



# BT21

## Informe de Ventas 2020: departamento “Electrónicos”

Elaborado por Rosa María Almaraz Macuil

Disponible en: <https://github.com/MacuilR/Proyecto1>

# Informe de Ventas

Departamento de  
Electrónica



## **01 Ventas de productos e Intereses de clientes**

## **02 Reseñas de servicio**

## **03 Ingresos por ventas**

### **Ventas de productos e Intereses de clientes**

En el primer apartado se analizan las ventas durante el año 2020; se parte de la jerarquía de productos más vendidos para reconocer el Top 50 y Top 10 de los 96 que ofrece nuestro departamento. Así mismo se recopilan los productos menos vendidos y los productos que no tuvieron ventas.

También se encuentran las búsquedas que han realizado los interesados en los productos que ofrecemos; se presenta el Top 50 y Top 10 de productos más buscados, también aquellos que fueron buscados menos de 10 veces durante un año y los que no fueron buscados.

Otro aspecto presentado son las categorías a las que pertenecen los productos más vendidos, menos vendidos y sin ninguna venta.

### **Reseñas de servicio**

Continuando con la información de ventas, en este apartado se estudia a aquellos productos que fueron mejor reseñados por los compradores, aquellos con calificación baja y devoluciones.

### **Ingresos por ventas**

En la última parte del reporte se encuentran los datos de ingresos por venta, se enlistan los ingresos anuales y mensuales. Incluye una referencia a los meses en que las ventas fueron mayores y aquellos en que no hubo ventas.

## INTRODUCCIÓN

Este Informe presenta 3 aspectos clave para la toma de decisiones empresariales: cantidad de productos vendidos, nicho de mercado, reseñas por servicio e ingresos derivados de ventas.

A partir del análisis que se realizó se detectaron dos problemas fundamentales:

1. Existe una gran cantidad de productos que no fueron vendidos o buscados durante el último año.
2. Se vio una caída estrepitosa en ventas desde inicios del segundo trimestre para llegar al cuarto trimestre con ventas extremadamente bajas o sin ellas.

En las siguientes páginas podrá encontrar la presentación del trabajo que consiste en:

- Elaboración de código para la organización y obtención de datos relevantes.
- Presentación de los resultados
- Sugerencias para la solución del problema que se ha presentado.

## ACCESO A LA INFORMACIÓN

La seguridad representa una parte fundamental para la preservación de la información en ventas por lo que podrá acceder al sistema únicamente aquel usuario que sea reconocido por la base de datos “admin”. Mediante el siguiente código se genera un Inicio de sesión y acceso a la información resguardada, especificando las condiciones de ingreso.

```
# Lista de administradores, son los únicos con el poder de acceder a información de ventas.

admin = [['Alberto1020', '030920lop'], ['LorenaD1702', '020920lin'], ['CynthiaG2109', 'len182330']]

#Se le pide al usuario sus datos
usuario = input("Usuario: ")
contrasenia = input("Contraseña: ")

text_bienvenida = print('\t\t\t\t;Hola administrador!\n\nIngrese a la información actualizada que desea
conocer')

#Las condiciones para que el usuario pueda ingresar
if usuario == admin[0][0] and contrasenia == admin[0][1]:
| print(text_bienvenida)

elif usuario == admin[1][0] and contrasenia == admin[1][1]:
| print(text_bienvenida)

elif usuario == admin[2][0] and contrasenia == admin [2][1]:
| print(text_bienvenida)
else:
| print('Lo sentimos, usted no está autorizado para ingresar a este sistema')
```

A continuación, se despliega un menú donde el administrador elige entre:

- A. Ventas de productos e intereses de los clientes
- B. Reseñas de servicios
- C. Ingresos por ventas

Otra medida de seguridad que seguimos es el reinicio de sesión cada que el usuario quiera pasar de una opción a otra.



