [id_producto, [reseñas de ese producto], promedio, nombre, categoría]
1889         """

```

Figura 7. Creación de lista ordenada de reseña promedio por producto.

Una vez que esta lista está hecha, se le pregunta al cliente qué información quiere ver: los 20 productos con mejores o peores reseñas (o regresar al menú principal). Si es que desea ver la primera opción, se despliega la información de las 20 primeras posiciones de la lista previamente hecha, en caso de tener esta más de 20 elementos (líneas 1902-1904). En caso de que posea menos de 20 elementos, se indica esto y se despliega la información de toda la lista ordenada de mayor a menor (líneas 1906-1909). Si el usuario desea ver las peores reseñas, se hace exactamente lo mismo pero desplegando la lista al revés utilizando los índices de forma negativa (líneas 1911-1920). Al final de todas las opciones se le pregunta al usuario si quiere regresar al menú principal o salir del programa (**Figura 8**).

```

1891 #Menú de categorías menos buscados
1892 print("""Elegiste información de la reseña de los productos. Por favor es
1893     1: 20 productos con mejores reseñas
1894     2: 20 productos con peores reseñas
1895     3: Regresar al menú principal""")
1896 eleccion = input()
1897
1898 eleccion = comprobar_opcion(eleccion, 1, 3)
1899
1900 if int(eleccion) == 1 : #Ver los 20 con mejores reseñas
1901     print("\nElegiste ver los 20 productos con mejores reseñas: ")
1902     if len(reseñas_acumuladas_ordenadas) >= 20:
1903         for numero_producto in range(0,20):
1904             print("\nProducto "+ reseñas_acumuladas_ordenadas[numero_prod
1905             regresar_menu()
1906     elif len(reseñas_acumuladas_ordenadas) < 20:
1907         print("\nLa cantidad de productos es menor a 20; desplegando prod
1908         for numero_producto in range(0,len(reseñas_acumuladas_ordenadas))
1909             print("\nProducto "+ reseñas_acumuladas_ordenadas[numero_prod
1910             regresar_menu()
1911     elif int(eleccion) == 2 : #Si quiere ver los 20 con peores reseñas
1912         print("\nElegiste ver los 20 productos con peores reseñas:")
1913         if len(reseñas_acumuladas_ordenadas) >= 20:
1914             for numero_producto in range(1,21):
1915                 print("\nProducto "+ reseñas_acumuladas_ordenadas[-numero_pro
1916                 regresar_menu()
1917         elif len(reseñas_acumuladas_ordenadas) < 20:
1918             print("\nLa cantidad de productos es menor a 20; desplegando prod
1919             for numero_producto in range(1,len(reseñas_acumuladas_ordenadas)+
1920                 print("\nProducto "+ reseñas_acumuladas_ordenadas[-numero_pro
1921                 regresar_menu()
1922     elif int(eleccion) == 3 : #Si quiere regresar al menu principal
1923         menu_principal()
1924

```

Figura 8. Interacción con el usuario con respecto a la información de reseñas solicitada

Análisis de ingresos

Finalmente, en la sección de análisis de ingresos, lo primero que hice fue hacer elegir al usuario si quería ver el ingreso total anual de la empresa, su venta promedio mensual o regresar al menú principal (líneas 1925-1931). Tras validar el input (línea 1933), cree una lista que asociara el valor numérico de cada mes, su nombre, y la cantidad de días; estos datos serán usados posteriormente para mejorar la presentación de resultados (**Figura 9**).

