



# LIFE STORE

---

## BUSINESS ANALYSIS



## TABLE OF CONTENTS

---

INTRODUCCIÓN .....	3
1. PRIMERA PARTE:.....	4
Productos más vendidos y productos rezagados ..	4
LISTADO CON LOS 5 PRODUCTOS CON MAYORES VENTAS .....	4
Resultado de este análisis:.....	5
LISTADO CON LOS 10 PRODUCTOS CON MAYORES BÚSQUEDAS .....	5
Resultado de este análisis:.....	5
POR CATEGORIA: 5 PRODUCTOS CON MENORES VENTAS.....	6
Resultado de este análisis:.....	8
POR CATEGORIA: 10 PRODUCTOS CON MENORES BÚSQUEDAS.....	8
Resultado de este análisis:.....	9
2. SEGUNDA PARTE: PRODUCTOS POR RESEÑA EN EL SERVICIO .....	10
LISTADO CON LOS 5 PRODUCTOS CON MEJORES RESEÑAS.....	10
Resultado de este análisis:.....	11
LISTADO CON LOS 5 PRODUCTOS CON PEORES RESEÑAS .....	11
Resultado de este análisis:.....	11
3. TERCERA PARTE: totales de ingresos y ventas promedio anuales y mensuales .....	12
TOTAL, DE INGRESO Y VENTAS PROMEDIO MENSUALES Y ANUALES .....	12
Resultado de este análisis:.....	13
4. SOLUCIÓN/ CONCLUSIÓN .....	14



## INTRODUCCIÓN

---

LifeStore es una tienda virtual que maneja una amplia gama de artículos, recientemente, la Gerencia de ventas, se percató que la empresa tiene una importante acumulación de inventario. Asimismo, se ha identificado una reducción en las búsquedas de un grupo importante de productos, lo que ha redundado en una disminución sustancial de sus ventas del último trimestre.

Debido a la situación, se ha realizado el siguiente análisis de la rotación de productos para identificar los siguientes elementos:

- 1) **Productos más vendidos y productos rezagados** a partir del análisis de las categorías con menores ventas y categorías con menores búsquedas.
- 2) **Productos por reseña en el servicio** a partir del análisis de categorías con mayores ventas y categorías con mayores búsquedas.
- 3) **Estrategias de solución:**
  - a. Retirar productos del mercado,
  - b. Reducir la acumulación de inventario considerando los datos de ingresos y ventas mensuales que se deriven de este análisis.

## 1. PRIMERA PARTE:

### PRODUCTOS MÁS VENDIDOS Y PRODUCTOS REZAGADOS

---

#### LISTADO CON LOS 5 PRODUCTOS CON MAYORES VENTAS

Creemos una función llamada `"get_ordenar_productos"` la cual utiliza como parámetros `"un archivo con datos a ordenar"`, `"un límite"`, `"top que puede ser True or False"`, y retorna una lista con tuplas que contiene el id de los productos y la cantidad de ventas por cada uno de ellos.

```
def get_ordenar_productos(lifestore_sales, limit = 5, top = True,):  
    ...  
    Retorna una lista de tuplas con la cantidad ventas que tuvo cada id_product y los ordena descendente  
    ...  
    productos = {}  
    for sale in lifestore_sales:  
        prod_id = sale[IDX_SALES_ID_PRODUCTO]  
        if prod_id not in productos.keys():  
            productos[prod_id] = 0  
        productos[prod_id] +=1  
    productos_ordenados = sorted(productos.items(), key = lambda x:x[1], reverse = top)  
    return productos_ordenados[:limit]
```

En la función `"print_reports ()"` en la que estaremos llamando a todas nuestras funciones para retornar los datos que requerimos para el reporte, asignamos la función anterior a la variable `"mayores_ventas"` pasando como argumento el archivo `"lifestore_sales"` para mediante un recorrido de la lista retornada por la función, imprimir en pantalla los productos con mayores ventas, de la siguiente manera:

```
mayores_ventas = get_ordenar_productos(lifestore_sales)  
print("\n*LOS PRODUCTOS CON MAYORES VENTAS SON:")  
for venta in mayores_ventas:  
    print(f"- id: {venta[0]}: {nombres[venta[0]][0][:30]}")
```