## VENTAS DE PRODUCTOS E INTERESES DE LOS CLIENTES

### VENTAS

#### TOP 50 Y TOP 10 EN VENTAS

Para realizar el análisis se tomaron los datos de identificación del producto de la lista “lifestore\_sales” y se contó la frecuencia con la que cada id se encontraba en la lista de ventas, con ambos datos se creó una nueva lista, “v\_acum” que fue ordenada de manera descendiente para concluir con un TOP 50 y un TOP10 en ventas.

#### MENORES VENTAS Y PRODUCTOS SIN VENTAS

Desde la lista “v\_acum” se ordeno de manera ascendente tomando todos aquellos elementos que se vendieron menos de 10 veces. A partir de la lista “lifestore\_products” y “lifestore\_sales”, se tomaron los identificadores de producto y se compararon para reconocer a aquellos que no habían registrado ventas durante el periodo, de la misma lista.

```
print('\t\tTOP 50 DE PRODUCTOS MÁS VENDIDOS\n')
ventas = [producto[1] for producto in lifestore_sales] #Id de los productos vendidos.
v_acum = [[ventas.count(frec), frec] for frec in set(ventas)] #frecuencia por producto

v_acum.sort(reverse = True)
top_50 = v_acum[0:49]
print('Se presentan los 50 productos con mayores ventas, [frecuencia de ventas, id del producto]:\n\n',
top_50)

top_10 = v_acum[0:9]
print('El Top 10 de los productos más vendidos es: \n\n', top_10)

v_acum.sort()
menores_ventas = v_acum[0:27]
print('Los productos con menores ventas son: \n\n', menores_ventas)

ids_lp = [elemento[0] for elemento in lifestore_products]
ids_ls = [elemento[1] for elemento in lifestore_sales]
sin_v = [[ids_ls.count(frec), frec] for frec in set(ids_lp)]
sin_v.sort()
no_ventas = sin_v[0:54]
print('Los productos sin ventas en el período de estudio segun [frecuencia de productos en ventas, id]:
\n\n', no_ventas)
```

### BUSQUEDAS

#### TOP 50 Y TOP 10 EN BÚSQUEDAS

De la lista “lifestore\_searches” se tomó por cada elemento el identificador de producto para rescatar cuántas veces había sido buscado por los posibles clientes, se creó una nueva lista llamada num\_b donde aparece la frecuencia de búsqueda y el id del producto. Para conocer el

TOP 50 y TOP 10 de los productos más buscados se ordenó num\_b de manera descendiente y se tomaron los primeros 50 y 10 elementos.

## LOS MENOS BUSCADOS Y LOS QUE NO BUSCARON

Para estos datos la lista num\_b se ordenó de manera ascendente y se extrajeron los elementos que fueron buscados menos de 10 veces durante el periodo de estudios. Posteriormente se tomaron los identificadores de productos de las listas “lifestore\_products” y “lifestore\_searches”, para encontrar aquellos elementos no buscados.

```
print('\n\t\tTOP EN BÚSQUEDAS\n')

busqueda = [producto[1] for producto in lifestore_searches]
num_b = [[busqueda.count(frec),frec] for frec in set(busqueda)] #Se presenta el id de los productos y su
frecuencia de búsqueda

num_b.sort(reverse = True)
top_50_busq = num_b[0:49]
print('Se presentan los 50 productos que más buscan los clientes, [frecuencia de busqueda, id producto]
:\n\n', top_50_busq)

top_10_busq = num_b[0:10]
print('Top 10 de productos más buscados, [frecuencia de busqueda, id producto: \n\n', top_10_busq)

print('\n\t\tLOS MENOS BUSCADOS\n')
num_b.sort()
menores_b = num_b[0:29]
print('Los productos menos buscados son, [frecuencia de busqueda, id producto]: \n\n', menores_b)

print('\n\t\tLOS QUE NADIE BUSCÓ\n')
ids_lp = [elemento[0] for elemento in lifestore_products]
ids_lse = [elemento[1] for elemento in lifestore_searches]
sin_busq = [[ids_lse.count(frec), frec] for frec in set(ids_lp)]
sin_busq.sort()
no_busq = sin_busq[0:40]
print('Los productos sin ventas en el periodo de estudio son, [frecuencia de busqueda, id]:\n\n', no_busq)
```

## PRODUCTOS POR CATEGORÍA

Primero se tomó la categoría de los productos en la lista “lifestore\_products” y se generó una nueva lista llamada categoría\_lp, posteriormente se contó y ordenó cuántos productos pertenecen a cada categoría con la lista num\_prod\_cat.

