



# **Emerging Technologies Institute**

INTRODUCCIÓN A PYTHON

## **REPORTE FINAL**

Autor:

Andrade Cortés Omar

13 de Febrero 2022

# Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Definición del Código</b>	<b>2</b>
2.1. Login de Acceso . . . . .	2
2.2. Productos más vendidos y productos rezagados . . . . .	3
2.2.1. Productos más vendidos y más buscados . . . . .	3
2.2.2. Productos menos vendidos y menos buscados . . . . .	8
2.3. Productos por reseña en el servicio . . . . .	25
2.3.1. Productos con mejores reseñas . . . . .	25
2.3.2. Productos con peores reseñas . . . . .	26
2.4. Total de Ingresos y ventas promedio mensuales, meses con más ventas del año . . .	27
2.4.1. Total de ingresos de la tabla . . . . .	27
2.4.2. Meses con mayores ventas, número de ventas promedio mensual e ingreso promedio mensual . . . . .	33
2.4.3. Meses con más ingresos del año . . . . .	35
<b>3. Solución al problema</b>	<b>38</b>
<b>4. Conclusión</b>	<b>39</b>

## 1. Introducción

*LifeStore* es una tienda virtual que maneja una amplia gama de artículos, sin embargo, la gerencia se ha percatado que tiene una considerable acumulación de inventario. Se nos ha solicitado que se haga un análisis de rotación de productos para identificar desde los productos más vendidos, los productos rezagados, los productos con mejores y peores reseñas de acuerdo a las evaluaciones del consumidor y como último dato, nos solicitan que les informemos del total de ingresos generado este año así como identificar los meses que tuvieron el mayor número de ventas y el mayor número de ingresos. Finalmente, se entregará una estrategia de productos a retirar del mercado, además de una sugerencia para reducir la acumulación de inventario.

**Liga de repositorio GitHub:** [https://github.com/omarac2299/myfirst\\_repo.git](https://github.com/omarac2299/myfirst_repo.git)

## 2. Definición del Código

```
[1]: """
    This is the LifeStore_SalesList data:

    lifestore_searches = [id_search, id product]
    lifestore_sales = [id_sale, id_product, score (from 1 to 5), date, refund (1 for_
    ↪true or 0 to false)]
    lifestore_products = [id_product, name, price, category, stock]
    """

    from lifestore_file import lifestore_searches, lifestore_sales,
    ↪lifestore_products
```

### 2.1. Login de Acceso

```
[2]: usuarioAccedio = False
    intentos = 0

    mensaje_ingreso = 'Bienvenido al sistema!\nIngresa tu usuario y contraseña a_
    ↪continuación'
    print(mensaje_ingreso)

    while not usuarioAccedio:
        usuario = input('Usuario: ')
        contras = input('Contrase;a: ')
        intentos += 1
        if usuario == 'Omar2299' and contras == '11052009':
            usuarioAccedio = True
            print('Hola de nuevo!')
        else:
            print(f'Tienes {3 - intentos} intentos restantes')
            if usuario == 'Omar2299':
                print('Te equivocaste en la contrase;a')
```

```
        else:
            print(f'El usuario: "{usuario}" no esta registrado')

    if intentos == 3:
        exit()

print('A continuación se despliega el reporte')
```

Bienvenido al sistema!

Ingresa tu usuario y contraseña a continuación

Usuario: Omar2299

Contraseña: 11052009

Hola de nuevo!

A continuación se despliega el reporte

## 2.2. Productos más vendidos y productos rezagados

### 2.2.1. Productos más vendidos y más buscados

```
[3]: """Por comodidad se generaron la siguientes variables, cuya información está
    ↳ relacionada al
    id del producto, su nombre y el precio, al menos para este primer análisis"""
id_and_prod = [[prod[0], prod[1]] for prod in lifestore_products]
id_prod_price = [[prod[0], prod[1], prod[2]] for prod in lifestore_products]
```

```
[4]: """Antes de empezar a ejecutar toda la secuencia de código, el administrador del
    ↳ proyecto decidió visualizar
    los tipos de datos en la tabla lifestore_sales, visualizando que el tipo de dato
    ↳ de la fecha estaba como
    string. La secuencia que se muestra a continuación, se encarga de separar la
    ↳ fecha por día, mes y año para luego
    convertir los datos a tipo entero y luego añadirlos a su registro
    ↳ correspondiente. Finalmente se remueve la
```

```

fecha en formato string para cada registro"""
new_sales = lifestore_sales.copy()
for venta in new_sales:
    dia = venta[3][:2]
    mes = venta[3][3:5]
    anio = venta[3][6:]

    if dia not in venta:
        venta.append(int(dia))

    if mes not in venta:
        venta.append(int(mes))

    else:
        venta.append(mes)

    if anio not in venta:
        venta.append(int(anio))

for venta in new_sales:
    venta.remove(venta[3])

c = 1
for v in new_sales:
    if c < 5:
        print(v)

    c += 1

# Ahora new_sales = [[id_sale, id_product, score, refund, day, month, year]]

```

[1, 1, 5, 0, 24, 7, 2020]

[2, 1, 5, 0, 27, 7, 2020]

[3, 2, 5, 0, 24, 2, 2020]

[4, 2, 5, 0, 22, 5, 2020]

[5]: *"""Como este reporte solo es efectivo para el año 2020, se verifica que si hay  
 ↪ existencia de datos atípicos, para  
 así no considerarlos en las siguientes ejecuciones de código. La secuencia a\_  
 ↪ continuación, muestra los años que*

```
se encontraron dentro de la tabla"""
anios_dist = []
for venta in new_sales:
    if venta[-1] not in anios_dist:
        anios_dist.append(venta[-1])
print(anios_dist)
```

[2020, 2019, 2002]

```
[6]: """Para los 10 productos con MAYORES BÚSQUEDAS, de la lista lifestore_searches,
    →tomamos el segundo elemento de
    cada registro que corresponde al id del producto para meterlo en la lista
    →busqueda, es decir, hay números que
    se repiten; después generamos el dic_of_searches donde contamos cuantas veces se
    →repite el id dentro de la lista
    busqueda para después aplicarle la función sorted que nos brinda una lista de
    →tuplas que podemos acomodar de
    mayor a menor con ayuda de la función lambda, le ponemos un contador condicional
    →para que solo se nos devuelvan
    los primeros diez productos"""
busqueda = []

for item in lifestore_searches:
    busqueda.append(item[1])

dict_of_searches = {item_id:busqueda.count(item_id) for item_id in busqueda}
sorted_by_value = sorted(dict_of_searches.items(), key = lambda v: v[1], reverse=
    →= True)

conta = 1
for i in sorted_by_value:
    if conta < 11:
```

```
for pr in id_and_prod:
    if i[0] == pr[0]:
        print(f'{pr[1]} con id: {i[0]} tuvo {i[1]} busquedas\n')
    conta += 1
```

SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm con id: 54 tuvo 263 busquedas

SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm con id: 57 tuvo 107 busquedas

Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD con id: 29 tuvo 60 busquedas

Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth con id: 3 tuvo 55 busquedas

Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire con id: 4 tuvo 41 busquedas

Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul con id: 85 tuvo 35 busquedas

TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro con id: 67 tuvo 32 busquedas

Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake) con id: 7 tuvo 31 busquedas

Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake) con id: 5 tuvo 30 busquedas

SSD XPG SX8200 Pro, 256GB, PCI Express, M.2 con id: 47 tuvo 30 busquedas

```
[7]: """Para los 5 productos con MAYORES VENTAS, el proceso es análogo al código
    ↪ anterior, sin embargo, las considera
    ciones cambian. Ahora trabajamos con la lista new_sales, pero consideramos
    ↪ solamente consideramos al año 2020,
    también debemos tener en cuenta que de las ventas que se realizaron, los
    ↪ productos devueltos no deben considerarse
    como una venta, por lo que se contabilizan aparte, se crean diccionarios para
    ↪ contabilizar por id como en la celda
    anterior y además se crea un diccionario de ventas netas donde se hace la
    ↪ sustracción correspondiente entre
    los diccionarios de ventas y devoluciones"""

ventas = []
devssss = []

for item in new_sales:
    if item[-1] == anios_dist[0]: # Con este condicional solo estamos teneiendo en
    ↪ cuenta al año 2020
        if item[3] == 0:
            ventas.append(item[1])
        else:
            devssss.append(item[1])

dict_of_sales = {item_id:ventas.count(item_id) for item_id in ventas}
dict_of_devssss = {d:devssss.count(d) for d in devssss}

dict_of_vn = {key: dict_of_sales[key] - dict_of_devssss.get(key, 0)
              for key in dict_of_sales.keys()}

sorted_by_value1 = sorted(dict_of_vn.items(), key = lambda v: v[1], reverse =
    ↪ True)
```



```

conta = 1
for i in sorted_by_value1:
    if conta < 6:
        for pr in id_and_prod:
            if i[0] == pr[0]:
                print(f'{pr[1]} con id: {i[0]} tuvo {i[1]} ventas\n')
            conta += 1

```

SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm con id: 54 tuvo 47 ventas

Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con  
Disipador Wraith Stealth con id: 3 tuvo 42 ventas

Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na.  
Generación - Coffee Lake) con id: 5 tuvo 20 ventas

Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB  
DDR4 para AMD con id: 42 tuvo 18 ventas

SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm con id: 57 tuvo 15 ventas

### 2.2.2. Productos menos vendidos y menos buscados

```

[8]: """Aquí lo que se busca generar es un diccionario cuyas llaves sean las
    →categorías y en los valores sea una lista
    con los productos correspondientes a las mismas, esto nos ayudara para las
    →siguientes secuencias"""

id_categoria = [[prod[0], prod[3]] for prod in lifestore_products]
prods_clasificados = {}
for par in id_categoria:
    cat = par[1]
    if cat not in prods_clasificados.keys():

```

```

        prods_clasificados[cat] = []
        prods_clasificados[cat].append(par[0])
print(prods_clasificados)

```

```

{'procesadores': [1, 2, 3, 4, 5, 6, 7, 8, 9], 'tarjetas de video': [10, 11, 12,
13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28], 'tarjetas
madre': [29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45,
46], 'discos duros': [47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59],
'memorias usb': [60, 61], 'pantallas': [62, 63, 64, 65, 66, 67, 68, 69, 70, 71,
72, 73], 'bocinas': [74, 75, 76, 77, 78, 79, 80, 81, 82, 83], 'audifonos': [84,
85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96]}

```

```

[9]: # 10 Productos por categoría con MENORES BUSQUEDAS
      """Con ayuda del diccionario: prods_clasificados y el dic_of_searches ejecutamos
      ↳secuencias for e if donde lo
      que se busca hacer es un cruce entre los valores del primer diccionario y las
      ↳llaves del segundo para así
      por categoria poder imprimir el nombre del producto, su id y la cantidad de
      ↳busquedas, de no encontrar uno de
      los valores del primer diccionario en las llaves del segundo contemplamos que
      ↳eso significa 0 busquedas
      por lo que le generamos su propia secuencia if especial. Nota: Para la secuencia
      ↳if j<5 se decidio que cualquier
      numero menor a 5 hace que un producto sea poco buscado."""
      for k,v in prods_clasificados.items():
          print(f'Para la categoría de {k} se reportaron las siguientes busquedas de
          ↳productos: \n')
          for i in v:
              for h,j in dict_of_searches.items():
                  if i == h:
                      if j < 5:
                          for p in id_and_prod:

```

```
        if p[0]==i:
            print(f'{p[1]} con id: {i} tuvo {j} busqueda(s)\n')
    if i not in dict_of_searches.keys():
        for p in id_and_prod:
            if p[0]==i:
                print(f'{p[1]} con id: {i} tuvo {0} busqueda(s)\n')
```

Para la categoría de procesadores se reportaron las siguientes busquedas de productos:

Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake) con id: 9 tuvo 1 busqueda(s)

Para la categoría de tarjetas de video se reportaron las siguientes busquedas de productos:

MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0 con id: 10 tuvo 1 busqueda(s)

Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix, 4GB 128-bit GDDR5, PCI Express 3.0 con id: 13 tuvo 2 busqueda(s)

Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB 64-bit GDDR3, PCI Express 2.0 con id: 14 tuvo 0 busqueda(s)

Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0 con id: 15 tuvo 4 busqueda(s)

Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming, 6GB 192-bit GDDR6, PCI Express 3.0 con id: 16 tuvo 0 busqueda(s)

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

3.0 con id: 17 tuvo 3 busqueda(s)

Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile, 4GB 128-bit GDDR5, PCI Express 3.0 x16 con id: 19 tuvo 0 busqueda(s)

Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC, 8 GB 256 bit GDDR6, PCI Express x16 3.0 con id: 20 tuvo 0 busqueda(s)

Tarjeta de Video MSI Radeon X1550, 128MB 64 bit GDDR2, PCI Express x16 con id: 23 tuvo 0 busqueda(s)

Tarjeta de Video PNY NVIDIA GeForce RTX 2080, 8GB 256-bit GDDR6, PCI Express 3.0 con id: 24 tuvo 0 busqueda(s)

Tarjeta de Video VisionTek AMD Radeon HD5450, 2GB GDDR3, PCI Express x16 con id: 27 tuvo 1 busqueda(s)

