

LifeStore

Reporte de Resultados del Análisis de Búsquedas y Ventas

Reporta:
Ing. Gerardo Steve Munguía Williams
Agencia de Consultoría EmTech

Diciembre 2021

Índice

Introducción	3
Comentarios sobre el código	4
Resultados	5
Conclusiones	8
Anexos	9

Introducción

Derivado de la solicitud hecha por la Gerencia de Ventas de LifeStore a nuestra consultora, se realizó un análisis de los productos ofrecidos, tomando en cuenta:

- Los productos más y menos vendidos
- Los productos más y menos buscados
- Los productos con mejores y peores reseñas
- Los productos que reportan más y menos ingresos

Lo anterior con el fin de definir qué productos conviene conservar y cuáles se sugiere que sean retirados.

Para el análisis se utilizó una base de datos con información de los productos ofrecidos, las ventas realizadas en 2020 y las búsquedas reportadas en la tienda en línea.

Los resultados se obtuvieron con un programa escrito en Python con el que el usuario puede interactuar y hacer peticiones de la información de cada producto. Al final del presente reporte se sugiere una estrategia de productos a retirar.

Comentarios sobre el código

El programa, que puede encontrarse en <https://github.com/gstevemw/Emtech/blob/main/PROYECTO-01-MUNGUIA-GERARDO.py> (véase Anexo 1), comienza con una solicitud de usuario y contraseña. Se ha definido como usuario a la Gerencia de Ventas de la empresa (Gerencia), por ser quien solicitó la información, y de contraseña fue establecido el nombre de la empresa (LifeStore). Se cuenta con tres intentos para iniciar sesión; de lo contrario, el programa termina.

Posterior a introducir los datos de login, se muestra información por categoría sobre los productos menos vendidos y menos buscados (véase Anexo 2), así como las ventas por mes de cada categoría (véase Anexo 3).

Finalmente, el usuario puede interactuar con el programa, eligiendo sobre qué quiere hacer su siguiente consulta: Ventas, Búsquedas, Reseñas o Ingresos. El programa permite consultar, por mes o en general, cuáles fueron los productos:

- Con mayores y menores ventas
- Con más y menos búsquedas
- Con mejores y peores reseñas
- Que reportaron mayores y menores ingresos

Resultados

La Gerencia solicitó específicamente los siguientes datos:

Los 5 productos con mayores ventas:

Producto: SSD Kingston A400, 120GB, SATA III, 2.5", 7mm

Ventas del periodo: 50

Producto: Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth

Ventas del periodo: 42

Producto: Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)

Ventas del periodo: 20

Producto: Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD

Ventas del periodo: 18

Producto: SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm

Ventas del periodo: 15

Los 10 productos con mayores búsquedas:

Producto: SSD Kingston A400, 120GB, SATA III, 2.5", 7mm

Búsquedas: 263

Producto: SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm

Búsquedas: 107

Producto: Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD

Búsquedas: 60

Producto: Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth

Búsquedas: 55

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

Búsquedas: 41

Producto: Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul

Búsquedas: 35

Producto: TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro

Búsquedas: 32

Producto: Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)

Búsquedas: 31

Producto: SSD XPG SX8200 Pro, 256GB, PCI Express, M.2

Búsquedas: 30

Producto: Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)

Búsquedas: 30

Productos con menores ventas y búsquedas por categoría: véase Anexo 2.

Los 5 productos con mejores reseñas:

Producto: Logitech Audífonos Gamer G635 7.1, Alámbrico, 1.5 Metros, 3.5mm, Negro/Azul

Promedio de reseñas: 5.0

Producto: Logitech Audífonos Gamer G332, Alámbrico, 2 Metros, 3.5mm, Negro/Rojo

Promedio de reseñas: 5.0

Producto: TV Monitor LED 24TL520S-PU 24, HD, Widescreen, HDMI, Negro

Promedio de reseñas: 5.0

Producto: TCL Smart TV LED 55S425 54.6, 4K Ultra HD, Widescreen, Negro

Promedio de reseñas: 5.0

Producto: Kit Memoria RAM Corsair Dominator Platinum DDR4, 3200MHz, 16GB (2x 8GB), Non-ECC, CL16, XMP

Promedio de reseñas: 5.0

Los 5 productos con peores reseñas:

Producto: Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)

Promedio de reseñas: 0

Producto: Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB 64-bit GDDR3, PCI Express 2.0