Resultado de este análisis:

```
*LOS PRODUCTOS CON MAYORES VENTAS SON:
- id: 54: SSD Kingston A400, 120GB, SATA
- id: 3: Procesador AMD Ryzen 5 2600, S
- id: 5: Procesador Intel Core i3-9100F
- id: 42: Tarjeta Madre ASRock Micro ATX
- id: 57: SSD Adata Ultimate SU800, 256G
```

## LISTADO CON LOS 10 PRODUCTOS CON MAYORES BÚSQUEDAS

Utilizando la función anterior pasando ahora como argumento el archivo "lifestore\_searches" y cambiando nuestro limite a 10, asignamos nuestra función a la variable "mayores\_búsquedas"

```
mayores_búsquedas = get_ordenar_productos(lifestore_searches, 10, True)
print("\n*LOS PRODUCTOS CON MAYORES BÚSQUEDAS SON:")
for busqueda in mayores_búsquedas:
    print(f"- id: {busqueda[0]}: {nombres[busqueda[0]][0][:30]}")
```

recorremos la lista de nuevo, pero ahora retorna los 10 productos con mayores búsquedas:

Resultado de este análisis:

```
*LOS PRODUCTOS CON MAYORES BÚSQUEDAS SON:
- id: 54: SSD Kingston A400, 120GB, SATA
- id: 57: SSD Adata Ultimate SU800, 256G
- id: 29: Tarjeta Madre ASUS micro ATX T
- id: 3: Procesador AMD Ryzen 5 2600, S
- id: 4: Procesador AMD Ryzen 3 3200G c
- id: 85: Logitech Audífonos Gamer G635
- id: 67: TV Monitor LED 24TL520S-PU 24,
- id: 7: Procesador Intel Core i7-9700K
- id: 5: Procesador Intel Core i3-9100F
- id: 47: SSD XPG SX8200 Pro, 256GB, PCI
```



## POR CATEGORIA: 5 PRODUCTOS CON MENORES VENTAS

Creamos una función llamada "get\_menos\_ventas\_y\_búsquedas\_por\_categoria" la cual utiliza como parámetros un diccionario que agrupa los id de productos por la categoría a la que pertenecen, "categorías" que creamos en una función anterior "get\_productos\_agrupar\_por\_categoria":

```
def get_productos_agrupar_por_categoria():
    ...

    Agrupa los produc_id por categorías, retorna un diccionario
    {'procesadores': [1, 2, 3...], 'otra categoría': [más product_id que pertenecen a esta]}
    ...

    categorias = {}
    for producto in lifestore_products:
        categoria_prod = producto[IDX_PRODUCTS_CATEGORY]
        if categoria_prod not in categorias.keys():
            categorias[categoria_prod] = []
        categorias[categoria_prod].append(producto[IDX_PRODUCTS_ID])
    return categorias
```

Y nuestro archivo con datos a ordenar, que en este caso es "lifestore\_sales", esta función retorna un diccionario que contiene el id de los productos, las ventas y búsquedas de los productos por categoría, y las ordena:

```
def get_menos_ventas_y_búsquedas_por_categorias(categorias, lifestore_sales):
    ...

    Relaciona las categorías de los productos con su id, ventas, búsquedas y las ordena
    A partir de las categorías agrega:
    - las ventas realizadas de esa categoría
    - las búsquedas a los productos de esa categoría
    - las productos con menores ventas de esa categoría
    - los productos con menores búsquedas de esa categoría
    {'procesadores':
        {
            prod_id: [1, 2, 3],
            ventas:[1, 1, 5, '24/07/2020', 0]],
            ordenados_ventas:((id_product, 32), (id_product, 47)), ordena ascendente
            búsquedas:[[id_search,id_prod ]],
            ordenados_búsquedas: [(id_prod, 54)] ordena ascendente
        }
    }
    ...
```

```

prod_vendidos_por_categorias = {}
for category in categorias:
    prod_vendidos_por_categorias[category] = {'prod_id':[], 'ventas':[], 'ordenados_ventas':[],
    'busquedas':[], 'ordenados_busquedas':[]}
    contador_ventas = {}
    contador_busquedas = {}
    for prod_id in categorias[category]:
        contador_ventas[prod_id] = 0
        prod_vendidos_por_categorias[category]['prod_id'].append(prod_id)
        contador_busquedas[prod_id] = 0
        for sale in lifestore_sales:
            devolucion = sale[IDX_SALES_REFUND]
            if devolucion == 0 and prod_id == sale[IDX_SALES_ID_PRODUCTO]:
                contador_ventas[prod_id] += 1
                prod_vendidos_por_categorias[category]['ventas'].append(sale)
                break
        for search in lifestore_searches:
            if prod_id == search[IDX_SEARCHES_ID_PRODUCT]:
                contador_busquedas[prod_id] += 1
                prod_vendidos_por_categorias[category]['busquedas'].append(search)
                break
    ordenados_ventas = sorted(contador_ventas.items(), key = lambda x:x[1], reverse = False)
    prod_vendidos_por_categorias[category]['ordenados_ventas'] = ordenados_ventas
    ordenados_menores_busquedas = sorted(contador_busquedas.items(), key = lambda x:x[1], reverse = False)
    prod_vendidos_por_categorias[category]['ordenados_busquedas'] = ordenados_menores_busquedas
return prod_vendidos_por_categorias

