



Proyecto 1: Análisis de la tienda “Lifestore”

EMTECH

GERARDO ERICK VILLA AGUILAR

Índice

Introducción	2
Definición del código.....	3
Solución al problema.....	19
Conclusión	21

Introducción

El presente trabajo es el análisis de un conjunto de datos de un almacén de productos de computación llamado Lifestore. Este almacén presenta problemas de acumulación de inventario y productos que han recibido menos búsquedas, por lo que ha solicitado un análisis en la base de datos para saber qué productos son los más vendidos, los menos vendidos, los más buscados y los menos buscados, así como un reporte de sus ventas. Esto con el fin de mejorar el flujo de productos. La base de datos está constituida por un registro de ventas de los productos, un registro de todos los productos y sus características y un registro de las búsquedas que se han realizado sobre los productos.

La solución propuesta es un software que relaciones y los datos y los presente de una manera intuitiva y estética lo que facilitará el análisis.

Definición del código

El código completo se encuentra en el siguiente repositorio de GitHub:
<https://github.com/gerardoev/proyecto1-EMTECH>

Login

En esta parte se implementa el login, primero generamos una estructura que contiene los usuarios y passwords permitidos. Limpiamos la consola para simular un cambio de pantalla.

```
#Login

usuarios = [['user@lifestore.com', 'user123456'], ['marioh@lifestore.com', 'mario123456'],
['gerardo@lifestore.com', 'gerardo123456']]

print('\n' * 50)
```

Solicitamos al usuario su usuario y password

```
usr = input('Ingresa tu usuario:')
password = input('Ingresa tu password:')
```

Creamos unas variables que nos ayudarán a identificar si el usuario o password introducidos son correctos

```
encontrado = False
login_success = False
```

Con ayuda de un ciclo for, iteramos sobre los valores permitidos y comparamos con los valores introducidos

```
for usuario in usuarios:
    if usuario[0] == usr:
        encontrado = True
        if usuario[1] == password:
            login_success = True
```

Mientras el usuario o contraseña sean incorrectos, pedimos nuevamente el usuario y contraseña e indicamos cual de los casos ha fallado. Para indicar el error, creamos una variable 'error' la cual nos ayudará a almacenar el error una vez que lo hayamos identificado. Una vez que ambos (Usuario y password) son verdaderos, se sale del ciclo y continua con la ejecución del programa, Simulando un login exitoso.

```

error = ''
while encontrado == False or login_success == False:
    print('\n'*50)
    if encontrado == False:
        error = 'El usuario no existe'
    else:
        error = 'La contraseña es incorrecta'
    print(f'El login ha fallado debido a lo siguiente:{error}\nVuelve a intentarlo o pulsa
ctrl + c para abortar')
    print('\n'*2)

    encontrado = False
    login_success = False

    usr = input('Ingresa tu usuario:')
    password = input('Ingresa tu password:')

    for usuario in usuarios:
        if usuario[0] == usr:
            encontrado = True
            if usuario[1] == password:
                login_success = True

```

Listar productos con mayores búsquedas

Para implementar este requerimiento, vamos a generar una lista ordenada que contenga el id del producto y la cantidad de ventas que tuvo ese producto, al ser una lista ordenada, si accedemos a los primeros elementos, estamos accediendo a los más vendidos, si accedemos a los últimos, estamos accediendo a los menos vendidos.

Guardamos en una variable el número de productos, ya que nos va a servir para recorrer los productos

```
num_products = len(lifestore_products)
```

Definimos una lista 'frecs_ventas' del tamaño de los productos en la cual cada posición representa un id de producto - 1, es decir el elemento 0. Esta estructura nos servirá para contar las veces que se ha comprado cada producto.