```

1925     if opcion_informacion == "4":
1926         print("\nElegiste información de los ingresos de la empresa")
1927         opcion_secundaria = input("""Elige la información que deseas ver:
1928             1: Ingreso total anual de la empresa
1929             2: Venta promedio mensual
1930             3: Regresar al menú principal
1931         """)
1932         #Comprobar que la opción sea correcta
1933         opcion_secundaria = comprobar_opcion(opcion_secundaria,1,3)
1934
1935         #Crear lista con equivalencia del nombre del mes y su número de días
1936         meses = [{"01","enero",31},
1937                 [{"02","febrero",28},
1938                 [{"03","marzo",31},
1939                 [{"04","abril",30},
1940                 [{"05","mayo",31},
1941                 [{"06","junio",30},
1942                 [{"07","julio",31},
1943                 [{"08","agosto",31},
1944                 [{"09","septiembre",30},
1945                 [{"10","octubre",31},
1946                 [{"11","noviembre",30},
1947                 [{"12","diciembre",31}
1948         ]

```

Figura 9. Interacción con el usuario y creación de lista con información de los meses

Posteriormente, generé una lista que tuviera en la primera posición el mes y en la segunda una lista con las ventas de ese mes. Para esto, se iteró sobre cada elemento en la lista de ventas para asignar a cada venta a su mes correspondiente (líneas 1949-1963). Dado que había ventas que no eran del 2020, añadí un filtro antes para comprobar que la venta fuera del 2020 (línea 1954). Posteriormente, obtuve el valor total de ventas por mes; para esto, para cada mes, inicialicé el valor de venta a cero y sumé el precio del producto de cada venta registrada en ese mes en la lista `lifestore_products` al valor de venta mensual para cada mes. Finalmente, este valor se agregó a la lista de cada mes (líneas 1965-1973). Después de esto, obtuve el valor promedio al dividir la venta total previamente obtenida entre el total de días de cada mes de la lista `meses` de la **figura 9** (líneas 1980-1982). Tras hacer esto, ordené la lista de mayor a menor de acuerdo a la venta promedio de cada mes (líneas 1984-1994), obteniendo así la lista `ventas_mensuales`, explicada en las líneas 1995-19997. Finalmente, obtuve el ingreso total al sumar la venta total de cada mes (líneas 2000-2002, **Figura 10**).

```

1949     ventas_mensuales = []
1950     meses_ids = []
1951     #Separar las ventas por meses eliminando los que no son del 2020
1952     #Modificar esta parte si se trabaja con datos de varios años
1953     for venta in lifestore_sales:
1954         if venta[3][6:10] == "2020":
1955             if venta[3][3:5] not in meses_ids:
1956                 meses_ids.append(venta[3][3:5])
1957                 ventas_mensuales.append([venta[3][3:5], [venta[1]]])
1958             elif venta[3][3:5] in meses_ids:
1959                 for mes in ventas_mensuales:
1960                     if venta[3][3:5] == mes[0]:
1961                         mes[1].append(venta[1])
1962                         break
1963     #Hasta aqui ventas_mensuales = [mes,[ids de productos vendidos]]
1964     #Obtener el total de ventas por mes
1965     for mes in ventas_mensuales:
1966         venta_del_mes = 0
1967         for producto_vendido in mes[1]:
1968             for producto in lifestore_products:
1969                 if producto[0] == producto_vendido: #Si ID de lifestore_products == producto vendido
1970                     venta_del_mes += producto[2] #Sumar a la venta del mes el precio de ese producto
1971                     break
1972         mes.append(venta_del_mes)
1973     #Hasta aqui ventas_mensuales = [mes,[ids de productos vendidos], venta_total_del_mes]
1974     #Obtener el promedio mensual y añadir el nombre de cada mes a la lista
1975     for mes_de_ventas in ventas_mensuales:
1976         for mes in meses:
1977             if mes_de_ventas[0] == mes[0]:
1978                 mes_de_ventas.append(mes[1])
1979                 mes_de_ventas.append(mes[2])
1980     #Obtener promedio mensual
1981     for mes in ventas_mensuales:
1982         mes.append(round(mes[2]/mes[4],2))
1983     #Ordenar la lista con base en el promedio mensual
1984     ventas_mensuales_ordenadas = []
1985     while ventas_mensuales:
1986         maximo = ventas_mensuales[0][5]
1987         lista_actual = ventas_mensuales[0]
1988         for mes in ventas_mensuales:
1989             if mes[5] > maximo:
1990                 maximo = mes[5]
1991                 lista_actual = mes
1992         ventas_mensuales_ordenadas.append(lista_actual)
1993         ventas_mensuales.remove(lista_actual)
1994     del ventas_mensuales #Eliminar lista desordenada que quedó vacía
1995     """Producto final:
1996     ventas_mensuales = [mes,[ids de productos vendidos], venta_total_del_mes, NombreDelMes, DiasDeEseMes, PromedioMensualDeVentas]
1997     Ordenado con base en promedio mensual
1998     """
1999     #Obtener ingreso total
2000     ingreso_anual = 0
2001     for mes in ventas_mensuales_ordenadas:
2002         ingreso_anual += mes[5]

```

Figura 10. Obtención de lista de promedio de venta mensual ordenada.