Promedio de reseñas: 0

Producto: Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0

Promedio de reseñas: 0

Producto: Tarjeta de Video EVGA NVIDIA GeForce RTX 2060 SC ULTRA Gaming, 6GB 192-bit GDDR6, PCI Express 3.0

Promedio de reseñas: 0

Producto: Tarjeta de Video Gigabyte NVIDIA GeForce GTX 1650 OC Low Profile, 4GB 128-bit GDDR5, PCI Express 3.0 x16

Promedio de reseñas: 0

Meses con mayores ventas e ingresos:

Abril: En el periodo hubieron ingresos por \$191066

Marzo: En el periodo hubieron ingresos por \$162931

Enero: En el periodo hubieron ingresos por \$117738

Conclusiones

Luego del análisis de resultados, la sugerencia principal es enfocarse en las ventas de procesadores, tarjetas madre, discos duros y tarjetas de video, pues fueron los productos más vendidos y los que más ganancias representan.

Las memorias USB, bocinas y audífonos tuvieron pocas ventas, además de representar menores ganancias por ser de bajo costo. A pesar de ocupar poco espacio en bodega, no se sugiere continuar con la venta de estos productos.

En el caso de las pantallas, a pesar de que tuvieron muchas búsquedas, las ventas concretadas fueron prácticamente nulas. La sugerencia es no continuar con su venta, pues el costo de mantenerlas en bodega es elevado.

Se entregó un código interactivo para que la Gerencia tenga mejores herramientas para la toma de decisiones. Derivado del análisis de ventas, búsquedas, reseñas e ingresos, se hicieron sugerencias sobre los productos que se sugiere conservar y los que se sugiere retirar.

En adelante se podría profundizar el análisis si se añadiera información sobre las búsquedas, pues al carecer de la fecha en la que se realizaron no se puede explotar su relevancia tanto como se pudiera.

Anexo 1

Código del programa en Python

```
"""
```

```
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]
```

```
"""
```

```
meses=['01','02','03','04','05','06','07','08','09','10','11','12']
```

```
nombre_meses=['enero','febrero','marzo','abril','mayo','junio','julio','agosto','septiembre','octubre','noviembre','diciembre']
```

```
#INICIO
```

```
#Lo primero que hace el usuario es el login; en este caso será usuario Gerencia y contraseña LifeStore
```

```
usuario = "Gerencia"
```

```
password = "LifeStore"
```

```
intento = 0
```

```
while intento<3:
```

```
user=input("Ingrese un nombre de usuario: ")
```

```
pswd=input("Ingrese una contraseña: ")
```

```
if user==usuario:
```

```
if password==pswd:
```

```
print(f"Inicio de sesión exitoso\nBienvenid@, {user}")
```

```
break
```

```
else:
```

```
print("Contraseña incorrecta")
```

```
intento+=1
```

```
continue
```

```
else:
```

```
print("Usuario no encontrado")
```

```
intento+=1
```

```
continue
```

```
if intento==3:
```

```
print("Número de intentos agotado")
```

```
exit()
```

```
#Una vez iniciada sesión, el programa despliega un resumen de cada categoría de producto:
```

```

#Primero encuentro las categorías existentes
categorias = []
for producto in lifestore_products:
    if producto[3] not in categorias:
        categorias.append(producto[3])
#Después encuentro la información solicitada por categoría
for categoria in categorias:
    print(f"\nPara {categoria}")
#Se obtienen los productos correspondientes a cada categoría
productos_categoria=[]
for producto in lifestore_products:
    if producto[3]==categoria:
        productos_categoria.append(producto)
#Se muestran los productos con menores ventas
print("Mostrando los 5 productos con menos ventas")
sublista=[]
for producto in productos_categoria:
    cantidad_productos=0
    for venta in lifestore_sales:
        if venta[1]==producto[0]:
            cantidad_productos+=1
    sublista.append([cantidad_productos,producto[0]])
sublista.sort()
i=0
for producto in sublista:
    if i<5:
        print(lifestore_products[producto[1]-1][1])
        i+=1
    else:
        break
#Se muestran los productos con menores búsquedas
print("Mostrando los 10 productos con menos búsquedas")
sublista=[]
for producto in productos_categoria:
    cantidad_productos=0
    for busqueda in lifestore_searches:
        if busqueda[1]==producto[0]:
            cantidad_productos+=1
    sublista.append([cantidad_productos,producto[0]])
sublista.sort()
i=0
for producto in sublista:
    if i<10:
        print(lifestore_products[producto[1]-1][1])