```

En nuestra función “`print_reports ()`” asignamos esta función a la variable “`menores_ventas_y_busquedas`” y recorriendo nuestras categorías en el diccionario obtenemos los 5 productos con menores ventas por categoría:

```

menores_ventas_y_busquedas = get_menos_ventas_y_busquedas_por_categorias(categorias, lifestore_sales)
print('\n*LOS PRODUCTOS CON MENORES VENTAS POR CATEGORÍA SON:')
for category in menores_ventas_y_busquedas:
    menores_ventas_y_busquedas[category]['ordenados_ventas']
    print(f"\n*** {category} ***")
    for i in range(5):
        if i < len(menores_ventas_y_busquedas[category]['ordenados_ventas']):
            product = menores_ventas_y_busquedas[category]['ordenados_ventas'][i]
            nombre = nombres[product[0]]
            id = menores_ventas_y_busquedas[category]['prod_id'][i]
            print(f"- El producto '{nombre[0][:15]}' con id: {id} se vendió: {product[1]} veces")
print('\n*LOS PRODUCTOS CON MENORES BUSQUEDAS POR CATEGORÍA SON:')

```

## Resultado de este análisis:

\*LOS PRODUCTOS CON MENORES VENTAS POR CATEGORÍA SON:

```

*** procesadores ***
- El producto 'Procesador Inte', con id: 1 se vendió: 0 veces
- El producto 'Procesador AMD ', con id: 2 se vendió: 1 veces
- El producto 'Procesador AMD ', con id: 3 se vendió: 1 veces
- El producto 'Procesador AMD ', con id: 4 se vendió: 1 veces
- El producto 'Procesador AMD ', con id: 5 se vendió: 1 veces

*** tarjetas madre ***
- El producto 'Tarjeta Madre A', con id: 29 se vendió: 0 veces
- El producto 'Tarjeta Madre A', con id: 30 se vendió: 0 veces
- El producto 'Tarjeta Madre A', con id: 31 se vendió: 0 veces
- El producto 'Tarjeta Madre G', con id: 32 se vendió: 0 veces
- El producto 'Tarjeta Madre G', con id: 33 se vendió: 0 veces

*** memorias usb ***
- El producto 'Kit Memoria RAM', con id: 60 se vendió: 0 veces
- El producto 'Kit Memoria RAM', con id: 61 se vendió: 1 veces

*** bocinas ***
- El producto 'Lenovo Barra de', con id: 74 se vendió: 0 veces
- El producto 'Acteck Bocina c', con id: 75 se vendió: 0 veces
- El producto 'Verbatim Bocina', con id: 76 se vendió: 0 veces
- El producto 'Ghia Bocina Por', con id: 77 se vendió: 0 veces
- El producto 'Naceb Bocina Po', con id: 78 se vendió: 0 veces

*** tarjetas de video ***
- El producto 'Tarjeta de Vide', con id: 10 se vendió: 0 veces
- El producto 'Tarjeta de Vide', con id: 11 se vendió: 0 veces
- El producto 'Tarjeta de Vide', con id: 12 se vendió: 0 veces
- El producto 'Tarjeta de Vide', con id: 13 se vendió: 0 veces
- El producto 'Tarjeta de Vide', con id: 14 se vendió: 0 veces

*** discos duros ***
- El producto 'SSD Addlink Tec', con id: 47 se vendió: 0 veces
- El producto 'SSD para Servid', con id: 48 se vendió: 0 veces
- El producto 'SSD para Servid', con id: 49 se vendió: 0 veces
- El producto 'SSD para Servid', con id: 50 se vendió: 0 veces
- El producto 'SSD Samsung 860', con id: 51 se vendió: 0 veces

*** pantallas ***
- El producto 'Makena Smart TV', con id: 62 se vendió: 0 veces
- El producto 'Seiki TV LED SC', con id: 63 se vendió: 0 veces
- El producto 'Samsung TV LED ', con id: 64 se vendió: 0 veces
- El producto 'Samsung Smart T', con id: 65 se vendió: 0 veces
- El producto 'Makena Smart TV', con id: 66 se vendió: 0 veces

*** audifonos ***
- El producto 'ASUS Audifonos ', con id: 84 se vendió: 0 veces
- El producto 'Acer Audifonos ', con id: 85 se vendió: 0 veces
- El producto 'Audifonos Gamer', con id: 86 se vendió: 0 veces
- El producto 'Energy Sistem A', con id: 87 se vendió: 0 veces
- El producto 'Genius GHP-400S', con id: 88 se vendió: 0 veces