También tenemos la lista 'frec_prod'. Esta lista será la unión de la anterior con su id real. Tener el id explícito nos ayudará al momento de ordenar la lista a no perder este id.

```
frecs_ventas = []
frec_prod = []
```

Iniciamos frecs_ventas en 0's

```
# Generamos una lista con 0's para ir contando las frecuencias
for i in range(num_products):
    frecs_ventas.append(0)
```

Obtenemos las frecuencias recorriendo las todas las ventas, por cada venta obtenemos su id y este a su vez es la posición en nuestra lista – 1. Una vez localizada la posición simplemente aumentamos en 1

```
# Obtenemos el id del producto, que al mismo tiempo corresponde a
# la posición de la lista generada - 1, una vez localizada la posición,
# incrementamos en 1 nuestro contador
for sale in lifestore_sales:
    if sale[4] != 1:
        frecs_ventas[sale[1] - 1] += 1
```

Llenamos la segunda lista con los valores de la primera más su id. Quedando de la forma: [[frecuencia, id_prod]]

Se coloca la frecuencia primero porque la función sort ordena tomando en cuenta el primer elemento en una lista de listas.

```
#Generamos una nueva lista que contenga las frecuencias y los id
for prod in range(1, num_products + 1):
    frec_prod.append([freces_ventas[prod - 1], prod])
```

Ordenamos la lista obtenida con la función sort

```
#Ordenamos la lista de menor a mayor
frec_prod.sort()
```

Esta línea solo sirve al momento de presentar la información ya que se presentará en forma de tabla y se simularán celdas

```
#Imprimimos los 15 productos más vendidos
tamano_celda = 16
```

Listar productos con mayores búsquedas

El procedimiento es similar al anterior, sin embargo es necesario usar listas independientes debido a que la presentación de la información se hará hasta el final mediante un menú interactivo. Recordemos que para obtener los productos más buscados simplemente debemos leer los primeros elementos de esta lista y para obtener los menos buscados, debemos leer los últimos elementos de esta misma.

```
frecs_searches = []
frec_search = []

#Lista con ceros para contar las frecuencias
for i in range(num_products):
    frecs_searches.append(0)

#Contamos las frecuencias
for search in lifestore_searches:
    frecs_searches[search[1] - 1] += 1

#Generamos una nueva lista que contenga las frecuencias y los id
for prod in range(1, num_products + 1):
    frec_search.append([frecs_searches[prod - 1], prod])

frec_search.sort()
```

Listado de productos con mejores y peores reseñas.

Este requerimiento es similar al anterior, sin embargo, para obtener la reseña por producto debemos obtener el promedio de las reseñas de ese producto. Por lo tanto, para obtener esta lista necesitamos de dos estructuras.

La primera estructura tendrá la forma `[[suma, cuenta]]` y nos servirá para acumular la reseña y llevar una cuenta del número total de reseñas, al llenar esta lista podremos obtener el promedio simplemente dividiendo el primer elemento entre el segundo. En esta lista cada sublista representa a cada producto en orden, por lo que esta lista es de tamaño `n` productos.

En el siguiente ejemplo creamos la lista y la llenamos con ceros

```
#Obtener el promedio de reseña por producto
#Creamos una estructura que permita acumular la suma y la cuenta por producto

prom_prods = [] #[[suma, cuenta]]
for prod in range(0, num_products):
    prom_prods.append([0,0])
```

Enseguida, recorreremos cada producto y vamos acumulando las reseñas y la cantidad de reseñas.

```
#Acumulamos las reseñas
for sale in lifestore_sales:
    if sale[4] != 1:
        indice_prods = sale[1] - 1
        prom_prods[indice_prods][0] += sale[2]
        prom_prods[indice_prods][1] += 1
```

La segunda lista que necesitamos es una estructura similar al requerimiento anterior, [[promedio_reseña, id_producto]]. Esta lista estará ordenada y la usaremos al final para obtener los primeros productos en caso de requerir los productos con mejores reseñas o los últimos, en caso de requerir aquellos con las peores reseñas