Para terminar, dependiendo de la elección del usuario, se despliega el ingreso total de la empresa de ese año (líneas 2005-2007), la lista del promedio mensual ordenada de mayor a menor (líneas 2010-2013) o se regresa al menú principal (línea 2018, **Figura 11**).


```

2004 #opcion ingreso total anual de la empresa
2005 if opcion_secundaria == "1":
2006     print("\nEl ingreso total de la empresa durante este año es de " + str(ingreso_anual)+ " pesos mexicanos.")
2007     #Preguntar al cliente si quiere hacer algo más o salir
2008     regresar_menu()
2009 #opcion venta promedio mensual
2010 if opcion_secundaria == "2":
2011     print("\nElegiste ver la venta promedio mensual de la empresa. Desplegando la venta promedio mensual de mayor a menor de este año: ")
2012     for mes in ventas_mensuales_ordenadas:
2013         print("\n+ mes[3]+": "+str(mes[5])+" pesos mexicanos.")
2014
2015     #Preguntar al cliente si quiere hacer algo más o salir
2016     regresar_menu()
2017 #opcion regresar al menú principal
2018 if opcion_secundaria == "3":
2019     menu_principal()
2020

```

Figura 11. Presentación de los resultados al usuario.

Solución al problema

Productos más y menos vendidos

Los productos más vendidos se muestran correctamente en la pantalla al elegir esa opción. Por mostrar un ejemplo, podemos observar el top 5 de productos más vendidos en todas las categorías:

- Producto número 1 en ventas: SSD Kingston A400, 120GB, SATA III, 2.5", 7mm, con 50 ventas totales. ID del producto: 54.
- Producto número 2 en ventas: Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth, con 42 ventas totales. ID del producto: 3.
- Producto número 3 en ventas: Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake), con 20 ventas totales. ID del producto: 5.
- Producto número 4 en ventas: Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD, con 18 ventas totales. ID del producto: 42.
- Producto número 5 en ventas: SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm, con 15 ventas totales. ID del producto: 57.

Así mismo, podemos ver el top 5 de productos menos vendidos:

- Producto número 42 en ventas: HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro, Inalámbrico, USB, 3.5mm, Negro, con 1 ventas totales. ID del producto: 94.
- Producto número 41 en ventas: Cougar Audífonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro., con 1 ventas totales. ID del producto: 89.

- Producto número 40 en ventas: Logitech Audífonos Gamer G332, Alámbrico, 2 Metros, 3.5mm, Negro/Rojo, con 1 ventas totales. ID del producto: 84.
- Producto número 39 en ventas: TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro, con 1 ventas totales. ID del producto: 67.
- Producto número 38 en ventas: TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Widescreen, Negro, con 1 ventas totales. ID del producto: 66.

Con esta información, podemos observar que los productos menos vendidos poseen todos 1 venta en total en lo que va del año, en comparación de los más vendidos que tienen más de 15.

Productos con más y menos búsquedas

Al igual que con las ventas, los productos más buscados se presentan correctamente. Como ejemplo, aquí podemos ver el top 5 de productos buscados en todas las categorías:

- Producto número 1 en búsquedas: SSD Kingston A400, 120GB, SATA III, 2.5", 7mm, con 263 búsquedas totales. ID del producto: 54.
- Producto número 2 en búsquedas: SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm, con 107 búsquedas totales. ID del producto: 57.