```

11

```
i+=1
else:
break

#Y muestra información por mes por cada categoría
print("\nMostrando resultados por mes de ventas de producto: ")

for mes in meses:
num_mes=int(mes)-1
print(f"Ventas del mes de {nombre_meses[num_mes]}")
ventas_mes=[]
for venta in lifestore_sales:
fecha=venta[3]
#Para no mezclar peras con manzanas, considero sólo el año 2020, dado que sólo
# encontré dos datos de ventas que no son de ese año:
#for venta in lifestore_sales:
# fecha=venta[3]
# if fecha[-4:]!="2020":
# print(venta)
if fecha[3:5]==mes and fecha[-4:]=="2020":
ventas_mes.append(venta)
for categoria in categorias:
cantidad=0
for venta in ventas_mes:
id_producto=venta[1]
producto=lifestore_products[id_producto-1]
if categoria==producto[3]:
cantidad+=1
print(f"Se vendieron {cantidad} {categoria}")

#Ahora el usuario puede elegir una serie de opciones de consulta:
salir = False
while not salir:
opcion = input("¿Qué desea consultar?\nVentas: 1\nBúsquedas: 2\nReseñas: 3\nIngresos: 4\n")
if opcion=='1':
#El siguiente es el código para trabajar con ventas
print("Ha seleccionado: Ventas")
#El usuario puede elegir entre visualizar las ventas de un mes o totales
opcion_mes = input("Si desea consultar ventas de un mes específico escriba 'mes'\n")
if opcion_mes=="mes":
mes=input("¿Qué mes desea revisar? Enero = 01, Febrero = 02, etc.\n")
while mes not in meses:
print("Dato incorrecto. Utilice el formato indicado: ")
mes=input("¿Qué mes desea revisar? Enero = 01, Febrero = 02, etc.\n")
```

```

num_mes=int(mes)-1
print(f"Ha seleccionado el mes de {nombre_meses[num_mes]}")
ventas_mes=[]
for venta in lifestore_sales:
    fecha=venta[3]
    #Para no mezclar peras con manzanas, considero sólo el año 2020, dado que sólo
    # encontré dos datos de ventas que no son de ese año:
    #for venta in lifestore_sales:
    # fecha=venta[3]
    # if fecha[-4:]!="2020":
    # print(venta)
    if fecha[3:5]==mes and fecha[-4:]=="2020":
        ventas_mes.append(venta)
    ventas=ventas_mes
else:
    print("Se desplegará información de ventas totales")
    ventas=lifestore_sales
    #Se obtiene la cantidad de ventas por producto
    cantidad_ventas=[]
    for producto in lifestore_products:
        id_producto = producto[0]
        cantidad=0
        for venta in ventas:
            if id_producto==venta[1]:
                cantidad+=1
        cantidad_ventas.append([cantidad, id_producto])
    #Se despliegan las ventas totales del periodo
    suma=0
    for venta in cantidad_ventas:
        suma+=venta[0]
    print(f"En el periodo hubieron {suma} ventas")
    #Después, divide por categorías
    for categoria in categorias:
        suma=0
        for venta in cantidad_ventas:
            producto=lifestore_products[venta[1]-1]
            if producto[3]==categoria:
                suma+=venta[0]
        print(f"Se vendieron {suma} {categoria}")
    #Después, las cantidades de venta por producto del periodo que el usuario haya elegido
    print("Mostrando ventas por producto:")
    #Y el usuario elige entre una serie de opciones de visualización por cantidad de ventas
    opcion = input("¿Cómo desea acomodar los productos?\nProductos más vendidos: 1\
nProductos menos vendidos: 2\nSin acomodar: 3\n")
    if opcion=='1':