Al llenar esta lista, vamos obteniendo el promedio de la reseña.

```
#Realizamos la división para obtener el promedio de reseña por producto, el resultado lo
guardamos en una lista con id y prom reseña para después ordenarla
res_prod = [] #[[prom_res, id_prod]]
for id_prod in range(1, num_products + 1):
    prod = prom_prods[id_prod - 1]
    if prod[0] == 0:
        res_prod.append([0, id_prod])
    else:
        res_prod.append([prod[0] / prod[1], id_prod])

res_prod.sort()
```

```
#Creamos la estructura para guardar los ingresos por meses
meses = []
prom_mens = []
ventas = []
for i in range(12):
    meses.append(0)
    prom_mens.append(0)
    ventas.append(0)

#Sumamos los ingresos por mes
for prod in lifestore_sales:
    dev = prod[4]
    if dev != 1:
        fecha = prod[3] #'01/01/2001'
        anio = fecha[6:]
        if anio == '2020':
            mes = int(fecha[3:5])
            id_prod = prod[1]
            ingreso = lifestore_products[id_prod - 1][2]
            meses[mes - 1] += ingreso
            ventas[mes - 1] += 1

ventas_mes = []
for i in range(12):
    ventas_mes.append([ventas[i], i+1])
    if ventas[i] == 0:
        continue
    prom_mens[i] = meses[i] / ventas[i]
```


Total de ingresos mensuales

Para este requerimiento necesitamos una estructura 'meses', que será de tamaño 12 y cada posición representa un mes del año en orden. Esta nos servirá para ir acumulando cada venta por mes.

La estructura 'ventas' será de tamaño 12, también representa cada mes y nos servirá para ir contando cuántas ventas se realizan por mes, esta estructura también nos ayuda a sacar el promedio.

La estructura 'prom_mens', contendrá el promedio por mes, también es de tamaño 12, necesitamos que sea una estructura independiente porque al momento de presentar la información, necesitamos tener disponible también los datos anteriores.

En el siguiente fragmento de código creamos las estructuras y las inicializamos en cero.

```
#Creamos la estructura para guardar los ingresos por meses
meses = []
prom_mens = []
ventas = []
for i in range(12):
    meses.append(0)
    prom_mens.append(0)
    ventas.append(0)
```

Total anual

Este requerimiento lo cumplimos únicamente sumando todos los elementos de meses. Esto se realiza más adelante en la parte de presentación de la información

Meses con más ventas al año

Recorremos las ventas, identificamos de qué mes es y lo mapeamos directamente a la estructura.

```
#Sumamos los ingresos por mes
for prod in lifestore_sales:
    dev = prod[4]
    if dev != 1:
        fecha = prod[3] #'01/01/2001'
        anio = fecha[6:]
        if anio == '2020':
            mes = int(fecha[3:5])
            id_prod = prod[1]
            ingreso = lifestore_products[id_prod - 1][2]
            meses[mes - 1] += ingreso
            ventas[mes - 1] += 1
```

Ventas promedio mensuales

Obtenemos el promedio de una manera similar a procedimientos anteriores, con esto se cumple el requerimiento de las ventas promedio mensuales.

```
ventas_mes = []
for i in range(12):
    ventas_mes.append([ventas[i], i+1])
    if ventas[i] == 0:
        continue
    prom_mens[i] = meses[i] / ventas[i]
```