- Producto número 3 en búsquedas: Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD, con 60 búsquedas totales. ID del producto: 29.
- Producto número 4 en búsquedas: Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth, con 55 búsquedas totales. ID del producto: 3.
- Producto número 5 en búsquedas: Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire, con 41 búsquedas totales. ID del producto: 4.

Cabe destacar que 3 de los 5 productos más buscados se encuentran en los productos más comprados de la tienda (ID 3, 54 y 57); sería interesante evaluar la existencia de una correlación significativa entre búsqueda y compra de productos con una mayor cantidad de datos, pues esto nos permitiría estimar las ventas futuras con base en la búsqueda de los usuarios en tiempo real.

Así mismo, los productos con menos búsquedas de los que se tiene registro en la base de datos (es decir, que sea diferente a 0), se presentan correctamente. Como ejemplo, podemos ver el top 5 de productos menos buscados en todas las categorías:

- Producto número 56 en búsquedas: Ginga Audífonos con Micrófono G118ADJ01BT-RO, Bluetooth, Alámbrico/Inalámbrico, 3.5mm, Rojo, con 1 búsquedas totales. ID del producto: 93.

- Producto número 55 en búsquedas: Ghia Bocina Portátil BX800, Bluetooth, Inalámbrico, 2.1 Canales, 31W, USB, Negro, con 1 búsquedas totales. ID del producto: 80.
- Producto número 54 en búsquedas: Samsung Smart TV LED 43, Full HD, Widescreen, Negro, con 1 búsquedas totales. ID del producto: 70.
- Producto número 53 en búsquedas: SSD Samsung 860 EVO, 1TB, SATA III, M.2, con 1 búsquedas totales. ID del producto: 59.
- Producto número 52 en búsquedas: Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel, con 1 búsquedas totales. ID del producto: 45.

En este caso, no podemos observar una relación directa con los productos menos vendidos; esto puede deberse a que muchos productos pueden presentar ventas o no sin que necesariamente esto implique que sean buscados, y viceversa.

Productos con buenas y malas reseñas

Los productos con mejor reseña se muestran correctamente en la pantalla al elegir esa opción. Como ejemplo, podemos observar el top 5 de productos con mejor reseña en todas las categorías:

- Producto Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache (ID: 1) con un promedio de 5.0 en reseñas.
- Producto Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake) (ID: 6) con un promedio de 5.0 en reseñas.
- Producto Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake) (ID: 7) con un promedio de 5.0 en reseñas.
- Producto Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake) (ID: 8) con un promedio de 5.0 en reseñas.
- Producto Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Express 3.0 (ID: 11) con un promedio de 5.0 en reseñas.

Es interesante observar que los productos con mejores reseñas no necesariamente corresponden a los productos más vendidos o con mayores búsquedas.

Así mismo, los productos con peores reseñas de los que se tiene registro en la base de datos (es decir, que se han comprado), se presentan correctamente. Como ejemplo, podemos ver el top 5 de productos con peores reseñas en todas las categorías:

- Producto Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel (ID: 45) con un promedio de 1.0 en reseñas.
- Producto Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0 (ID: 17) con un promedio de 1.0 en reseñas.

- Producto Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD (ID: 31) con un promedio de 1.8 en reseñas.
- Producto Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel (ID: 46) con un promedio de 2.0 en reseñas.
- Producto Cougar Audífonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro. (ID: 89) con un promedio de 3.0 en reseñas.

Ventas mensuales

El ingreso de la empresa en lo que va del año es de 25010.47 pesos mexicanos, lo cual se muestra correctamente. Así mismo se presenta el promedio mensual de ventas de mayor a menor como se espera:

- abril: 6443.17 pesos mexicanos.
- marzo: 5313.84 pesos mexicanos.
- febrero: 3933.54 pesos mexicanos.
- enero: 3878.61 pesos mexicanos.
- mayo: 3101.13 pesos mexicanos.
- junio: 1231.63 pesos mexicanos.
- julio: 869.32 pesos mexicanos.
- septiembre: 139.97 pesos mexicanos.