```

```

print("Ha seleccionado: Productos más vendidos")
#Si son los más vendidos, los productos se acomodan de más a menos ventas
cantidad_ventas.sort(reverse=True)
elif opcion=='2':
print("Ha seleccionado: Productos menos vendidos")
#Si son los menos vendidos, se acomodan de menos a más
cantidad_ventas.sort()
elif opcion=='3':
print("Ha seleccionado: Sin acomodar")
else:
print("Opción no encontrada")
continue
#El usuario elige de cuántos productos quiere hacer su top
cuantos=int(input("¿Cuántos productos desea ver?\n"))
if cuantos>len(cantidad_ventas):
cuantos=len(cantidad_ventas)
#Y finalmente, se muestra lo que el usuario eligió
i=0
for producto in cantidad_ventas:
if i<cuantos:
id_producto=producto[1]-1
print(f"Producto: {lifestore_products[id_producto][1]}")
print(f"Ventas del periodo: {producto[0]}")
i+=1
else:
break
pregunta=input("Si desea continuar escriba 's'; de lo contrario, escriba cualquier letra\n")
if pregunta!="s":
salir=True
elif opcion=='2':
#El siguiente es el código para trabajar con búsquedas
print("Ha seleccionado: Búsquedas")
#Se consiguen las búsquedas por producto
cantidad_búsquedas=[]
for producto in lifestore_products:
id_producto = producto[0]
cantidad=0
for busqueda in lifestore_searches:
if id_producto==busqueda[1]:
cantidad+=1
cantidad_búsquedas.append([cantidad, id_producto])
#Después, divide por categorías
for categoria in categorias:
suma=0
for busqueda in cantidad_búsquedas:

```

```

producto=lifestore_products[busqueda[1]-1]
if producto[3]==categoria:
    suma+=busqueda[0]
print(f"Se buscaron {suma} {categoria}")
#Después el usuario elige entre una serie de opciones de visualización por cantidad de
búsquedas
opcion = input("¿Cómo desea acomodar los productos?\nProductos más buscados: 1\
nProductos menos buscados: 2\nSin acomodar: 3\n")
if opcion=='1':
    print("Ha seleccionado: Productos más buscados")
    #Si son los más buscados, los productos se acomodan de más a menos búsquedas
    cantidad_busquedas.sort(reverse=True)
elif opcion=='2':
    print("Ha seleccionado: Productos menos buscados")
    #Si son los menos buscados, se acomodan de menos a más
    cantidad_busquedas.sort()
elif opcion=='3':
    print("Ha seleccionado: Sin acomodar")
else:
    print("Opción no encontrada")
    continue
#El usuario elige de cuántos productos quiere hacer su top
cuantos=int(input("¿Cuántos productos desea ver?\n"))
if cuantos>len(cantidad_busquedas):
    cuantos=len(cantidad_busquedas)
#Y finalmente, se muestra lo que el usuario eligió
i=0
for producto in cantidad_busquedas:
    if i<cuantos:
        id_producto=producto[1]-1
        print(f"Producto: {lifestore_products[id_producto][1]}")
        print(f"Búsquedas: {producto[0]}")
        i+=1
    else:
        break
pregunta=input("Si desea continuar escriba 's'; de lo contrario, escriba cualquier letra\n")
if pregunta!="s":
    salir=True
elif opcion=='3':
    print("Ha seleccionado: Reseñas")
    #El usuario puede elegir entre visualizar las reseñas de un mes o totales
    opcion_mes = input("Si desea consultar reseñas de un mes específico escriba 'mes'\n")
    if opcion_mes=="mes":
        mes=input("¿Qué mes desea revisar? Enero = 01, Febrero = 02, etc.\n")
        while mes not in meses:

```

```

print("Dato incorrecto. Utilice el formato indicado: ")
mes=input("¿Qué mes desea revisar? Enero = 01, Febrero = 02, etc.\n")
num_mes=int(mes)-1
print(f"Ha seleccionado el mes de {nombre_meses[num_mes]}")
ventas_mes=[]
for venta in lifestore_sales:
    fecha=venta[3]
    #Para no mezclar peras con manzanas, considero sólo el año 2020, dado que sólo
    # encontré dos datos de ventas que no son de ese año:
    #for venta in lifestore_sales:
    # fecha=venta[3]
    # if fecha[-4:]!="2020":
    # print(venta)
    if fecha[3:5]==mes and fecha[-4:]=="2020":
        ventas_mes.append(venta)
ventas=ventas_mes
else:
    print("Se desplegará información de reseñas totales")
    ventas=lifestore_sales
    #Se consiguen reseñas por producto del periodo que el usuario haya elegido
    promedio_score=[]
    for producto in lifestore_products:
        id_producto = producto[0]
        suma=0
        cantidad=0
        for venta in ventas:
            if id_producto==venta[1]:
                cantidad+=1
                suma+=venta[2]
            if cantidad == 0:
                promedio = 0
            else:
                promedio = suma/cantidad
        promedio_score.append([promedio, id_producto])
    #Después el usuario elige entre una serie de opciones de visualización por promedio de
    reseñas
    opcion = input("¿Cómo desea acomodar los productos?\nProductos con mejores reseñas: 1\
nProductos con peores reseñas: 2\nSin acomodar: 3\n")
    if opcion=='1':
        print("Ha seleccionado: Productos con mejores reseñas")
        #Si son los mejor reseñados, los productos se acomodan del mayor al menor promedio
        promedio_score.sort(reverse=True)
    elif opcion=='2':
        print("Ha seleccionado: Productos con peores reseñas")
        #Si son los peor reseñados, se acomodan de menor a mayor