Menú interactivo

Para realizar el menú, hacemos un while que evalúe la opción introducida por el usuario hasta que ingrese -1. Dentro de las opciones también existen submenús para facilitar la visualización

```
resp = 1
while resp != '-1':
    print('\n' * 50)
    print("""
*****
*                                     *
*           Menú                     *
*                                     *
*****

1. Productos con mayores ventas y búsquedas
2. Productos con menores ventas y búsquedas por categoría
3. Reporte de productos con mejores y peores reseñas
4. Reporte de ingresos y ventas""")

    resp = input('      Ingrese una opción:')
    if resp == '1':
        print('\n' * 50)
        print("""
*****
*                                     *
*   Productos con mayores ventas y búsquedas   *
*                                     *
*****

1. Listar los 15 productos con mayores ventas
2. Listar los 20 productos con mayores búsquedas
""")
        resp = input('      Ingrese una opción (Ingrese cualquier tecla para volver
al menú:~')
        if resp == '1':
            print('-----~')
            print('          15 PRODUCTOS CON MAYORES VENTAS          ~')
            print('-----~')
            print('~ Num. Ventas      | Producto~')
            print('~-----~')
            for top in range(num_products - 1, num_products - 16, -1):
```

```

        espaciado = (tamaño_celda - len(str(frec_prod[top][0])))
        print(f"| {frec_prod[top][0]}{' '*espaciado}|
{lifystore_products[frec_prod[top][1] - 1][1]}")

    print('-----')
    input('Ingrese cualquier tecla para volver al menú')
    if resp == '2':
        print('-----')
        print('                20 PRODUCTOS CON MAYORES BÚSQUEDAS                ')
        print('-----')
        print('| Num. Búsquedas | Producto')
        print('|-----|')
        base = num_products - 21
        #Nos aseguramos que no se salga del tamaño de la lista
        if base < 0:
            base = 0
        for top in range(num_products - 1, base, -1):
            espaciado = (tamaño_celda - len(str(frec_search[top][0])))
            print(f"| {frec_search[top][0]}{' '* espaciado}|
{lifystore_products[frec_search[top][1] - 1][1]}")
            input('Ingrese cualquier letra para volver al menú')
        else:
            continue
    if resp == '2':
        print('\n' * 50)
        print(
            """
            *****
            *                                           *
            *  Productos con menores ventas y búquedas por categoría *
            *                                           *
            *****

            1. Categoría Procesadores
            2. Categoría Tarjetas de Video
            3. Categoría Tarjetas Madre
            4. Categoría Discos Duros
            5. Categoría Memorias USB
            6. Categoría Pantallas
            7. Categoría Bocinas
            8. Categoría Audífonos
            """)
        resp = input('        Ingrese una opción (Ingrese cualquier letra para volver
al menú):')

    if resp == '1':
        ### CATEGORIA PROCESADORES
        print()
        print()
        print("                CATEGORIA: PROCESADORES")
        print()
        print()
        cont = 0
        print('-----')
        print('                5 PRODUCTOS CON MENORES VENTAS                ')
        print('-----')
        print('| Num. Ventas      | Producto')
        print('|-----|')
        for prod in frec_prod:
            if cont >= 5:
                break
            categoria = lifystore_products[prod[1] - 1][3]
            if categoria == 'procesadores':
                espaciado = (tamaño_celda - len(str(prod[0])))

```

```

        print(f"| {prod[0]}{' '* espaciado}| {lifestore_products[prod[1]
- 1][1]}")
        cont += 1

    cont = 0
    print('-----')
    print('          20 PRODUCTOS CON MENORES BÚSQUEDAS          ')
    print('-----')
    print('| Num. Búsquedas | Producto')
    print('|-----|')
    for prod in frec_search:
        if cont >= 20:
            break
        categoria = lifestore_products[prod[1] - 1][3]
        if categoria == 'procesadores':
            espaciado = (tamano_celda - len(str(prod[0])))
            print(f"| {prod[0]}{' '* espaciado}| {lifestore_products[prod[1]
- 1][1]}")
            cont += 1
    input('Ingrese cualquier letra para volver al menú')
if resp == '2':
    ### CATEGORIA Tarjetas de video
    print()
    print()
    print("          CATEGORIA: TARJETAS DE VIDEO")
    print()
    print()
    cont = 0
    print('-----')
    print('          5 PRODUCTOS CON MENORES VENTAS          ')
    print('-----')
    print('| Num. Ventas | Producto')
    print('|-----|')
    for prod in frec_prod:
        if cont >= 5:
            break
        categoria = lifestore_products[prod[1] - 1][3]
        if categoria == 'tarjetas de video':
            espaciado = (tamano_celda - len(str(prod[0])))
            print(f"| {prod[0]}{' '* espaciado}| {lifestore_products[prod[1]
- 1][1]}")
            cont += 1

    cont = 0
    print('-----')
    print('          20 PRODUCTOS CON MENORES BÚSQUEDAS          ')
    print('-----')
    print('| Num. Búsquedas | Producto')
    print('|-----|')
    for prod in frec_search:
        if cont >= 20:
            break
        categoria = lifestore_products[prod[1] - 1][3]
        if categoria == 'tarjetas de video':
            espaciado = (tamano_celda - len(str(prod[0])))
            print(f"| {prod[0]}{' '* espaciado}| {lifestore_products[prod[1]
- 1][1]}")
            cont += 1

    input('Ingrese cualquier letra para volver al menú')
if resp == '3':
    ### CATEGORIA TARJETAS MADRE