Para el análisis de ventas por categoría se tomaron de la lista “lifestore\_products” el identificador de producto y la categoría a la que pertenecen y se creó una lista llamada id\_categ, después se implementaron ciclos para conocer la frecuencia con la que aparecían estas categorías en el TOP 50 de ventas y en los productos no vendidos.

```

print('\n\t\tPRODUCTOS POR CATEGORÍA\n')
categoria_lp = [elemento[3] for elemento in lifestore_products]
num_prod_cat = [[categoria_lp.count(frec), frec] for frec in set(categoria_lp)]
num_prod_cat.sort(reverse = True)
orden_cat = num_prod_cat[0:8]
print('Categorías ordenadas según el número de productos de cada una de ellas: \n', orden_cat)

print('\n\t\tVENTAS POR CATEGORÍA\n')
id_categ = [[elemento[0], elemento[3]] for elemento in lifestore_products] #lista con [id,categoria]

print('Dentro del top 50 de ventas se muestran las categorías de productos a las que pertenecen.Se presentan
como: [id, frecuencia de venta, categoria] \n')

for id in top_50:
    for c in id_categ:
        if id[1] == c[0]:
            categ_top50 = [id[1], id[0],c[1]]
            print(categ_top50)

print('Dentro de los productos sin ventas se muestran las categorías a las que pertenecen. Se presentan
como: [id, frecuencia de venta, categoria]\n')
for id in no_ventas:
    for c in id_categ:
        if id[1] == c[0]:
            categ_NoVentas = [id[1],id[0],c[1]]
            print(categ_NoVentas)

```

## RESEÑAS

Se creó una lista llamada `id_ScoreDev` que recoge los elementos de identificación del producto, la calificación que dieron los compradores del 0 al 5 y si el producto se devolvió (1= sí, 0=no); después implementamos ciclos y condiciones para saber qué productos habían sido calificados con 5, 4 ,3, 2 y 1. Este mismo proceso se realizó para conocer aquellos productos que fueron devueltos.

```
print('\t\t RESEÑAS\n')
print('\t\t MEJORES RESEÑAS\n')

id_ScoreDev = [[el[1], el[2], el[4]] for el in lifestore_sales]

print('Los productos con reseñas de 5 estrellas y sin devoluciones son: [id, score, dev 1=True]\n')

for score in id_ScoreDev:
    if score[1] == 5:
        score5 = [score[0], score[1], score[2]]
        print(score5)

print('Los productos con reseñas de 4 estrellas y sin devoluciones son: [id, score, dev 1=True] \n')
for score in id_ScoreDev:
    if score[1] == 4:
        score4 = [score[0], score[1], score[2]]
        print(score4)

print('Los productos con devoluciones son: [id, score, dev 1=True]\n')
for dev in id_ScoreDev:
    if dev[2] == 1:
        devolucion = [dev[0], dev[1], dev[2]]
        print(devolucion)
```



## INGRESOS POR VENTAS

Para conocer los ingresos por ventas anuales y mensuales creamos ciclos con condiciones y contadores. Tomamos las listas “lifestore\_sales” y “lifestore\_products”. Para los ingresos anuales contamos cuantas veces se presentaba un producto en la lista de ventas e hicimos la sumatoria de los ingresos que generó su venta mediante el precio. Para sacar el ingreso promedio mensual únicamente dividimos el ingreso anual entre 12.

```
print('\t\tINGRESOS POR VENTAS\n')

ingreso_v = 0

for id in lifestore_sales:
    for precio in lifestore_products:
        if id[1] == precio[0]:
            ingreso_v += precio[2]

ingreso_promM = ingreso_v / 12 # Las ventas mensuales promedio

print('El ingreso por ventas durante 2020 fue de:\t$', ingreso_v, '\nEl valor promedio por ventas mensuales es de:\t', round(ingreso_promM,2))
```