```

```

promedio_score.sort()
elif opcion=='3':
    print("Ha seleccionado: Sin acomodar")
else:
    print("Opción no encontrada")
    continue
#El usuario elige de cuántos productos quiere hacer su top
cuantos=int(input("¿Cuántos productos desea ver?\n"))
if cuantos>len(promedio_score):
    cuantos=len(promedio_score)
#Y finalmente, se muestra lo que el usuario eligió
i=0
for producto in promedio_score:
    if i<cuantos:
        id_producto=producto[1]-1
        print(f"Producto: {lifestore_products[id_producto][1]}")
        print(f"Promedio de reseñas: {producto[0]}")
        i+=1
    else:
        break
pregunta=input("Si desea continuar escriba 's'; de lo contrario, escriba cualquier letra\n")
if pregunta!="s":
    salir=True
elif opcion=='4':
    print("Ha seleccionado: Ingresos")
    #El usuario puede elegir entre visualizar los ingresos de un mes o totales
    opcion_mes = input("Si desea consultar los ingresos de un mes específico escriba 'mes'\n")
    if opcion_mes=="mes":
        mes=input("¿Qué mes desea revisar? Enero = 01, Febrero = 02, etc.\n")
        while mes not in meses:
            print("Dato incorrecto. Utilice el formato indicado: ")
            mes=input("¿Qué mes desea revisar? Enero = 01, Febrero = 02, etc.\n")
            num_mes=int(mes)-1
            print(f"Ha seleccionado el mes de {nombre_meses[num_mes]}")
            ventas_mes=[]
            for venta in lifestore_sales:
                fecha=venta[3]
                #Para no mezclar peras con manzanas, considero sólo el año 2020, dado que sólo
                # encontré dos datos de ventas que no son de ese año:
                #for venta in lifestore_sales:
                # fecha=venta[3]
                # if fecha[-4:]!="2020":
                # print(venta)
            if fecha[3:5]==mes and fecha[-4:]=="2020":
                ventas_mes.append(venta)

```



```

ventas=ventas_mes
else:
print("Se desplegará información de ingresos totales")
ventas=lifestore_sales
#Se obtienen las ganancias del periodo
ganancias=[]
for producto in lifestore_products:
id_producto = producto[0]
suma=0
for venta in ventas:
if venta[4]==1:
continue
if id_producto==venta[1]:
suma+=producto[2]
ganancias.append([suma, id_producto])
#Se despliega primero información de los ingresos totales del periodo
suma=0
for venta in ganancias:
suma+=venta[0]
print(f"En el periodo hubieron ingresos por ${suma}")
#Después, se consiguen ingresos por producto del periodo que el usuario haya elegido
#Después el usuario elige entre una serie de opciones de visualización por ganancias
print("Mostrando ganancias por producto:")
opcion = input("¿Cómo desea acomodar los productos?\nProductos con mayores ganancias:
1\nProductos con menores ganancias: 2\nSin acomodar: 3\n")
if opcion=='1':
print("Ha seleccionado: Productos con mayores ganancias")
#Si son los de más ganancias, los productos se acomodan de mayor a menor ganancia
ganancias.sort(reverse=True)
elif opcion=='2':
print("Ha seleccionado: Productos con menores ganancias")
#Si son los de menos ganancias, se acomodan de menor a mayor
ganancias.sort()
elif opcion=='3':
print("Ha seleccionado: Sin acomodar")
else:
print("Opción no encontrada")
continue
#El usuario elige de cuántos productos quiere hacer su top
cuantos=int(input("¿Cuántos productos desea ver?\n"))
if cuantos>len(ganancias):
cuantos=len(ganancias)
#Y finalmente, se muestra lo que el usuario eligió
i=0
for producto in ganancias:

```

```
if i<cuantos:
id_producto=producto[1]-1
print(f"Producto: {lifestore_products[id_producto][1]}")
print(f"Ingreso: {producto[0]}")
i+=1
else:
break
pregunta=input("Si desea continuar escriba 's'; de lo contrario, escriba cualquier letra\n")
if pregunta!="s":
salir=True
else:
print("Opción no encontrada")
pregunta=input("Si desea continuar escriba 's'; de lo contrario, escriba cualquier letra\n")
if pregunta!="s":
salir=True
print("Gracias por usar el sistema de consultas de LifeStore\nHasta pronto")
```

Anexo 2

Productos con menos ventas y búsquedas por categoría

Para procesadores

Mostrando los 5 productos con menos ventas

Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)

Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache

Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake)

Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake)

Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)

Mostrando los 10 productos con menos búsquedas

Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)

Procesador AMD Ryzen 3 3300X S-AM4, 3.80GHz, Quad-Core, 16MB L2 Cache

Procesador Intel Core i9-9900K, S-1151, 3.60GHz, 8-Core, 16MB Smart Cache (9na. Generación Coffee Lake)

Procesador Intel Core i5-9600K, S-1151, 3.70GHz, Six-Core, 9MB Smart Cache (9na. Generación - Coffee Lake)

Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth

Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)

Procesador Intel Core i7-9700K, S-1151, 3.60GHz, 8-Core, 12MB Smart Cache (9na. Generación Coffee Lake)

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

Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth

Para tarjetas de video

Mostrando los 5 productos con menos ventas

Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB 64-bit GDDR3, PCI Express 2.0

Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0

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

Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC, 8 GB 256 bit GDDR6, PCI Express x16 3.0

Mostrando los 10 productos con menos búsquedas

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

Tarjeta de Video Gigabyte NVIDIA GeForce RTX 2060 SUPER WINDFORCE OC, 8 GB 256 bit GDDR6, PCI Express x16 3.0

Tarjeta de Video MSI Radeon X1550, 128MB 64 bit GDDR2, PCI Express x16

Tarjeta de Video PNY NVIDIA GeForce RTX 2080, 8GB 256-bit GDDR6, PCI Express 3.0

MSI GeForce 210, 1GB GDDR3, DVI, VGA, HDCP, PCI Express 2.0

Tarjeta de Video VisionTek AMD Radeon HD5450, 2GB GDDR3, PCI Express x16

Tarjeta de Video Asus NVIDIA GeForce GTX 1050 Ti Phoenix, 4GB 128-bit GDDR5, PCI Express 3.0

Tarjeta de Video Gigabyte AMD Radeon R7 370 OC, 2GB 256-bit GDDR5, PCI Express 3.0

Para tarjetas madre

Mostrando los 5 productos con menos ventas

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 Z390 M GAMING, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel

Tarjeta Madre Gigabyte micro ATX Z490M GAMING X (rev. 1.0), Intel Z490, HDMI, 128GB DDR4 para Intel

Mostrando los 10 productos con menos búsquedas

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 PRIME Z390-A, 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

Tarjeta Madre ASRock ATX Z490 STEEL LEGEND, S-1200, Intel Z490, HDMI, 128GB DDR4 para Intel

Tarjeta Madre Gigabyte Micro ATX H310M DS2 2.0, S-1151, Intel H310, 32GB DDR4 para Intel

Tarjeta Madre ASUS micro ATX Prime H370M-Plus/CSM, S-1151, Intel H370, HDMI, 64GB DDR4 para Intel

Tarjeta Madre ASUS ATX ROG STRIX Z390-E GAMING, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel

Tarjeta Madre Gigabyte micro ATX Z390 M GAMING, S-1151, Intel Z390, HDMI, 64GB DDR4 para Intel

Para discos duros

Mostrando los 5 productos con menos ventas

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 S4500, 480GB, SATA III, 3.5", 7mm

SSD para Servidor Lenovo Thinksystem S4510, 480GB, SATA III, 2.5", 7mm

SSD Samsung 860 EVO, 1TB, SATA III, M.2

Mostrando los 10 productos con menos búsquedas

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

SSD Samsung 860 EVO, 1TB, SATA III, M.2

SSD para Servidor Lenovo Thinksystem S4500, 480GB, SATA III, 3.5", 7mm

SSD Western Digital WD Blue 3D NAND, 2TB, M.2

SSD Crucial MX500, 1TB, SATA III, M.2

Kit SSD Kingston KC600, 1TB, SATA III, 2.5, 7mm

SSD Kingston UV500, 480GB, SATA III, mSATA

SSD Kingston A2000 NVMe, 1TB, PCI Express 3.0, M2

Para memorias usb

Mostrando los 5 productos con menos ventas

Kit Memoria RAM Corsair Vengeance LPX DDR4, 2400MHz, 32GB, Non-ECC, CL16

Kit Memoria RAM Corsair Dominator Platinum DDR4, 3200MHz, 16GB (2x 8GB), Non-ECC, CL16, XMP