```

```

print()
print()
print("
                                CATEGORIA: TARJETAS MADRE")
print()
print()
cont = 0
print('-----')
print('          5 PRODUCTOS CON MENORES VENTAS          ')
print('-----')
print('| Num. Ventas      | Producto')
print('-----')
for prod in frec_prod:
    if cont >= 5:
        break
    categoria = lifestore_products[prod[1] - 1][3]
    if categoria == 'tarjetas madre':
        espaciado = (tamano_celda - len(str(prod[0])))
        print(f"| {prod[0]}{' ' * espaciado} | {lifestore_products[prod[1]
- 1][1]}")
        cont += 1

cont = 0
print('-----')
print('          20 PRODUCTOS CON MENORES BÚSQUEDAS          ')
print('-----')
print('| Num. Búsquedas   | Producto')
print('-----')
for prod in frec_search:
    if cont >= 20:
        break
    categoria = lifestore_products[prod[1] - 1][3]
    if categoria == 'tarjetas madre':
        espaciado = (tamano_celda - len(str(prod[0])))
        print(f"| {prod[0]}{' ' * espaciado} | {lifestore_products[prod[1]
- 1][1]}")
        cont += 1
    input('Ingrese cualquier letra para volver al menú')
if resp == '4':
    ### CATEGORIA DISCOS DUROS

print()
print()
print("
                                CATEGORIA: DISCOS DUROS")
print()
print()
cont = 0
print('-----')
print('          5 PRODUCTOS CON MENORES VENTAS          ')
print('-----')
print('| Num. Ventas      | Producto')
print('-----')
for prod in frec_prod:
    if cont >= 5:
        break
    categoria = lifestore_products[prod[1] - 1][3]
    if categoria == 'discos duros':
        espaciado = (tamano_celda - len(str(prod[0])))
        print(f"| {prod[0]}{' ' * espaciado} | {lifestore_products[prod[1]
- 1][1]}")
        cont += 1

```

```

        cont = 0
        print('-----')
        print('                20 PRODUCTOS CON MENORES BÚSQUEDAS')
        print('-----')
        print('| Num. Búsquedas | Producto')
        print('-----')
        for prod in frec_search:
            if cont >= 20:
                break
            categoria = lifestore_products[prod[1] - 1][3]
            if categoria == 'discos duros':
                espaciado = (tamano_celda - len(str(prod[0])))
                print(f"| {prod[0]}{' '* espaciado}| {lifestore_products[prod[1]
- 1][1]}")
                cont += 1

        input('Ingrese cualquier letra para volver al menú')
    if resp == '5':
        ### CATEGORIA MEMORIAS USB

        print()
        print()
        print("                CATEGORIA: MEMORIAS USB")
        print()
        print()
        cont = 0
        print('-----')
        print('                5 PRODUCTOS CON MENORES VENTAS')
        print('-----')
        print('| Num. Ventas | Producto')
        print('-----')
        for prod in frec_prod:
            if cont >= 5:
                break
            categoria = lifestore_products[prod[1] - 1][3]
            if categoria == 'memorias usb':
                espaciado = (tamano_celda - len(str(prod[0])))
                print(f"| {prod[0]}{' '* espaciado}| {lifestore_products[prod[1]
- 1][1]}")
                cont += 1

        cont = 0
        print('-----')
        print('                20 PRODUCTOS CON MENORES BÚSQUEDAS')
        print('-----')
        print('| Num. Búsquedas | Producto')
        print('-----')
        for prod in frec_search:
            if cont >= 20:
                break
            categoria = lifestore_products[prod[1] - 1][3]
            if categoria == 'memorias usb':
                espaciado = (tamano_celda - len(str(prod[0])))
                print(f"| {prod[0]}{' '* espaciado}| {lifestore_products[prod[1]
- 1][1]}")
                cont += 1

        input('Ingrese cualquier letra para volver al menú')