Para la categoría de tarjetas madre se reportaron las siguientes busquedas de productos:

Tarjeta Madre AORUS ATX Z390 ELITE, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel con id: 30 tuvo 0 busqueda(s)

Tarjeta Madre ASRock Z390 Phantom Gaming 4, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel con id: 32 tuvo 0 busqueda(s)

Tarjeta Madre ASUS ATX PRIME Z390-A, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel con id: 33 tuvo 0 busqueda(s)

Tarjeta Madre ASUS ATX ROG STRIX B550-F GAMING WI-FI, S-AM4, AMD B550, HDMI, max. 128GB DDR4 para AMD con id: 34 tuvo 0 busqueda(s)

Tarjeta Madre Gigabyte micro ATX Z390 M GAMING, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel con id: 35 tuvo 1 busqueda(s)

Tarjeta Madre Gigabyte micro ATX Z490M GAMING X (rev. 1.0), Intel Z490, HDMI, 128GB DDR4 para Intel con id: 36 tuvo 0 busqueda(s)

Tarjeta Madre ASRock ATX Z490 STEEL LEGEND, S-1200, Intel Z490, HDMI, 128GB DDR4 para Intel con id: 37 tuvo 0 busqueda(s)

Tarjeta Madre Gigabyte Micro ATX H310M DS2 2.0, S-1151, Intel H310, 32GB DDR4 para Intel con id: 38 tuvo 0 busqueda(s)

ASUS T. Madre uATX M4A88T-M, S-AM3, DDR3 para Phenom II/Athlon II/Sempron 100 con id: 39 tuvo 3 busqueda(s)

Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM, S-1151, Intel H370, HDMI, 64GB DDR4 para Intel con id: 41 tuvo 0 busqueda(s)

Tarjeta Madre ASUS ATX ROG STRIX Z390-E GAMING, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel con id: 43 tuvo 0 busqueda(s)

Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel con id: 45 tuvo 1 busqueda(s)

Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel con id: 46 tuvo 4 busqueda(s)

Para la categoría de discos duros se reportaron las siguientes busquedas de productos:

SSD Addlink Technology S70, 512GB, PCI Express 3.0, M.2 con id: 53 tuvo 0 busqueda(s)

SSD para Servidor Supermicro SSD-DM128-SMCMVN1, 128GB, SATA III, mSATA, 6Gbit/s  
con id: 55 tuvo 0 busqueda(s)

SSD para Servidor Lenovo Thinksystem S4500, 480GB, SATA III, 3.5'', 7mm con id:  
56 tuvo 2 busqueda(s)

SSD para Servidor Lenovo Thinksystem S4510, 480GB, SATA III, 2.5'', 7mm con id:  
58 tuvo 0 busqueda(s)

SSD Samsung 860 EVO, 1TB, SATA III, M.2 con id: 59 tuvo 1 busqueda(s)

Para la categoría de memorias usb se reportaron las siguientes busquedas de  
productos:

Kit Memoria RAM Corsair Dominator Platinum DDR4, 3200MHz, 16GB (2x 8GB), Non-  
ECC, CL16, XMP con id: 60 tuvo 0 busqueda(s)

Kit Memoria RAM Corsair Vengeance LPX DDR4, 2400MHz, 32GB, Non-ECC, CL16 con id:  
61 tuvo 0 busqueda(s)

Para la categoría de pantallas se reportaron las siguientes busquedas de  
productos:

Makena Smart TV LED 32S2 32'', HD, Widescreen, Gris con id: 62 tuvo 0  
busqueda(s)

Seiki TV LED SC-39HS950N 38.5, HD, Widescreen, Negro con id: 63 tuvo 4  
busqueda(s)

Samsung TV LED LH43QMREBGXG0 43, 4K Ultra HD, Widescreen, Negro con id: 64 tuvo  
0 busqueda(s)

Samsung Smart TV LED UN70RU7100FXZX 70, 4K Ultra HD, Widescreen, Negro con id: 65 tuvo 0 busqueda(s)

Makena Smart TV LED 40S2 40'', Full HD, Widescreen, Negro con id: 68 tuvo 0 busqueda(s)

Hisense Smart TV LED 40H5500F 39.5, Full HD, Widescreen, Negro con id: 69 tuvo 0 busqueda(s)

Samsung Smart TV LED 43, Full HD, Widescreen, Negro con id: 70 tuvo 1 busqueda(s)

Samsung Smart TV LED UN32J4290AF 32, HD, Widescreen, Negro con id: 71 tuvo 0 busqueda(s)

Hisense Smart TV LED 50H8F 49.5, 4K Ultra HD, Widescreen, Negro con id: 72 tuvo 0 busqueda(s)

Samsung Smart TV LED UN55TU7000FXZX 55, 4K Ultra HD, Widescreen, Negro/Gris con id: 73 tuvo 4 busqueda(s)

Para la categoría de bocinas se reportaron las siguientes busquedas de productos:

Lenovo Barra de Sonido, Alámbrico, 2.5W, USB, Negro con id: 75 tuvo 0 busqueda(s)

Acteck Bocina con Subwoofer AXF-290, Bluetooth, Inalámbrico, 2.1, 18W RMS, 180W PMP0, USB, Negro con id: 76 tuvo 2 busqueda(s)

Verbatim Bocina Portátil Mini, Bluetooth, Inalámbrico, 3W RMS, USB, Blanco con

id: 77 tuvo 0 búsqueda(s)

Ghia Bocina Portátil BX300, Bluetooth, Inalámbrico, 40W RMS, USB, Rojo -

Resistente al Agua con id: 78 tuvo 0 búsqueda(s)

Naceb Bocina Portátil NA-0301, Bluetooth, Inalámbrico, USB 2.0, Rojo con id: 79

tuvo 0 búsqueda(s)

Ghia Bocina Portátil BX800, Bluetooth, Inalámbrico, 2.1 Canales, 31W, USB, Negro

con id: 80 tuvo 1 búsqueda(s)

Ghia Bocina Portátil BX900, Bluetooth, Inalámbrico, 2.1 Canales, 34W, USB, Negro

- Resistente al Agua con id: 81 tuvo 0 búsqueda(s)

Ghia Bocina Portátil BX400, Bluetooth, Inalámbrico, 8W RMS, USB, Negro con id:

82 tuvo 0 búsqueda(s)

Ghia Bocina Portátil BX500, Bluetooth, Inalámbrico, 10W RMS, USB, Gris con id:

83 tuvo 0 búsqueda(s)

Para la categoría de audífonos se reportaron las siguientes búsquedas de productos:

ASUS Audífonos Gamer ROG Theta 7.1, Alámbrico, USB C, Negro con id: 86 tuvo 0

búsqueda(s)