Para la elaboración de ingreso mensuales se creó una lista llamada fecha\_v donde únicamente tomamos el identificador de producto y la fecha de venta desde la lista “lifestore\_sales”, mediante el uso de slice capturamos el elemento de la fecha que nos servía para el análisis, en este caso el mes.

```
print('\n\t\tINGRESOS MENSUALES POR VENTAS\n')

fecha_v = [[el[3], el[1]] for el in lifestore_sales]
mes = slice(3,5,1)
```

En cada mes se implementaron ciclos y condiciones, en conjunto con banderas y condiciones; para cada elemento de la lista fecha\_v si la bandera corresponde al mes de enero se cuentan las ventas en ese mes; mediante el identificador de producto se extrae y suma el precio de los productos vendidos en el mes. Finalmente se obtiene el ingreso promedio diario dividiendo el ingreso mensual entre treinta.

```
## ENERO
enero = '01'
producto_v1 = 0
ingreso_v1 = 0

for elemento in fecha_v:
    if enero == elemento[0][mes]:
        producto_v1 += 1
```

```
for precio in lifestore_products:
    if elemento[1] == precio[0]:
        ingreso_v1 += precio[2]

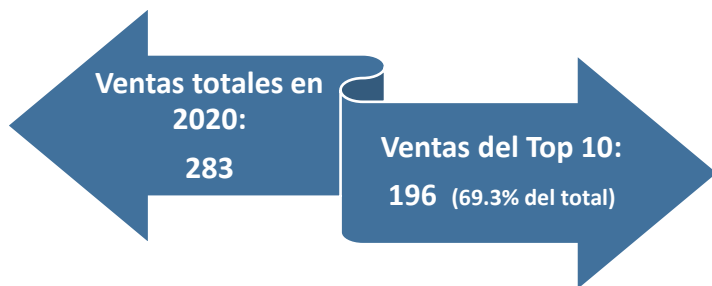
ingreso_vpm1 = ingreso_v1 / 30

print('\n\tENERO\n\nLas ventas durante el mes de enero fueron de: \t$', ingreso_v1, '\nLas ventas promedio
por día ascienden a: \t$', round(ingreso_vpm1,2))
```

## RESULTADOS

# Top 10 en Ventas

Frecuencia Ventas	ID	Nombre
50	54	SSD Kingston A400, 120GB, SATA III, 2.5", 7mm
42	3	Procesador AMD Ryzen 5 2600, S-AM4, 3.40GHz, Six-Core, 16MB L3 Cache, con Disipador Wraith Stealth
20	5	Procesador Intel Core i3-9100F, S-1151, 3.60GHz, Quad-Core, 6MB Cache (9na. Generación - Coffee Lake)
18	42	Tarjeta Madre ASRock Micro ATX B450M Steel Legend, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
15	57	SSD Adata Ultimate SU800, 256GB, SATA III, 2.5", 7mm
14	29	Tarjeta Madre ASUS micro ATX TUF B450M-PLUS GAMING, S-AM4, AMD B450, HDMI, 64GB DDR4 para AMD
13	2	Procesador AMD Ryzen 5 3600, S-AM4, 3.60GHz, 32MB L3 Cache, con Disipador Wraith Stealth
13	4	Procesador AMD Ryzen 3 3200G con Gráficos Radeon Vega 8, S-AM4, 3.60GHz, Quad-Core, 4MB L3, con Disipador Wraith Spire
11	47	SSD XPG SX8200 Pro, 256GB, PCI Express, M.2