```

## POR CATEGORIA: 10 PRODUCTOS CON MENORES BÚSQUEDAS

Utilizando nuestra función anterior pero ahora haciendo referencia a nuestras búsquedas ordenadas:

```

print('\n*LOS PRODUCTOS CON MENORES BUSQUEDAS POR CATEGORÍA SON:')
for category in menores_ventas_y_búsquedas:
    menores_ventas_y_búsquedas[category]['ordenados_búsquedas']
    print(f"\n*** {category} ***")
    for i in range(10):
        if i < len(menores_ventas_y_búsquedas[category]['ordenados_búsquedas']):
            product = menores_ventas_y_búsquedas[category]['ordenados_búsquedas'][i]
            nombre = nombres[product[0]]
            id = menores_ventas_y_búsquedas[category]['prod_id'][i]
            print(f"- El producto '{nombre[0][:15]}', con id: {id} se buscó: {product[1]} veces")

```



## Resultado de este análisis:

\*LOS PRODUCTOS CON MENORES BUSQUEDAS POR CATEGORÍA SON:

## \*\*\* procesadores \*\*\*

- El producto 'Procesador AMD ', con id: 1 se buscó: 1 veces
- El producto 'Procesador AMD ', con id: 2 se buscó: 1 veces
- El producto 'Procesador AMD ', con id: 3 se buscó: 1 veces
- El producto 'Procesador AMD ', con id: 4 se buscó: 1 veces
- El producto 'Procesador Inte', con id: 5 se buscó: 1 veces
- El producto 'Procesador Inte', con id: 6 se buscó: 1 veces
- El producto 'Procesador Inte', con id: 7 se buscó: 1 veces
- El producto 'Procesador Inte', con id: 8 se buscó: 1 veces
- El producto 'Procesador Inte', con id: 9 se buscó: 1 veces

## \*\*\* tarjetas madre \*\*\*

- El producto 'Tarjeta Madre A', con id: 29 se buscó: 0 veces
- El producto 'Tarjeta Madre A', con id: 30 se buscó: 0 veces
- El producto 'Tarjeta Madre A', con id: 31 se buscó: 0 veces
- El producto 'Tarjeta Madre A', con id: 32 se buscó: 0 veces
- El producto 'Tarjeta Madre G', con id: 33 se buscó: 0 veces
- El producto 'Tarjeta Madre A', con id: 34 se buscó: 0 veces
- El producto 'Tarjeta Madre G', con id: 35 se buscó: 0 veces
- El producto 'Tarjeta Madre A', con id: 36 se buscó: 0 veces
- El producto 'Tarjeta Madre A', con id: 37 se buscó: 0 veces
- El producto 'Tarjeta Madre A', con id: 38 se buscó: 1 veces

## \*\*\* memorias usb \*\*\*

- El producto 'Kit Memoria RAM', con id: 60 se buscó: 0 veces
- El producto 'Kit Memoria RAM', con id: 61 se buscó: 0 veces

## \*\*\* bocinas \*\*\*

- El producto 'Lenovo Barra de', con id: 74 se buscó: 0 veces
- El producto 'Verbatim Bocina', con id: 75 se buscó: 0 veces
- El producto 'Ghia Bocina Por', con id: 76 se buscó: 0 veces
- El producto 'Naceb Bocina Po', con id: 77 se buscó: 0 veces
- El producto 'Ghia Bocina Por', con id: 78 se buscó: 0 veces
- El producto 'Ghia Bocina Por', con id: 79 se buscó: 0 veces
- El producto 'Ghia Bocina Por', con id: 80 se buscó: 0 veces
- El producto 'Logitech Bocina', con id: 81 se buscó: 1 veces
- El producto 'Acteck Bocina c', con id: 82 se buscó: 1 veces
- El producto 'Ghia Bocina Por', con id: 83 se buscó: 1 veces