Acer Audífonos Gamer Galea 300, Alámbrico, 3.5mm, Negro con id: 87 tuvo 0

búsqueda(s)

Audífonos Gamer Balam Rush Orphix RGB 7.1, Alámbrico, USB, Negro con id: 88 tuvo

0 búsqueda(s)



Energy Sistem Audífonos con Micrófono Headphones 1, Bluetooth, Inalámbrico, Negro/Grafito con id: 90 tuvo 0 búsqueda(s)

Genius GHP-400S Audífonos, Alámbrico, 1.5 Metros, Rosa con id: 91 tuvo 2 búsqueda(s)

Getttech Audífonos con Micrófono Sonority, Alámbrico, 1.2 Metros, 3.5mm, Negro/Rosa con id: 92 tuvo 0 búsqueda(s)

Ginga Audífonos con Micrófono GI18ADJ01BT-R0, Bluetooth, Alámbrico/Inalámbrico, 3.5mm, Rojo con id: 93 tuvo 1 búsqueda(s)

Iogear Audífonos Gamer GHG601, Alámbrico, 1.2 Metros, 3.5mm, Negro con id: 95 tuvo 3 búsqueda(s)

Klip Xtreme Audífonos Blast, Bluetooth, Inalámbrico, Negro/Verde con id: 96 tuvo 0 búsqueda(s)

```
[10]: # 5 Productos por categoría con MENORES VENTAS
      """Para este proceso la diferencia es que se utiliza el dict_of_vn (diccionario
      ↪ de ventas netas)"""
      for k,v in prods_clasificados.items():
          print(f'Para la categoría de {k} se reportaron las siguientes ventas: \n')
          for i in v:
              for h,j in dict_of_vn.items():
                  if i == h:
                      if j < 5:
                          for p in id_and_prod:
                              if p[0]==i:
                                  print(f'{p[1]} con id: {i} tuvo {j} venta(s)\n')
              if i not in dict_of_vn.keys():
```

```
for p in id_and_prod:
    if p[0]==i:
        print(f'{p[1]} con id: {i} tuvo {0} venta(s)\n')
```

Para la categoría de procesadores se reportaron las siguientes ventas:

Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache con id: 1  
tuvo 2 venta(s)

Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na.  
Generación Coffee Lake) con id: 6 tuvo 3 venta(s)

Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na.  
Generación - Coffee Lake) con id: 8 tuvo 4 venta(s)

Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va.  
Generación - Coffee Lake) con id: 9 tuvo 0 venta(s)

Para la categoría de tarjetas de video se reportaron las siguientes ventas:

MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0 con id: 10 tuvo 1  
venta(s)

Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Express 3.0 con  
id: 11 tuvo 3 venta(s)

Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix, 4GB 128-bit GDDR5, PCI  
Express 3.0 con id: 13 tuvo 1 venta(s)

Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB 64-bit GDDR3, PCI Express 2.0  
con id: 14 tuvo 0 venta(s)

Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0 con id: 15 tuvo 0 venta(s)

Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming, 6GB 192-bit GDDR6, PCI Express 3.0 con id: 16 tuvo 0 venta(s)

Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0 con id: 17 tuvo 0 venta(s)

Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile, 4GB 128-bit GDDR5, PCI Express 3.0 x16 con id: 19 tuvo 0 venta(s)

Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC, 8 GB 256 bit GDDR6, PCI Express x16 3.0 con id: 20 tuvo 0 venta(s)

Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC, 8GB 128-bit GDDR6, PCI Express 4.0 con id: 21 tuvo 2 venta(s)

Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC, 4GB 128-bit GDDR5, PCI Express x16 3.0 con id: 22 tuvo 1 venta(s)

Tarjeta de Video MSI Radeon X1550, 128MB 64 bit GDDR2, PCI Express x16 con id: 23 tuvo 0 venta(s)

Tarjeta de Video PNY NVIDIA GeForce RTX 2080, 8GB 256-bit GDDR6, PCI Express 3.0 con id: 24 tuvo 0 venta(s)

Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming, 8GB 128-bit GDDR6, PCI Express 4.0 con id: 25 tuvo 2 venta(s)

Tarjeta de Video VisionTek AMD Radeon HD 5450, 1GB DDR3, PCI Express x16 2.1 con id: 26 tuvo 0 venta(s)

Tarjeta de Video VisionTek AMD Radeon HD5450, 2GB GDDR3, PCI Express x16 con id: 27 tuvo 0 venta(s)

Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti, 6GB 192-bit GDDR6, PCI Express x16 3.0 con id: 28 tuvo 1 venta(s)

Para la categoría de tarjetas madre se reportaron las siguientes ventas:

Tarjeta Madre AORUS ATX Z390 ELITE, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel con id: 30 tuvo 0 venta(s)

Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD con id: 31 tuvo 0 venta(s)

Tarjeta Madre ASRock Z390 Phantom Gaming 4, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel con id: 32 tuvo 0 venta(s)

Tarjeta Madre ASUS ATX PRIME Z390-A, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel con id: 33 tuvo 2 venta(s)

Tarjeta Madre ASUS ATX ROG STRIX B550-F GAMING WI-FI, S-AM4, AMD B550, HDMI, max. 128GB DDR4 para AMD con id: 34 tuvo 0 venta(s)

Tarjeta Madre Gigabyte micro ATX Z390 M GAMING, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel con id: 35 tuvo 0 venta(s)

Tarjeta Madre Gigabyte micro ATX Z490M GAMING X (rev. 1.0), Intel Z490, HDMI, 128GB DDR4 para Intel con id: 36 tuvo 0 venta(s)

Tarjeta Madre ASRock ATX Z490 STEEL LEGEND, S-1200, Intel Z490, HDMI, 128GB DDR4 para Intel con id: 37 tuvo 0 venta(s)

Tarjeta Madre Gigabyte Micro ATX H310M DS2 2.0, S-1151, Intel H310, 32GB DDR4 para Intel con id: 38 tuvo 0 venta(s)

ASUS T. Madre uATX M4A88T-M, S-AM3, DDR3 para Phenom II/Athlon II/Sempron 100 con id: 39 tuvo 0 venta(s)

Tarjeta Madre Gigabyte XL-ATX TRX40 Designare, S-sTRX4, AMD TRX40, 256GB DDR4 para AMD con id: 40 tuvo 1 venta(s)

Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM, S-1151, Intel H370, HDMI, 64GB DDR4 para Intel con id: 41 tuvo 0 venta(s)

Tarjeta Madre ASUS ATX ROG STRIX Z390-E GAMING, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel con id: 43 tuvo 0 venta(s)

Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel con id: 45 tuvo 0 venta(s)

Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel con id: 46 tuvo 0 venta(s)