- agosto: 99.26 pesos mexicanos.

Es interesante observar que las ventas mensuales promedio alcanzan su pico en abril (**Figura 12**). Si bien no se tienen datos suficientes para observar las ventas en los meses de septiembre, octubre, noviembre y diciembre, este comportamiento, si se repite a lo largo de los años, podría ser muy útil para ajustar el stock de productos en tienda a lo largo del año.

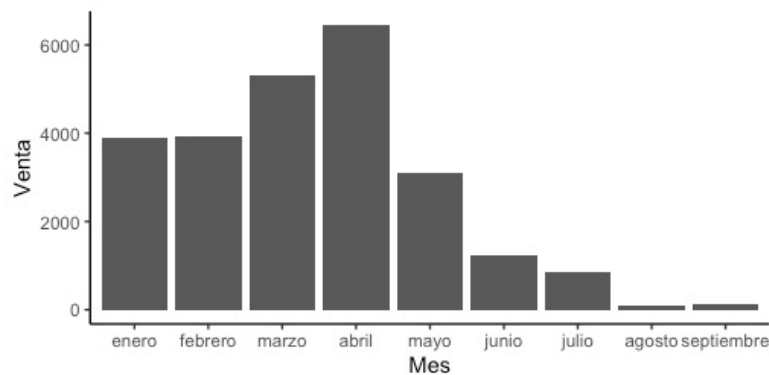


Figura 12. Venta promedio mensual.

Estrategia sugerida

Al obtener las listas creadas en cada uno de los pasos, se puede observar la venta, búsqueda, reseña, stock y las devoluciones de cada producto en la tienda. Con el objetivo de observar de manera global el comportamiento en cada uno de los rubros de los productos ofrecidos en la tienda, generé un mapa de calor en el cual se puede ver cómo se encuentra cada producto por cada área (**Figura 13A**). En esta figura, los ID de los productos se encuentran a la derecha y se puede observar los productos cuyo valor en cada rubro se separa por más (color rojo) o menos (color azul) de una desviación estandar del promedio global de cada área. De esta manera, podemos ver de manera rápida los productos cuyas ventas, stock, búsquedas, reseñas o devoluciones son mayores o menores que el promedio, identificando rápidamente productos cuyo desempeño sea mejor o peor que el promedio en diversas áreas, facilitando la integración de la información.

El primer dato que salta a la vista es la presencia de una gran cantidad de recuadros grises (indicativo de ausencia de información) en ventas, búsquedas y reseñas; es decir, la existencia de muchos productos que no se venden y no se buscan, cuyo stock en tienda se encuentra relativamente cerca del promedio de almacén. Debido a esto, la estrategia que sugiero se basará en dos grupos de productos: productos rezagados (que no tienen ventas ni búsquedas en el año) y productos activos (que se han vendido y han sido buscados a lo largo del año).

Productos rezagados

Los productos rezagados en tienda son 38 (IDs: "14" "16" "19" "20" "23" "24" "30" "32" "34" "36" "37" "38" "41" "43" "53" "55" "58" "61" "62" "64" "65" "68" "69" "71" "72" "75" "77" "78" "79" "81" "82" "83" "86" "87" "88" "90" "92" y "96"), los cuales no presentan búsquedas ni ventas. De manera inicial, estos productos deberían ser los primeros en retirar del mercado o reducir su stock a menos de 10. Como podemos ver en la distribución de stock de este grupo de productos (**Figura 13B**), si bien la gran mayoría tiene una cantidad en almacén de menos de 50, existen 6 productos (IDs 41, 68, 92, 69, 64 y 37) que se encuentran en gran cantidad. Se recomienda poner en oferta paulatinamente estos productos, comenzando por los 6 mencionados, para eliminarlos del almacén y restringir su compra, así como discutir con los socios su posible eliminación de la tienda una vez que se agoten.