## \*\*\* tarjetas de video \*\*\*

- El producto 'Tarjeta de Vide', con id: 10 se buscó: 0 veces
- El producto 'Tarjeta de Vide', con id: 11 se buscó: 0 veces
- El producto 'Tarjeta de Vide', con id: 12 se buscó: 0 veces
- El producto 'Tarjeta de Vide', con id: 13 se buscó: 0 veces
- El producto 'Tarjeta de Vide', con id: 14 se buscó: 0 veces
- El producto 'Tarjeta de Vide', con id: 15 se buscó: 0 veces
- El producto 'MSI GeForce 210', con id: 16 se buscó: 1 veces
- El producto 'Tarjeta de Vide', con id: 17 se buscó: 1 veces
- El producto 'Tarjeta de Vide', con id: 18 se buscó: 1 veces
- El producto 'Tarjeta de Vide', con id: 19 se buscó: 1 veces

## \*\*\* discos duros \*\*\*

- El producto 'SSD Addlink Tec', con id: 47 se buscó: 0 veces
- El producto 'SSD para Servid', con id: 48 se buscó: 0 veces
- El producto 'SSD para Servid', con id: 49 se buscó: 0 veces
- El producto 'SSD XPG SX8200 ', con id: 50 se buscó: 1 veces
- El producto 'SSD Kingston A2', con id: 51 se buscó: 1 veces
- El producto 'Kit SSD Kingsto', con id: 52 se buscó: 1 veces
- El producto 'SSD Crucial MX5', con id: 53 se buscó: 1 veces
- El producto 'SSD Kingston UV', con id: 54 se buscó: 1 veces
- El producto 'SSD Western Dig', con id: 55 se buscó: 1 veces
- El producto 'SSD Kingston A4', con id: 56 se buscó: 1 veces

## \*\*\* pantallas \*\*\*

- El producto 'Makena Smart TV', con id: 62 se buscó: 0 veces
- El producto 'Samsung TV LED ', con id: 63 se buscó: 0 veces
- El producto 'Samsung Smart T', con id: 64 se buscó: 0 veces
- El producto 'Makena Smart TV', con id: 65 se buscó: 0 veces
- El producto 'Hisense Smart T', con id: 66 se buscó: 0 veces
- El producto 'Samsung Smart T', con id: 67 se buscó: 0 veces
- El producto 'Hisense Smart T', con id: 68 se buscó: 0 veces
- El producto 'Seiki TV LED SC', con id: 69 se buscó: 1 veces
- El producto 'TCL Smart TV LE', con id: 70 se buscó: 1 veces
- El producto 'TV Monitor LED ', con id: 71 se buscó: 1 veces

## \*\*\* audifonos \*\*\*

- El producto 'ASUS Audífonos ', con id: 84 se buscó: 0 veces
- El producto 'Acer Audífonos ', con id: 85 se buscó: 0 veces
- El producto 'Audífonos Gamer', con id: 86 se buscó: 0 veces
- El producto 'Energy Sistem A', con id: 87 se buscó: 0 veces
- El producto 'Getttech Audífo', con id: 88 se buscó: 0 veces
- El producto 'Klip Xtreme Aud', con id: 89 se buscó: 0 veces
- El producto 'Logitech Audifo', con id: 90 se buscó: 1 veces
- El producto 'Logitech Audífo', con id: 91 se buscó: 1 veces
- El producto 'Cougar Audifono', con id: 92 se buscó: 1 veces
- El producto 'Genius GHP-400S', con id: 93 se buscó: 1 veces