Para la categoría de discos duros se reportaron las siguientes ventas:

Kit SSD Kingston KC600, 1TB, SATA III, 2.5, 7mm con id: 49 tuvo 3 venta(s)

SSD Crucial MX500, 1TB, SATA III, M.2 con id: 50 tuvo 1 venta(s)

SSD Kingston UV500, 480GB, SATA III, mSATA con id: 51 tuvo 3 venta(s)

SSD Western Digital WD Blue 3D NAND, 2TB, M.2 con id: 52 tuvo 2 venta(s)

SSD Addlink Technology S70, 512GB, PCI Express 3.0, M.2 con id: 53 tuvo 0 venta(s)

SSD para Servidor Supermicro SSD-DM128-SMCMVN1, 128GB, SATA III, mSATA, 6Gbit/s con id: 55 tuvo 0 venta(s)

SSD para Servidor Lenovo Thinksystem S4500, 480GB, SATA III, 3.5'', 7mm con id: 56 tuvo 0 venta(s)

SSD para Servidor Lenovo Thinksystem S4510, 480GB, SATA III, 2.5'', 7mm con id: 58 tuvo 0 venta(s)

SSD Samsung 860 EVO, 1TB, SATA III, M.2 con id: 59 tuvo 0 venta(s)

Para la categoría de memorias usb se reportaron las siguientes ventas:

Kit Memoria RAM Corsair Dominator Platinum DDR4, 3200MHz, 16GB (2x 8GB), Non-ECC, CL16, XMP con id: 60 tuvo 1 venta(s)

Kit Memoria RAM Corsair Vengeance LPX DDR4, 2400MHz, 32GB, Non-ECC, CL16 con id: 61 tuvo 0 venta(s)

Para la categoría de pantallas se reportaron las siguientes ventas:

Makena Smart TV LED 32S2 32'', HD, Widescreen, Gris con id: 62 tuvo 0 venta(s)

Seiki TV LED SC-39HS950N 38.5, HD, Widescreen, Negro con id: 63 tuvo 0 venta(s)

Samsung TV LED LH43QMREBGCXG0 43, 4K Ultra HD, Widescreen, Negro con id: 64 tuvo 0 venta(s)

Samsung Smart TV LED UN70RU7100FXZX 70, 4K Ultra HD, Widescreen, Negro con id:

65 tuvo 0 venta(s)

TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Widescreen, Negro con id: 66 tuvo 1 venta(s)

TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro con id: 67 tuvo 1 venta(s)

Makena Smart TV LED 40S2 40'', Full HD, Widescreen, Negro con id: 68 tuvo 0 venta(s)

Hisense Smart TV LED 40H5500F 39.5, Full HD, Widescreen, Negro con id: 69 tuvo 0 venta(s)

Samsung Smart TV LED 43, Full HD, Widescreen, Negro con id: 70 tuvo 0 venta(s)

Samsung Smart TV LED UN32J4290AF 32, HD, Widescreen, Negro con id: 71 tuvo 0 venta(s)

Hisense Smart TV LED 50H8F 49.5, 4K Ultra HD, Widescreen, Negro con id: 72 tuvo 0 venta(s)

Samsung Smart TV LED UN55TU7000FXZX 55, 4K Ultra HD, Widescreen, Negro/Gris con id: 73 tuvo 0 venta(s)

Para la categoría de bocinas se reportaron las siguientes ventas:

Logitech Bocinas para Computadora con Subwoofer G560, Bluetooth, Inalámbrico, 2.1, 120W RMS, USB, negro con id: 74 tuvo 2 venta(s)

Lenovo Barra de Sonido, Alámbrico, 2.5W, USB, Negro con id: 75 tuvo 0 venta(s)

Acteck Bocina con Subwoofer AXF-290, Bluetooth, Inalámbrico, 2.1, 18W RMS, 180W PMP0, USB, Negro con id: 76 tuvo 0 venta(s)

Verbatim Bocina Portátil Mini, Bluetooth, Inalámbrico, 3W RMS, USB, Blanco con id: 77 tuvo 0 venta(s)

Ghia Bocina Portátil BX300, Bluetooth, Inalámbrico, 40W RMS, USB, Rojo - Resistente al Agua con id: 78 tuvo 0 venta(s)

Naceb Bocina Portátil NA-0301, Bluetooth, Inalámbrico, USB 2.0, Rojo con id: 79 tuvo 0 venta(s)

Ghia Bocina Portátil BX800, Bluetooth, Inalámbrico, 2.1 Canales, 31W, USB, Negro con id: 80 tuvo 0 venta(s)

Ghia Bocina Portátil BX900, Bluetooth, Inalámbrico, 2.1 Canales, 34W, USB, Negro - Resistente al Agua con id: 81 tuvo 0 venta(s)

Ghia Bocina Portátil BX400, Bluetooth, Inalámbrico, 8W RMS, USB, Negro con id: 82 tuvo 0 venta(s)

Ghia Bocina Portátil BX500, Bluetooth, Inalámbrico, 10W RMS, USB, Gris con id: 83 tuvo 0 venta(s)

Para la categoría de audifonos se reportaron las siguientes ventas:

Logitech Audífonos Gamer G332, Alámbrico, 2 Metros, 3.5mm, Negro/Rojo con id: 84 tuvo 1 venta(s)

Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul con id: 85 tuvo 2 venta(s)



ASUS Audífonos Gamer ROG Theta 7.1, Alámbrico, USB C, Negro con id: 86 tuvo 0 venta(s)

Acer Audífonos Gamer Galea 300, Alámbrico, 3.5mm, Negro con id: 87 tuvo 0 venta(s)

Audífonos Gamer Balam Rush Orphix RGB 7.1, Alámbrico, USB, Negro con id: 88 tuvo 0 venta(s)

Cougar Audífonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro. con id: 89 tuvo 1 venta(s)

Energy Sistem Audífonos con Micrófono Headphones 1, Bluetooth, Inalámbrico, Negro/Grafito con id: 90 tuvo 0 venta(s)

Genius GHP-400S Audífonos, Alámbrico, 1.5 Metros, Rosa con id: 91 tuvo 0 venta(s)

Getttech Audífonos con Micrófono Sonority, Alámbrico, 1.2 Metros, 3.5mm, Negro/Rosa con id: 92 tuvo 0 venta(s)

Ginga Audífonos con Micrófono GI18ADJ01BT-R0, Bluetooth, Alámbrico/Inalámbrico, 3.5mm, Rojo con id: 93 tuvo 0 venta(s)

HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro, Inalámbrico, USB, 3.5mm, Negro con id: 94 tuvo 1 venta(s)