Mostrando los 10 productos con menos búsquedas

Kit Memoria RAM Corsair Dominator Platinum DDR4, 3200MHz, 16GB (2x 8GB), Non-ECC, CL16, XMP

Kit Memoria RAM Corsair Vengeance LPX DDR4, 2400MHz, 32GB, Non-ECC, CL16

Para pantallas

Mostrando los 5 productos con menos ventas

Makena Smart TV LED 32S2 32", HD, Widescreen, Gris

Seiki TV LED SC-39HS950N 38.5, HD, Widescreen, Negro

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

Mostrando los 10 productos con menos búsquedas

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

Hisense Smart TV LED 40H5500F 39.5, Full HD, Widescreen, Negro

Samsung Smart TV LED UN32J4290AF 32, HD, Widescreen, Negro

Hisense Smart TV LED 50H8F 49.5, 4K Ultra HD, Widescreen, Negro

Samsung Smart TV LED 43, Full HD, Widescreen, Negro

Seiki TV LED SC-39HS950N 38.5, HD, Widescreen, Negro

Samsung Smart TV LED UN55TU7000FXZX 55, 4K Ultra HD, Widescreen, Negro/Gris

Para bocinas

Mostrando los 5 productos con menos ventas

Lenovo Barra de Sonido, Alámbrico, 2.5W, USB, Negro

Acetek Bocina con Subwoofer AXF-290, Bluetooth, Inalámbrico, 2.1, 18W RMS, 180W PMPO, USB, Negro

Verbatim Bocina Portátil Mini, Bluetooth, Inalámbrico, 3W RMS, USB, Blanco
 Ghia Bocina Portátil BX300, Bluetooth, Inalámbrico, 40W RMS, USB, Rojo - Resistente al Agua
 Naceb Bocina Portátil NA-0301, Bluetooth, Inalámbrico, USB 2.0, Rojo
 Mostrando los 10 productos con menos búsquedas
 Lenovo Barra de Sonido, Alámbrico, 2.5W, USB, Negro
 Verbatim Bocina Portátil Mini, Bluetooth, Inalámbrico, 3W RMS, USB, Blanco
 Ghia Bocina Portátil BX300, Bluetooth, Inalámbrico, 40W RMS, USB, Rojo - Resistente al Agua
 Naceb Bocina Portátil NA-0301, Bluetooth, Inalámbrico, USB 2.0, Rojo
 Ghia Bocina Portátil BX900, Bluetooth, Inalámbrico, 2.1 Canales, 34W, USB, Negro - Resistente al Agua
 Ghia Bocina Portátil BX400, Bluetooth, Inalámbrico, 8W RMS, USB, Negro
 Ghia Bocina Portátil BX500, Bluetooth, Inalámbrico, 10W RMS, USB, Gris
 Ghia Bocina Portátil BX800, Bluetooth, Inalámbrico, 2.1 Canales, 31W, USB, Negro
 Acteck Bocina con Subwoofer AXF-290, Bluetooth, Inalámbrico, 2.1, 18W RMS, 180W PMPO, USB, Negro
 Logitech Bocinas para Computadora con Subwoofer G560, Bluetooth, Inalámbrico, 2.1, 120W RMS, USB, negro

Para audífonos

Mostrando los 5 productos con menos ventas
 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
 Genius GHP-400S Audífonos, Alámbrico, 1.5 Metros, Rosa
 Mostrando los 10 productos con menos búsquedas
 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
 Getttech Audífonos con Micrófono Sonority, Alámbrico, 1.2 Metros, 3.5mm, Negro/Rosa
 Klip Xtreme Audífonos Blast, Bluetooth, Inalámbrico, Negro/Verde
 Ginga Audífonos con Micrófono GI18ADJ01BT-RO, Bluetooth, Alámbrico/Inalámbrico, 3.5mm, Rojo
 Genius GHP-400S Audífonos, Alámbrico, 1.5 Metros, Rosa
 Iogear Audífonos Gamer GHG601, Alámbrico, 1.2 Metros, 3.5mm, Negro
 HyperX Audífonos Gamer Cloud Flight para PC/PS4/PS4 Pro, Inalámbrico, USB, 3.5mm, Negro