## 2. SEGUNDA PARTE: PRODUCTOS POR RESEÑA EN EL SERVICIO

### LISTADO CON LOS 5 PRODUCTOS CON MEJORES RESEÑAS

Creamos una función llamada “get\_ordenar\_por\_resenas” la cual utiliza como parámetros “un límite” y un “top que puede ser True or False”, y retorna un diccionario que contiene un **score acumulado**, una **cuenta** de las reseñas que tuvo el producto y un **promedio**, además de ordenar por mejores o peores reseñas según el **top**.

```
def get_ordenar_por_resenas(limit = 5, top = True):  
    """  
    Retorna un diccionario con el score acumulado, cuenta y promedio los 5 productos con las mejores o peores reseñas  
    """  
    resenas = {}  
    for sale in lifestore_sales:  
        prod_id = sale[IDX_SALES_ID_PRODUCTO]  
        if sale[IDX_SALES_REFUND] == 1:  
            score = 1  
        else:  
            score = sale[IDX_SALES_SCORE]  
        if prod_id not in resenas.keys():  
            resenas[prod_id] = {'score_acumulado':0,'cuenta':0}  
            resenas[prod_id]['score_acumulado'] = resenas[prod_id]['score_acumulado'] + score  
            resenas[prod_id]['cuenta'] = resenas[prod_id]['cuenta'] + 1  
    for prod_id in resenas:  
        prom = resenas[prod_id]['score_acumulado'] / resenas[prod_id]['cuenta']  
        resenas[prod_id]['prom'] = prom  
    promedios_ordenados = sorted(resenas.items(), key = lambda x:x[1]['prom'], reverse = top)  
    return promedios_ordenados[:limit]
```

En nuestra función “print\_reports ()” asignamos esta función a la variable “mejores\_resenas” y recorriendo nuestras categorías en el diccionario obtenemos los 5 productos con mejores reseñas:

```
print('\n\n *** SEGUNDA PARTE: Productos por reseña en el servicio ***')  
mejores_resenas = get_ordenar_por_resenas(limit = 5, top = True)  
print("\n\n*LOS PRODUCTOS CON MEJORES RESEÑAS SON: ")  
for producto in mejores_resenas:  
    print(f"- id: {producto[0]}: {nombres[producto[0]][0][:30]}")
```

Resultado de este análisis:

```
*LOS PRODUCTOS CON MEJORES RESEÑAS SON:
- id: 1: Procesador AMD Ryzen 3 3300X S
- id: 6: Procesador Intel Core i9-9900K
- id: 7: Procesador Intel Core i7-9700K
- id: 8: Procesador Intel Core i5-9600K
- id: 11: Tarjeta de Video ASUS AMD Rade
```

## LISTADO CON LOS 5 PRODUCTOS CON PEORES RESEÑAS

Utilizamos nuestra función anterior y en “`print_reports ()`” asignamos esta función a la variable “`peores_resenas`” y recorriendo nuestras categorías en el diccionario obtenemos los 5 productos con menores mejores reseñas:

```
peores_resenas = get_ordenar_por_resenas(limit = 5, top = False)
print("\n*LOS PRODUCTOS CON PEORES RESEÑAS SON: ")
for producto in peores_resenas:
    print(f"id: {producto[0]}: {nombres[producto[0]][0][:30]}")
```

Resultado de este análisis:

```
*LOS PRODUCTOS CON PEORES RESEÑAS SON:
id: 17: Tarjeta de Video Gigabyte AMD
id: 45: Tarjeta Madre ASRock ATX H110
id: 46: Tarjeta Madre Gigabyte micro A
id: 31: Tarjeta Madre AORUS micro ATX
id: 89: Cougar Audífonos Gamer Phontum
```



### 3. TERCERA PARTE:

## TOTALES DE INGRESOS Y VENTAS PROMEDIO ANUALES Y MENSUALES

### TOTAL, DE INGRESO Y VENTAS PROMEDIO MENSUALES Y ANUALES

Creamos la función `"get_totales"` la cual utiliza como parámetros un archivo con los ingresos y ventas, y un diccionario que agrupa los id de los productos con su nombre y precio `"name_and_price_product"` que creamos en la función `"get_product_name_and_price"`:

```
def get_product_name_and_price():
    """
    Iterar en mi listado para retornar un diccionario que contiene el id_product junto con su nombre y precio
    """
    name_and_price_product = {}
    for product in lifestore_products:
        name_and_price_product[product[IDX_PRODUCTS_ID]] = (product[IDX_PRODUCTS_NAME], product[IDX_PRODUCTS_PRICE])
    return name_and_price_product
```