```

```

if resp == '6':
    ### CATEGORIA PANTALLAS

    print()
    print()
    print("                CATEGORIA: PANTALLAS")
    print()
    print()
    cont = 0
    print('-----')
    print('                5 PRODUCTOS CON MENORES VENTAS')
    print('-----')
    print('| Num. Ventas | Producto')
    print('-----')
    for prod in frec_prod:
        if cont >= 5:
            break
        categoria = lifestore_products[prod[1] - 1][3]
        if categoria == 'pantallas':
            espaciado = (tamano_celda - len(str(prod[0])))
            print(f"| {prod[0]}{' '* espaciado}| {lifestore_products[prod[1]
- 1][1]}")
            cont += 1

    cont = 0
    print('-----')
    print('                20 PRODUCTOS CON MENORES BÚSQUEDAS')
    print('-----')
    print('| Num. Búsquedas | Producto')
    print('-----')
    for prod in frec_search:
        if cont >= 20:
            break
        categoria = lifestore_products[prod[1] - 1][3]
        if categoria == 'pantallas':
            espaciado = (tamano_celda - len(str(prod[0])))
            print(f"| {prod[0]}{' '* espaciado}| {lifestore_products[prod[1]
- 1][1]}")
            cont += 1

    input('Ingrese cualquier letra para volver al menú')
if resp == '7':
    ### CATEGORIA BOCINAS

    print()
    print()
    print("                CATEGORIA: BOCINAS")
    print()
    print()
    cont = 0
    print('-----')
    print('                5 PRODUCTOS CON MENORES VENTAS')
    print('-----')
    print('| Num. Ventas | Producto')
    print('-----')
    for prod in frec_prod:
        if cont >= 5:
            break
        categoria = lifestore_products[prod[1] - 1][3]
        if categoria == 'bocinas':

```

```

        espaciado = (tamano_celda - len(str(prod[0])))
        print(f"| {prod[0]}{' '* espaciado}| {lifestore_products[prod[1]
- 1][1]}")

        cont += 1

    cont = 0
    print('-----')
    print('          20 PRODUCTOS CON MENORES BÚSQUEDAS')
    print('-----')
    print('| Num. Búsquedas | Producto')
    print('|-----')
    for prod in frec_search:
        if cont >= 20:
            break
        categoria = lifestore_products[prod[1] - 1][3]
        if categoria == 'bocinas':
            espaciado = (tamano_celda - len(str(prod[0])))
            print(f"| {prod[0]}{' '* espaciado}| {lifestore_products[prod[1]
- 1][1]}")

            cont += 1

    input('Ingrese cualquier letra para volver al menú')
if resp == '8':
    ### CATEGORIA AUDIFONOS

    print()
    print()
    print("          CATEGORIA: AUDIFONOS")
    print()
    print()
    cont = 0
    print('-----')
    print('          5 PRODUCTOS CON MENORES VENTAS')
    print('-----')
    print('| Num. Ventas   | Producto')
    print('|-----')
    for prod in frec_prod:
        if cont >= 5:
            break
        categoria = lifestore_products[prod[1] - 1][3]
        if categoria == 'audifonos':
            espaciado = (tamano_celda - len(str(prod[0])))
            print(f"| {prod[0]}{' '* espaciado}| {lifestore_products[prod[1]
- 1][1]}")

            cont += 1

    cont = 0
    print('-----')
    print('          20 PRODUCTOS CON MENORES BÚSQUEDAS')
    print('-----')
    print('| Num. Búsquedas | Producto')
    print('|-----')
    for prod in frec_search:
        if cont >= 20:
            break
        categoria = lifestore_products[prod[1] - 1][3]
        if categoria == 'audifonos':
            espaciado = (tamano_celda - len(str(prod[0])))