Anexo 3

Resultados de ventas por mes de cada categoría

Mostrando resultados por mes de ventas de producto:

Ventas del mes de enero

Se vendieron 25 procesadores
Se vendieron 0 tarjetas de video
Se vendieron 7 tarjetas madre
Se vendieron 20 discos duros
Se vendieron 0 memorias usb
Se vendieron 0 pantallas
Se vendieron 0 bocinas
Se vendieron 1 audifonos

Ventas del mes de febrero

Se vendieron 12 procesadores
Se vendieron 5 tarjetas de video
Se vendieron 7 tarjetas madre
Se vendieron 15 discos duros
Se vendieron 0 memorias usb
Se vendieron 0 pantallas
Se vendieron 2 bocinas
Se vendieron 0 audifonos

Ventas del mes de marzo

Se vendieron 15 procesadores
Se vendieron 9 tarjetas de video
Se vendieron 13 tarjetas madre
Se vendieron 14 discos duros
Se vendieron 0 memorias usb
Se vendieron 0 pantallas
Se vendieron 0 bocinas
Se vendieron 0 audifonos

Ventas del mes de abril

Se vendieron 28 procesadores
Se vendieron 7 tarjetas de video
Se vendieron 13 tarjetas madre
Se vendieron 24 discos duros
Se vendieron 0 memorias usb
Se vendieron 1 pantallas
Se vendieron 0 bocinas
Se vendieron 2 audifonos

Ventas del mes de mayo

Se vendieron 11 procesadores
Se vendieron 2 tarjetas de video
Se vendieron 8 tarjetas madre

Se vendieron 11 discos duros
Se vendieron 0 memorias usb
Se vendieron 1 pantallas
Se vendieron 0 bocinas
Se vendieron 2 audifonos
Ventas del mes de junio
Se vendieron 6 procesadores
Se vendieron 2 tarjetas de video
Se vendieron 0 tarjetas madre
Se vendieron 2 discos duros
Se vendieron 1 memorias usb
Se vendieron 0 pantallas
Se vendieron 0 bocinas
Se vendieron 0 audifonos
Ventas del mes de julio
Se vendieron 6 procesadores
Se vendieron 0 tarjetas de video
Se vendieron 1 tarjetas madre
Se vendieron 4 discos duros
Se vendieron 0 memorias usb
Se vendieron 0 pantallas
Se vendieron 0 bocinas
Se vendieron 0 audifonos
Ventas del mes de agosto
Se vendieron 0 procesadores
Se vendieron 0 tarjetas de video
Se vendieron 0 tarjetas madre
Se vendieron 3 discos duros
Se vendieron 0 memorias usb
Se vendieron 0 pantallas
Se vendieron 0 bocinas
Se vendieron 0 audifonos
Ventas del mes de septiembre
Se vendieron 0 procesadores
Se vendieron 1 tarjetas de video
Se vendieron 0 tarjetas madre
Se vendieron 0 discos duros
Se vendieron 0 memorias usb
Se vendieron 0 pantallas
Se vendieron 0 bocinas
Se vendieron 0 audifonos
Ventas del mes de octubre
Se vendieron 0 procesadores
Se vendieron 0 tarjetas de video
Se vendieron 0 tarjetas madre
Se vendieron 0 discos duros
Se vendieron 0 memorias usb

Se vendieron 0 pantallas
Se vendieron 0 bocinas
Se vendieron 0 audifonos
Ventas del mes de noviembre
Se vendieron 0 procesadores
Se vendieron 0 tarjetas de video
Se vendieron 0 tarjetas madre
Se vendieron 0 discos duros
Se vendieron 0 memorias usb
Se vendieron 0 pantallas
Se vendieron 0 bocinas
Se vendieron 0 audifonos
Ventas del mes de diciembre
Se vendieron 0 procesadores
Se vendieron 0 tarjetas de video
Se vendieron 0 tarjetas madre
Se vendieron 0 discos duros
Se vendieron 0 memorias usb
Se vendieron 0 pantallas
Se vendieron 0 bocinas
Se vendieron 0 audifonos