Productos activos

Dentro de los productos activos (**Figura 13C**), podemos observar claramente que existen 3 productos, ubicados en la parte superior del gráfico, cuya puntuación en reseña es particularmente baja, y cuya tasa de devoluciones (calculado como número de devoluciones/ total de ventas del producto) es alta. Estos productos (IDs 46,17 y 45) deberían ser observados cuidadosamente para eliminarlos potencialmente de la tienda, pues probablemente tengan una mala calidad y la devolución representa una pérdida para la empresa tanto en ingreso como en popularidad.

Por otro lado, existen productos cuyas ventas son altas pero no se encuentran en grandes proporciones en el almacén y viceversa. Esto representa una situación peligrosa que puede resultar en el rezago de algunos productos y en la deficiencia de otros. Es por esto que se sugiere limitar la compra de los productos 4 y 67 hasta que su stock vuelva a estar cerca del stock promedio, para evitar una acumulación del producto. Así mismo, se recomienda aumentar la cantidad de productos con el ID 5 y 42, los cuales tienen muchas ventas y una presencia en inventario promedio. Al aumentar considerablemente su adquisición con los proveedores podemos evitar una escasez del producto.

Finalmente, se observan dos productos en la parte inferior cuya venta, búsqueda y stock es alto; esto representa una buena estrategia donde los productos más vendidos y buscados se encuentran en mayor cantidad en almacén (IDs 3 y 54). Debe procurarse que esto siga así para este tipo de productos.

Por último, se recomienda seguir evaluando en años consecutivos el comportamiento de ventas mensuales. Si la tendencia sigue, se sugiere adaptarse a la dinámica de compras del consumidor y adaptar los productos en stock de acuerdo a la época del año; es decir, a partir de noviembre ir incrementando la adquisición de productos hasta llegar a marzo (1 mes antes del pico de ventas); a partir de ahí ir disminuyendo la adquisición de los mismos para adaptarse a la temporada de ventas flojas. De esta manera, se evita el almacén innecesario de productos y se evita el desabasto en temporadas de ventas altas.