```



```

        print(f"| {prod[0]}{' '*espaciado}| {lifestore_products[prod[1]
- 1][1]}")
        cont += 1
        input('Ingrese cualquier letra para volver al menú')
    if resp == '3':
        print('\n' * 50)
        print(
            """
            *****
            *                                     *
            *   Productos con mejores y peores reseñas   *
            *                                     *
            *****

            1. Listar los 10 productos con mejores reseñas
            2. Listar los 10 productos con peores reseñas
            """
        )
        resp = input('      Ingrese una opción (Ingrese cualquier letra para volver
al menú):')

        if resp == '1':
            print('-----')
            print('          10 PRODUCTOS CON MEJORES RESEÑAS          ')
            print('-----')
            print('| Promedio Reseña | Producto')
            print('|-----|')
            for top in range(num_products - 1, num_products - 11, -1):
                resenia = str(res_prod[top][0])
                product = lifestore_products[res_prod[top][1] - 1][1]
                espaciado = tamano_celda - len(resenia)
                print(f"| {resenia}{' '*espaciado}| {product}")
            input('Ingrese cualquier letra para volver al menú')
        if resp == '2':
            print('-----')
            print('          10 PRODUCTOS CON PEORES RESEÑAS          ')
            print('-----')
            print('| Promedio Reseña | Producto')
            print('|-----|')
            cont = 0
            for prod in res_prod:
                if prod[0] == 0.0:
                    continue
                if cont >= 10:
                    break
                resenia = str(prod[0])
                product = lifestore_products[prod[1] - 1][1]
                espaciado = tamano_celda - len(resenia)
                print(f"| {resenia}{' '*espaciado}| {product}")
                cont += 1
            input('Ingrese cualquier letra para volver al menú')
    if resp == '4':
        print('\n' * 50)
        print(
            """
            *****
            *                                     *
            *   Reporte de ingresos y ventas   *
            *                                     *
            *****

            1. Total de ingresos mensual
            2. Ventas promedio mensuales
            3. Total de ingresos anual
            4. Meses con más ventas al año
            """
        )

```

```

resp = input('      Ingrese una opción (Ingrese cualquier letra para volver al
menú):')

if resp == '1':
    print('-----')
    print('          Total de ingresos mensual          ')
    print('-----')
    print(' | Mes                | Prom. Ventas |')
    print(' |-----|')
    print(f' | Enero                | ${meses[0]} |')
    print(f' | Febrero              | ${meses[1]} |')
    print(f' | Marzo                | ${meses[2]} |')
    print(f' | Abril                | ${meses[3]} |')
    print(f' | Mayo                 | ${meses[4]} |')
    print(f' | Junio                | ${meses[5]} |')
    print(f' | Julio                | ${meses[6]} |')
    print(f' | Agosto               | ${meses[7]} |')
    print(f' | Septiembre           | ${meses[8]} |')
    print(f' | Octubre              | ${meses[9]} |')
    print(f' | Noviembre            | ${meses[10]} |')
    print(f' | Diciembre            | ${meses[11]} |')
    input('Ingrese cualquier letra para volver al menú')

if resp == '2':
    print('-----')
    print('          Promedio de ventas mensual          ')
    print('-----')
    print(' | Mes                | Prom. Ventas |')
    print(' |-----|')
    print(f' | Enero                | ${prom_mens[0]} |')
    print(f' | Febrero              | ${prom_mens[1]} |')
    print(f' | Marzo                | ${prom_mens[2]} |')
    print(f' | Abril                | ${prom_mens[3]} |')
    print(f' | Mayo                 | ${prom_mens[4]} |')
    print(f' | Junio                | ${prom_mens[5]} |')
    print(f' | Julio                | ${prom_mens[6]} |')
    print(f' | Agosto               | ${prom_mens[7]} |')
    print(f' | Septiembre           | ${prom_mens[8]} |')
    print(f' | Octubre              | ${prom_mens[9]} |')
    print(f' | Noviembre            | ${prom_mens[10]} |')
    print(f' | Diciembre            | ${prom_mens[11]} |')
    input('Ingrese cualquier letra para volver al menú')

if resp == '3':
    ingresos = 0
    for mes in meses:
        ingresos += mes
    print('-----')
    print('          Total de ingresos anual          ')
    print('-----')
    print(' | Año                | Ingresos          |')
    print(' |-----|')
    print(f' | 2020                | ${ingresos} |')
    input('Ingrese cualquier letra para volver al menú')

if resp == '4':
    ventas_mes.sort()
    print('-----')
    print('          Meses con más ventas al año          ')
    print('-----')
    for i in range(1,4):
        top = ventas_mes[-i]
        mes = top[1]
        if mes == 1:
            print('          -Enero')
        if mes == 2:
            print('          -Febrero')
        if mes == 3:
            print('          -Marzo')