Y retorna un diccionario que contiene las ventas e ingresos totales por año y por mes:

```
def get_totales(lifestore_sales, name_and_price_product):
    """
    Retorna un diccionario con las ventas e ingresos totales por año y por mes
    """
    totales = {}
    for sale in lifestore_sales:
        anio = sale[IDX_SALES_DATE].split('/')[2]
        if anio not in totales:
            totales[anio] = {'total_anual': 0, 'cuenta_ventas': 0, 'meses': {}}
        mes = sale[IDX_SALES_DATE].split('/')[1]
        if mes not in totales[anio]['meses']:
            totales[anio]['meses'][mes] = {'total_mes': 0, 'cuenta_mes': 0}
        devolucion = sale[IDX_SALES_REFUND]
        if devolucion == 0:
            totales[anio]['cuenta_ventas'] += 1
            totales[anio]['meses'][mes]['cuenta_mes'] += 1
            id_product = sale[IDX_SALES_ID_PRODUCTO]
            totales[anio]['total_anual'] = totales[anio]['total_anual'] + name_and_price_product[id_product][1]
            totales[anio]['meses'][mes]['total_mes'] += name_and_price_product[id_product][1]
    return totales
```

En nuestra función `"print_reports ()"` asignamos esta función a la variable `"totales"`, creamos un diccionario con los meses del año por nombre para ordenar nuestros resultados y



recorremos "totales" por año recuperando "total\_anual" e imprimiendo el ingreso anual directo en pantalla, por mes recuperando "total\_mes" sumándolo en nuestra variable "venta\_anual" para retornar el ingreso por mes. Utiliza nuestra variable "venta\_anual" la divide entre los 12 meses y devuelve el promedio de ingreso al mes. Luego recuperando "cuenta\_mes" de nuestro diccionario, asignamos a la variable "meses\_con\_mas\_ventas" la lista de los meses ordenados de acuerdo con las ventas.

### Resultado de este análisis:

```
*INGRESOS TOTALES

***EN 2020 EL INGRESO TOTAL FUE: $ 737916***
- El ingreso total en Enero del 2020 fue: $117738
- El ingreso total en Febrero del 2020 fue: $107270
- El ingreso total en Marzo del 2020 fue: $162931
- El ingreso total en Abril del 2020 fue: $191066
- El ingreso total en Mayo del 2020 fue: $91936
- El ingreso total en Junio del 2020 fue: $36949
- El ingreso total en Julio del 2020 fue: $26949
- El ingreso total en Agosto del 2020 fue: $3077
- En Septiembre del 2020 no hubo ingresos
- En Octubre del 2020 no hubo ingresos
- En Noviembre del 2020 no hubo ingresos
- En Diciembre del 2020 no hubo ingresos

-Las ventas promedio mensuales del 2020 fueron: 22.83

*MESES CON MAYORES VENTAS
- El mes Abril tuvo 74 ventas
- El mes Enero tuvo 52 ventas
- El mes Marzo tuvo 49 ventas
- El mes Febrero tuvo 40 ventas

***EN 2019 EL INGRESO TOTAL FUE: $ 0***
- En Enero del 2019 no hubo ingresos
- En Febrero del 2019 no hubo ingresos
- En Marzo del 2019 no hubo ingresos
- En Abril del 2019 no hubo ingresos
- En Mayo del 2019 no hubo ingresos
- En Junio del 2019 no hubo ingresos
- En Julio del 2019 no hubo ingresos
- En Agosto del 2019 no hubo ingresos
- En Septiembre del 2019 no hubo ingresos
- En Octubre del 2019 no hubo ingresos
- En Noviembre del 2019 no hubo ingresos
- En Diciembre del 2019 no hubo ingresos

-Las ventas promedio mensuales del 2019 fueron: 0.00

*MESES CON MAYORES VENTAS
- Este año no hubo ventas
```



## 4. SOLUCIÓN/ CONCLUSIÓN

---

Hay un claro aumento de ventas e ingresos del año 2019 al 2020. Por otro lado, en los primeros 4 meses del 2020 se registraron las mayores ventas y es evidente una disminución de estas los últimos 3 meses de este año.

- *En 2020 debido a la pandemia que se vive alrededor del mundo, las ventas online han aumentado de manera considerable, una posible solución para aumentar las ventas es implementar una tienda online o vía redes sociales para llegar más fácil a los clientes potenciales.*
- *Se sugiere también eliminar todos aquellos productos que no están teniendo interacción ni en búsquedas o ventas con los clientes.*
- *Potenciar la publicidad en redes de los productos con mejores reseñas, y hacer un estudio de los proveedores de aquellos con las peores para encontrar la causa de los problemas que estos productos están presentando. Y de este modo considerar el cambiar de proveedores o retirar el producto de la oferta.*