Iogear Audífonos Gamer GHG601, Alámbrico, 1.2 Metros, 3.5mm, Negro con id: 95 tuvo 0 venta(s)

Klip Xtreme Audífonos Blast, Bluetooth, Inalámbrico, Negro/Verde con id: 96 tuvo 0 venta(s)

## 2.3. Productos por reseña en el servicio

### 2.3.1. Productos con mejores reseñas

```
[11]: resena_prod = {}
for venta in lifestore_sales:
    if venta[1] not in resena_prod.keys():
        resena_prod[venta[1]] = []
    resena_prod[venta[1]].append(venta[2])
#print(resena_prod)
for key in resena_prod:
    resena_prod[key] = sum(resena_prod[key])/len(resena_prod[key])
print("Los 5 productos con las mejores reseñas fueron:\n")
mejor_resena = sorted(resena_prod.items(), key = lambda v: v[1], reverse = True)
#print(mejor_resena)
conta = 1
for i in mejor_resena:
    if i[1] < 5:
        if conta < 6:
            for pr in id_and_prod:
                if i[0] == pr[0]:
                    print(f'{pr[1]} con id: {i[0]} y puntaje de {i[1]} \n')
            conta += 1
```

Los 5 productos con las mejores reseñas fueron:

SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm con id: 57 y puntaje de 4.8666666666666666

Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth con id: 3 y puntaje de 4.809523809523809

Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6, PCI Express x16 3.0 con id: 12 y puntaje de 4.777777777777778

SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm con id: 54 y puntaje de 4.72

Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake) con id: 5 y puntaje de 4.7

Se observó que los productos con puntaje de 5 realmente tenían pocas ventas, por lo que posicionarlos para este ranking era generar información sesgada así que se optó por colocar los que tenían más ventas, es decir, más evaluaciones.

### 2.3.2. Productos con peores reseñas

```
[12]: resena_prod1 = {}  
  
for venta in lifestore_sales:  
    if venta[1] not in resena_prod1.keys():  
        resena_prod1[venta[1]] = []  
        resena_prod1[venta[1]].append(venta[2])  
  
for key in resena_prod1:  
    resena_prod1[key] = sum(resena_prod1[key])/len(resena_prod1[key])  
print("Los 5 productos con las peores reseñas fueron:\n")  
mejor_resena = sorted(resena_prod1.items(), key = lambda v: v[1], reverse =   
    ↪False)  
conta = 1  
for i in mejor_resena:  
    if conta < 6:  
        for pr in id_and_prod:  
            if i[0] == pr[0]:  
                print(f'{pr[1]} con id: {i[0]} y puntaje de {i[1]} \n')  
        conta += 1
```

Los 5 productos con las peores reseñas fueron:

Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0 con id: 17 y puntaje de 1.0

Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel con id: 45 y puntaje de 1.0

Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD con id: 31 y puntaje de 1.8333333333333333

Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel con id: 46 y puntaje de 2.0

Cougar Audífonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro. con id: 89 y puntaje de 3.0

## 2.4. Total de Ingresos y ventas promedio mensuales, meses con más ventas del año

### 2.4.1. Total de ingresos de la tabla

```
[13]: """Esta celda de código brindará los ingresos brutos de la tabla_  
→correspondientes al año 2020 a partir clasificar  
por id los ingresos generados"""  
dic_ingresos = {}  
for venta in new_sales:  
    if venta[-1] == anios_dist[0]:  
        if venta[1] not in dic_ingresos.keys():  
            dic_ingresos[venta[1]] = []  
        for clv in id_prod_price:  
            if venta[1] == clv[0]:  
                dic_ingresos[venta[1]].append(clv[2])
```

```
for key in dic_ingresos:
    dic_ingresos[key] = sum(dic_ingresos[key])
print(f'El total de ingresos brutos de la tienda fue de: ${sum(dic_ingresos.
    values())}')
```

El total de ingresos brutos de la tienda fue de: \$755709

```
[14]: """Esta celda brindará el monto total de las devoluciones en el año 2020, de
    igual forma que en la celda anterior,
    el diccionario clasificara por id las sumas de dinero devueltas. Al final se
    hace la diferencia entre los ingresos
    brutos y las devoluciones para obtener los ingresos netos"""
dic_dev = {}
for venta in new_sales:
    if venta[-1] == anios_dist[0]: # Con esta condición garantizamos 2020
        if venta[3] == 1:
            if venta[1] not in dic_dev.keys():
                dic_dev[venta[1]] = []
            for clv in id_prod_price:
                if venta[1] == clv[0]:
                    dic_dev[venta[1]].append(clv[2])
for key in dic_dev:
    dic_dev[key] = sum(dic_dev[key])
print(f'El total de las devoluciones hacia la tienda durante 2020 fue de:
    ${sum(dic_dev.values())}\n quedando así, un ingreso neto de:
    ${sum(dic_ingresos.values())-sum(dic_dev.values())}')
```

El total de las devoluciones hacia la tienda durante 2020 fue de: \$18052  
quedando así, un ingreso neto de: \$737657

```
[15]: """Como elemento adicional decidimos generar un diccionario adicional para
    obtener por id los ingresos netos
```

```
y ordenarlos del producto que generó los mayores ingresos, al que menos lo  
→hizo"""  
dic_ingresos_netos = {key: dic_ingresos[key] - dic_dev.get(key, 0)  
                      for key in dic_ingresos.keys()}  
sorted_by_ing = sorted(dic_ingresos_netos.items(), key = lambda v: v[1], reverse_  
→= True)  
for i in sorted_by_ing:  
    for pr in id_and_prod:  
        if i[0] == pr[0]:  
            print(f'{pr[1]} con id: {i[0]} generó ingresos por: ${i[1]}\n')
```

Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con  
Disipador Wraith Stealth con id: 3 generó ingresos por: \$129738

Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na.  
Generación Coffee Lake) con id: 7 generó ingresos por: \$59913

Tarjeta de Video ASUS NVIDIA GeForce GTX 1660 SUPER EVO OC, 6GB 192-bit GDDR6,  
PCI Express x16 3.0 con id: 12 generó ingresos por: \$59571

Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith  
Stealth con id: 2 generó ingresos por: \$50508

Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na.  
Generación - Coffee Lake) con id: 5 generó ingresos por: \$35580

Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na.  
Generación Coffee Lake) con id: 6 generó ingresos por: \$35427

Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB  
DDR4 para AMD con id: 29 generó ingresos por: \$32487

Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD con id: 42 generó ingresos por: \$32022

Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire con id: 4 generó ingresos por: \$28717

SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2 con id: 48 generó ingresos por: \$23031

Tarjeta de Video ASUS AMD Radeon RX 570, 4GB 256-bit GDDR5, PCI Express 3.0 con id: 11 generó ingresos por: \$22197

Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake) con id: 8 generó ingresos por: \$21596

Tarjeta Madre Gigabyte XL-ATX TRX40 Designare, S-sTRX4, AMD TRX40, 256GB DDR4 para AMD con id: 40 generó ingresos por: \$17439

Tarjeta Madre MSI ATX B450 TOMAHAWK MAX, S-AM4, AMD B450, 64GB DDR4 para AMD con id: 44 generó ingresos por: \$16554

SSD Adata Ultimate SU800, 256GB, SATA III, 2.5'', 7mm con id: 57 generó ingresos por: \$13335

SSD XPG SX8200 Pro, 256GB, PCI Express, M.2 con id: 47 generó ingresos por: \$13299

SSD Kingston A400, 120GB, SATA III, 2.5'', 7mm con id: 54 generó ingresos por: \$12432

SSD Western Digital WD Blue 3D NAND, 2TB, M.2 con id: 52 generó ingresos por: \$11318

Tarjeta de Video Sapphire AMD Pulse Radeon RX 5500 XT Gaming, 8GB 128-bit GDDR6, PCI Express 4.0 con id: 25 generó ingresos por: \$11058

Tarjeta de Video Gigabyte NVIDIA GeForce GT 1030, 2GB 64-bit GDDR5, PCI Express x16 3.0 con id: 18 generó ingresos por: \$10995

Tarjeta de Video MSI AMD Mech Radeon RX 5500 XT MECH Gaming OC, 8GB 128-bit GDDR6, PCI Express 4.0 con id: 21 generó ingresos por: \$10318

Tarjeta de Video Zotac NVIDIA GeForce GTX 1660 Ti, 6GB 192-bit GDDR6, PCI Express x16 3.0 con id: 28 generó ingresos por: \$9579

Kit SSD Kingston KC600, 1TB, SATA III, 2.5, 7mm con id: 49 generó ingresos por: \$9417

Tarjeta Madre ASUS ATX PRIME Z390-A, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel con id: 33 generó ingresos por: \$8538

Logitech Bocinas para Computadora con Subwoofer G560, Bluetooth, Inalámbrico, 2.1, 120W RMS, USB, negro con id: 74 generó ingresos por: \$8478

TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Widescreen, Negro con id: 66 generó ingresos por: \$8049

SSD Kingston UV500, 480GB, SATA III, mSATA con id: 51 generó ingresos por: \$7197

Tarjeta Madre AORUS micro ATX B450 AORUS M (rev. 1.0), S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD con id: 31 generó ingresos por: \$6687

Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache con id: 1 generó ingresos por: \$6038



Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul con id: 85 generó ingresos por: \$4318

Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix, 4GB 128-bit GDDR5, PCI Express 3.0 con id: 13 generó ingresos por: \$3989

Tarjeta de Video MSI NVIDIA GeForce GTX 1050 Ti OC, 4GB 128-bit GDDR5, PCI Express x16 3.0 con id: 22 generó ingresos por: \$3429

TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro con id: 67 generó ingresos por: \$3229

SSD Crucial MX500, 1TB, SATA III, M.2 con id: 50 generó ingresos por: \$2949

HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro, Inalámbrico, USB, 3.5mm, Negro con id: 94 generó ingresos por: \$2869

Kit Memoria RAM Corsair Dominator Platinum DDR4, 3200MHz, 16GB (2x 8GB), Non-ECC, CL16, XMP con id: 60 generó ingresos por: \$2519

Logitech Audífonos Gamer G332, Alámbrico, 2 Metros, 3.5mm, Negro/Rojo con id: 84 generó ingresos por: \$1089

MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0 con id: 10 generó ingresos por: \$889

Cougar Audífonos Gamer Phontum Essential, Alámbrico, 1.9 Metros, 3.5mm, Negro. con id: 89 generó ingresos por: \$859

Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0 con id: 17 generó ingresos por: \$0

Tarjeta Madre ASRock ATX H110 Pro BTC+, S-1151, Intel H110, 32GB DDR4, para Intel con id: 45 generó ingresos por: \$0

Tarjeta Madre Gigabyte micro ATX GA-H110M-DS2, S-1151, Intel H110, 32GB DDR4 para Intel con id: 46 generó ingresos por: \$0

#### 2.4.2. Meses con mayores ventas, número de ventas promedio mensual e ingreso promedio mensual

```
[16]: """Por unica excepción, si hubo que generar un diccionario calendario
→manualmente, ya que no se poseía el nombre
correspondiente a cada número de mes, después generamos los diccionarios de
→ventas y devoluciones ahora teniendo
como llave el mes y como valores el numero ventas/devoluciones"""
meses = {1:'Enero', 2:'Febrero', 3:'Marzo', 4:'Abril', 5:'Mayo',
        6:'Junio', 7:'Julio', 8:'Agosto', 9:'Septiembre', 10:'Octubre', 11:
        →'Noviembre',12:'Diciembre'}

list_month_ndev = []
list_month_dev = []
for venta in new_sales:
    if venta[-1] == anios_dist[0]:
        if venta[3] == 0:
            list_month_ndev.append(venta[-2])
        else:
            list_month_dev.append(venta[-2])
dic_meses_ventas = {mes : list_month_ndev.count(mes) for mes in list_month_ndev}
dic_meses_dev = {mes : list_month_dev.count(mes) for mes in list_month_dev}
sorted_mv = sorted(dic_meses_ventas.items(), key = lambda v: v[1], reverse =
→True)
sorted_md = sorted(dic_meses_dev.items(), key = lambda v: v[1], reverse = True)
```

```
for i in sorted_mv:
    for key in meses.keys():
        if i[0] == key:
            print(f'El mes de {meses[key]} tuvo {i[1]} ventas')
for i in sorted_md:
    for key in meses.keys():
        if i[0] == key:
            print(f'\nEl mes de {meses[key]} tuvo {i[1]} devoluciones')
print(f'A lo largo del año 2020 se reportaron {sum(dic_meses_ventas.values())}
→ventas y {sum(dic_meses_dev.values())} devoluciones')
print(f'Obteniendo así un aproximado de {round(sum(dic_meses_ventas.values())/
→12,0)} ventas por mes')
print(f'El ingreso promedio mensual durante 2020 fue de
→${round((sum(dic_ingresos.values())-sum(dic_dev.values())/12,2)}')
```