```

```
if mes == 4:
    print('    -Abril')
if mes == 5:
    print('    -Mayo')
if mes == 6:
    print('    -Junio')
if mes == 7:
    print('    -Julio')
if mes == 8:
    print('    -Agosto')
if mes == 9:
    print('    -Septiembre')
if mes == 10:
    print('    -Octubre')
if mes == 11:
    print('    -Noviembre')
if mes == 12:
    print('    -Diciembre')
input('Ingrese cualquier letra para volver al menú')
```

Solución al problema

Como solución se propone retirar aquellos productos que tengan 0 búsquedas y 0 ventas.

En la categoría de procesadores no hay ningún producto que cumple este criterio ya que incluso el único producto que tiene 0 ventas ha sido buscado, por lo que al menos ha sido considerado como una opción.

En la categoría de tarjetas de video, los siguientes productos cumplen estas características:

- Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB 64-bit GDDR3, PCI Express 2.0
- Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming, 6GB 192-bit GDDR6, PCI Express 3.0
- Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile, 4GB 128-bit GDDR5, PCI Express 3.0 x16

En la categoría tarjetas madre:

- Tarjeta Madre AORUS ATX Z390 ELITE, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel
- Tarjeta Madre ASRock Z390 Phantom Gaming 4, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel
- Tarjeta Madre ASUS ATX ROG STRIX B550-F GAMING WI-FI, S-AM4, AMD B550, HDMI, max. 128GB DDR4 para AMD
- Tarjeta Madre Gigabyte micro ATX Z490M GAMING X (rev. 1.0), Intel Z490, HDMI, 128GB DDR4 para Intel

Categoría Discos Duros:

- SSD Addlink Technology S70, 512GB, PCI Express 3.0, M.2
- SSD para Servidor Supermicro SSD-DM128-SMCMVN1, 128GB, SATA III, mSATA, 6Gbit/s
- SSD para Servidor Lenovo Thinksystem S4510, 480GB, SATA III, 2.5", 7mm

Para la categoría Memorias USB el único producto satisface este criterio es: "Kit Memoria RAM Corsair Vengeance LPX DDR4, 2400MHz, 32GB, Non-ECC, CL16"

En la categoría pantallas, podría considerar retirar los siguientes artículos:

- Makena Smart TV LED 32S2 32", HD, Widescreen, Gris
- Samsung TV LED LH43QMREBGCXGO 43, 4K Ultra HD, Widescreen, Negro
- Samsung Smart TV LED UN70RU7100FXZX 70, 4K Ultra HD, Widescreen, Negro
- Makena Smart TV LED 40S2 40", Full HD, Widescreen, Negro

Categoría Bocinas:

- Lenovo Barra de Sonido, Alámbrico, 2.5W, USB, Negro
- Verbatim Bocina Portátil Mini, Bluetooth, Inalámbrico, 3W RMS, USB, Blanco
- Naceb Bocina Portátil NA-0301, Bluetooth, Inalámbrico, USB 2.0, Rojo

Categoría Audífonos:

- ASUS Audífonos Gamer ROG Theta 7.1, Alámbrico, USB C, Negro
- Acer Audífonos Gamer Galea 300, Alámbrico, 3.5mm, Negro
- Audífonos Gamer Balam Rush Orphix RGB 7.1, Alámbrico, USB, Negro
- Energy Sistem Audífonos con Micrófono Headphones 1, Bluetooth, Inalámbrico, Negro/Grafito

Conclusión

Hemos utilizado la programación en Python como herramienta en el análisis de datos de una tienda de productos. Este lenguaje ha resultado particularmente muy útil debido a su simplicidad y fácil aprendizaje, además de que cuenta con funciones que facilitan el análisis y la representación de información. Vimos además, que es muy complicado extraer información valiosa cuando los conjuntos de datos son listas enormes, por lo que procesarlas y presentarlas de una manera visualmente atractiva es muy beneficioso y permite llevar a cabo análisis más profundos, y por lo tanto solucionar problemas complejos.