Rango de precios:  
\$259.00 - \$4,209.00

El top 10 conforma el  
66.21% de las búsquedas

**BT21**

## Ingresos por ventas

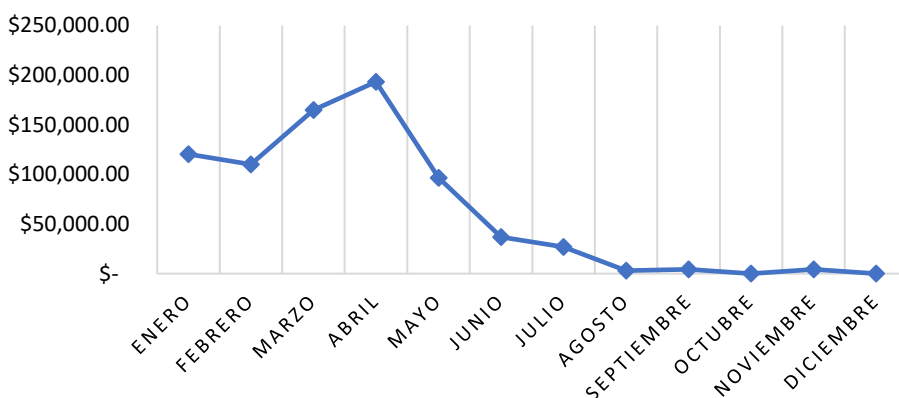
**BT21**

Ingresos por Ventas		
	Mes	Promedio Diario
Enero	\$ 120,237.00	\$ 4,007.90
Febrero	\$ 110,139.00	\$ 3,671.30
Marzo	\$ 164,729.00	\$ 5,490.97
Abril	\$ 193,295.00	\$ 6,443.17
Mayo	\$ 96,394.00	\$ 3,213.13
Junio	\$ 36,949.00	\$ 1,231.63
Julio	\$ 26,949.00	\$ 898.30
Agosto	\$ 3,077.00	\$ 102.57
Septiembre	\$ 4,199.00	\$ 139.97
Octubre	\$ -	\$ -
Noviembre	\$ 4,209.00	\$ 140.30
Diciembre	\$ -	\$ -
Anual	\$ 760,177.00	\$ 2,111.60

Promedio de ventas mensuales	\$ 63,348.08
------------------------------	--------------

Las mayores ventas se encontraron en los meses de marzo, enero y febrero. A partir de abril se ve una tendencia en la caída de las ventas para llegar a octubre y diciembre con cero ventas.

**INGRESOS POR VENTAS DURANTE EL AÑO 2020**



De los 96 productos con los que cuenta el departamento electrónico de BT21, 55 de ellos no generaron ninguna venta durante el 2020. Del total de 5341 productos en el almacén, de los cuales un 40.7% corresponde a los productos que no tuvieron ventas durante el período de estudio.

Una característica importante es que el rango de precio en esta lista de productos va de \$137.00 a \$30,449.00, y pertenecen a categorías de productos que tampoco encontramos en el top 10 de más vendidos, un ejemplo de ello son las pantallas y tarjetas madre de precios elevados.

## CONCLUSIONES



Dentro de la evidencia obtenida a través del análisis de datos de las ventas durante 2020 podemos llegar a la conclusión de que el rango de precios representa un elemento importante para este departamento ya que se observó que el producto más caro dentro del Top 10 tiene un precio de \$4,209.00 mientras el límite superior en la lista de los productos no vendidos es de \$30,449.00.

Así mismo el gasto que implica mantener todo el stock de productos que no se vendieron durante el último año irá mermando las ganancias del departamento si no se toma una acción para venderlos, además de que al ser un sector tecnológico y las actualizaciones de versiones y avances en la tecnología se dan varias veces al año hará que nuestros productos queden rezagados ante las nuevas versiones que pueden llegar a ser más eficientes y menos costosas.

Las estrategias que se proponen para evitar que el stock de los productos no vendidos siga representando un gasto al departamento son dos: la primera consiste en abrir un período de descuentos donde se les aplique cierto porcentaje de descuento a los productos que han permanecido sin ventas por más tiempo; la segunda y más importante es evaluar la posibilidad de implementar un estudio de mercado donde el objetivo es reconocer el mercado al que se dirigen los productos que están disponibles y ampliar el espectro para que nuestros productos rezagados puedan llegar a clientes que estén dispuestos a consumirlos.