El mes de Abril tuvo 74 ventas

El mes de Enero tuvo 52 ventas

El mes de Marzo tuvo 49 ventas

El mes de Febrero tuvo 40 ventas

El mes de Mayo tuvo 33 ventas

El mes de Julio tuvo 11 ventas

El mes de Junio tuvo 11 ventas

El mes de Agosto tuvo 3 ventas

El mes de Mayo tuvo 2 devoluciones

El mes de Marzo tuvo 2 devoluciones

El mes de Septiembre tuvo 1 devoluciones

El mes de Enero tuvo 1 devoluciones

El mes de Abril tuvo 1 devoluciones

El mes de Febrero tuvo 1 devoluciones

A lo largo del año 2020 se reportaron 273 ventas y 8 devoluciones

Obteniendo así un aproximado de 23.0 ventas por mes

El ingreso promedio mensual durante 2020 fue de \$61471.42

### 2.4.3. Meses con más ingresos del año

```
[17]: """Agregaremos un campo más a la tabla new_sales que incluire el monto generado_  
→por transacción"""  
for v in new_sales:  
    for p in id_prod_price:  
        if v[1] == p[0]:  
            v.append(p[2])  
  
"""A partir de aquí se generara un diccionario que tenga por llaves los meses y_  
→como valores la suma de ingresos  
brutos en el año 2020"""  
dic_ingresos_m = {}  
for venta in new_sales:  
    if venta[-2] == anios_dist[0]:  
        if venta[-3] not in dic_ingresos_m.keys():  
            dic_ingresos_m[venta[-3]] = []  
        for key in dic_ingresos_m.keys():  
            if key == venta[-3]:  
                dic_ingresos_m[venta[-3]].append(venta[-1])  
for key in dic_ingresos_m.keys():  
    dic_ingresos_m[key] = sum(dic_ingresos_m[key])  
  
"""De igual forma que como lo hicimos en diccionarios anteriores, usamos la_  
→función sorted con apoyo nuestra función
```

```

lambda para presentar de los meses donde se reportaron más ingresos a los que
→menos tuvieron. Finalmente, se pre
senta el total de ingresos brutos"""
s_ing_m = sorted(dic_ingresos_m.items(), key = lambda v:v[1], reverse = True)
for i in s_ing_m:
    for mes in meses.keys():
        if i[0] == mes:
            print(f'El mes de {meses[mes]} tuvo ingresos brutos por ${i[1]}')
print(f'\n Ingresos brutos por un total de ${sum(dic_ingresos_m.values())}\n')

"""Generamos un diccionario de devoluciones para presentar los meses con
→devoluciones, es análogo al
dic_ingresos_m. Finalmente se presenta el total de devoluciones"""
dic_dev_m = {}
for venta in new_sales:
    if venta[-2] == anios_dist[0]:
        if venta[3] == 1:
            if venta[-3] not in dic_dev_m.keys():
                dic_dev_m[venta[-3]] = []
            for key in dic_dev_m.keys():
                if key == venta[-3]:
                    dic_dev_m[venta[-3]].append(venta[-1])
for key in dic_dev_m.keys():
    dic_dev_m[key] = sum(dic_dev_m[key])
s_dev_m = sorted(dic_dev_m.items(), key = lambda v: v[1], reverse = True)
for i in s_dev_m:
    for mes in meses.keys():
        if i[0] == mes:
            print(f'El mes de {meses[mes]} tuvo devoluciones por ${i[1]}')
print(f'\n Devoluciones por un total de ${sum(dic_dev_m.values())}\n')

```

```
"""Ahora generamos el diccionario de ingresos netos por mes haciendo la
→diferencia por mes entre los dos
diccionarios anteriores y obtenemos realmente los meses con mayores y menores
→ventas"""

dic_ingnet_mes = {key: dic_ingresos_m[key] - dic_dev_m.get(key, 0)
                  for key in dic_ingresos_m.keys()}
s_ingnet_m = sorted(dic_ingnet_mes.items(), key = lambda v: v[1], reverse = True)
for i in s_ingnet_m:
    for mes in meses.keys():
        if i[0] == mes:
            print(f'El mes de {meses[mes]} tuvo ingresos netos por ${i[1]}')
print(f'\n Ingresos netos por un total de ${sum(dic_ingnet_mes.values())}')
```

El mes de Abril tuvo ingresos brutos por \$193295  
El mes de Marzo tuvo ingresos brutos por \$164729  
El mes de Enero tuvo ingresos brutos por \$120237  
El mes de Febrero tuvo ingresos brutos por \$110139  
El mes de Mayo tuvo ingresos brutos por \$96135  
El mes de Junio tuvo ingresos brutos por \$36949  
El mes de Julio tuvo ingresos brutos por \$26949  
El mes de Septiembre tuvo ingresos brutos por \$4199  
El mes de Agosto tuvo ingresos brutos por \$3077

Ingresos brutos por un total de \$755709

El mes de Mayo tuvo devoluciones por \$4458  
El mes de Septiembre tuvo devoluciones por \$4199  
El mes de Febrero tuvo devoluciones por \$2869  
El mes de Enero tuvo devoluciones por \$2499  
El mes de Abril tuvo devoluciones por \$2229  
El mes de Marzo tuvo devoluciones por \$1798

Devoluciones por un total de \$18052

El mes de Abril tuvo ingresos netos por \$191066

El mes de Marzo tuvo ingresos netos por \$162931

El mes de Enero tuvo ingresos netos por \$117738

El mes de Febrero tuvo ingresos netos por \$107270

El mes de Mayo tuvo ingresos netos por \$91677

El mes de Junio tuvo ingresos netos por \$36949

El mes de Julio tuvo ingresos netos por \$26949

El mes de Agosto tuvo ingresos netos por \$3077

El mes de Septiembre tuvo ingresos netos por \$0

Ingresos netos por un total de \$737657

### 3. Solución al problema

La solución es retirar del inventario aquellos productos que generaron ingresos por menos \$1000 ya que solo ocupan espacio dentro del mismo y no generan ganancias, de esta forma podemos considerarlos incluso una pérdida para la empresa de no venderse pronto, por lo que se sugieren 2 posibles opciones: venderlos al costo para evitar perdidas o venderlos al 75 % del valor inicial para así tener una perdida general del 25 % antes de que se vuelvan obsoletos y sean prácticamente no objetos de venta. Podemos señalar incluso que nuestra información coincide con los productos cuyo ranking fue bajo.

A partir del mes de Mayo, se puede observar que las ventas dentro de la tienda fueron disminuyendo hasta el mes de Agosto, a partir de ahí, el ultimo cuatrimestre del 2020 se observaron 0 ventas; para solucionar este problema, se sugiere hacer campañas de promoción y descuento a partir de los meses de Mayo de 2021 que incentiven a los compradores a no dejar de buscar productos en *LifeStore*.

## 4. Conclusión

Para generar este reporte se utilizaron secuencias de código if, for, else; diccionarios y muchas listas. Por lo que se concluye que se puso en práctica el contenido del curso número 1. La solución para LifeStore ya fue ofrecida con anterioridad.