A manera de opinión personal, sería importante esperar a que el año termine para tomar las decisiones definitivas de qué productos retirar del mercado, para contar con la información de un ciclo anual completo. Esto se debe a que probablemente algunos productos en la tienda se vendan únicamente en ciertos meses del año, por lo que si bien actualmente no se venden, lo podrían hacer en los meses restantes. Si bien la cantidad de datos actuales es insuficiente, sería interesante realizar este tipo de análisis en el futuro.

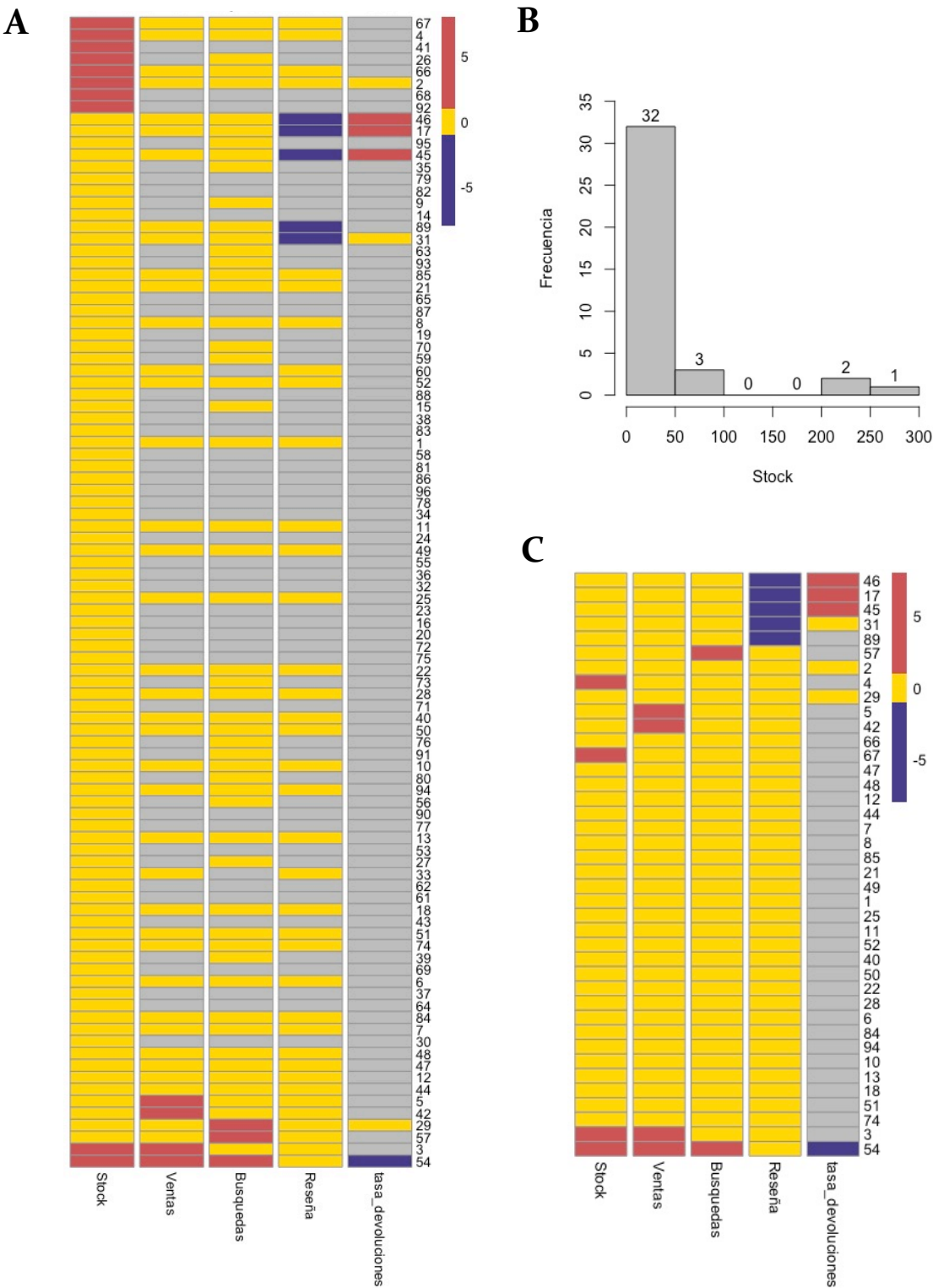


Figura 13. Evaluación de cada producto en tienda en distintos rubros. A) Mapa de calor donde se observa si la cantidad en cada producto en cada área se encuentra a más de una desviación estándar del promedio hacia arriba (color rojo), hacia abajo (color azul), o

si se encuentra dentro de ese rango (color amarillo. El color gris indica ausencia de datos. B) Histograma del stock de los productos clasificados como rezagados. C) Mapa de calor con las mismas características que el panel A, pero únicamente observando los productos activos.

Conclusiones

Utilizando las listas generadas por el sistema de análisis EMTECH, se logró observar exitosamente el comportamiento de los productos a través de 5 diferentes áreas: stock, ventas, búsquedas, reseñas y devoluciones. A partir de esto, la información se integró y se lograron hacer sugerencias a partir de decisiones inteligentes basadas en los datos observados. Estas se pueden resumir en:

- Eliminar paulatinamente los productos en almacén catalogados como rezagados mediante ofertas en tienda, otorgando prioridad a 6 productos específicos.
- Evaluar el comportamiento de productos rezagados a lo largo del año; en caso de no cambiar, retirar del mercado.
- Retirar 3 productos activos con bajas reseñas y altas tasas de devolución.
- Hacer planes de adquisición de productos activos personalizados para cada uno; aumentar el stock de productos con altas ventas y búsquedas y reducir el mismo en los que actualmente tienen una alta cantidad en inventario y una venta promedio.
- Observar el comportamiento de las ventas promedio en años subsecuentes para implementar un plan dinámico de mercado que se adapte a las cantidad diferencial de compras de los clientes a lo largo del año.

Con el paso de los años y utilizando el sistema de análisis EMTECH, se podrán recabar más datos y aumentar la precisión y utilidad de este programa, incrementando las ventas y manteniendo a la empresa adaptada a las necesidades del cliente en todo momento.