



# **PROYECTO 1. INTRODUCCIÓN A PYTHON**

**CAROLINA ZÚÑIGA  
RAMÍREZ**



# ÍNDICE

<b>1.INTRODUCCIÓN .....</b>	<b>3</b>
<b>2.DEFINICIÓN DEL CÓDIGO .....</b>	<b>4</b>
a.Listado de 15 Productos Más Vendidos	
b.Listado de 20 Productos Más Buscados	
c.Listado de 5 Productos con Menores Ventas	
d. Listado de 5 Productos con Menores Búsquedas	
e.Listado de 10 productos con las Mejores y Peores Reseñas	
f.Fecha y venta de los productos (20) Más y Menos vendidos	
g.Menú para ver los resultados Anteriores como Resumen	
<b>3.SOLUCIÓN AL PROBLEMA.....</b>	<b>11</b>
<b>4.REFERENCIAS.....</b>	<b>11</b>

## 1.INTRODUCCIÓN

Para esta primera entrega se desarrollo un código de Python con el fin de poner en práctica los elementos y bases de programación para el análisis de datos.

El objetivo en concreto de este proyecto fue que de acuerdo a los datos otorgados por LifeStore, una tienda virtual que maneja una variedad de artículos y por petición de la Gerencia de Ventas, hacer una identificación de grupos importantes dentro de los datos tales como mayores ventas por categoría, mayores búsquedas, así como las menores. También se pidió analizar cómo han estado calificados dichos productos y categorías para ofrecer una solución derivado de la baja de ventas en el anterior trimestre.

De acuerdo a los resultados obtenidos se llegó a la conclusión de eliminar de su inventario a la categoría de pantalla si se debiera tomar la decisión debido al gran inventario y pocas ventas , así como los productos en específico que cuentan con score 0 ya que llegan a representar retornos a la empresa y una mala reputación.

## 2. DEFINICIÓN DEL CÓDIGO

### a. Listado de 15 Productos Más Vendidos

```
"""
Listado de productos más vendidos con retornos a tienda (15 mayores)
"""
#Se empieza con una lista vacía en la cuál en la primera posición tendrá el id del
#producto y se espera que en el siguiente lugar estén las ventas por lo que se empieza en 0.
producto_ventas_retorno = []

for producto in lifestore_products:
    lista_interna = [producto[0], 0]

    producto_ventas_retorno.append(lista_interna)

#Ahora con este for, se busca en la lista lifestore_sales el id del producto
#y si este fue devuelto

for venta in lifestore_sales:
    producto_vendido = venta[1]
    devuelto = venta[4]

#Se usa una condición para validar si el producto fue devuelto o no
    if devuelto == 0:
        validez = True
    else:
        validez = False

#Si es devuelto se le restará a las ventas del producto y lo guardamos
#en la lista previamente creada
    if validez == True:
        producto_ventas_retorno[producto_vendido - 1][1] += 1

print("Los 15 productos con más ventas:")
print(sorted(producto_ventas_retorno, key = lambda x: -x[1])[:15])

Los 15 productos con más ventas:
[[54, 49], [3, 42], [5, 20], [42, 18], [57, 15], [4, 13], [29, 13], [2, 12], [47, 11], [12, 9], [48, 9], [7, 7], [44,
6], [18, 5], [8, 4]]
```

## b. Listado de 20 Productos Más Buscados

```
"""
Listado de 20 productos más buscados
"""
#Se empieza con una lista vacía en la cuál en la primera posición tendrá el id del
#producto y se espera que en el siguiente lugar estén las búsquedas por lo que se empieza en 0.

producto_buscado = []

for producto in lifestore_products:
    lista_interna =[producto[0],0]
    producto_buscado.append(lista_interna)

#Ahora con este for, se busca en la lista lifestore_searches el id del producto

for busqueda in lifestore_searches:
    producto_bus = busqueda[1]
    producto_buscado[producto_bus-1][1] +=1

print("Regresa el id y numero de busqueda de los productos top 20 mas buscados")
print(sorted(producto_buscado, key = lambda x: -x[1])[:20])

Regresa el id y numero de busqueda de los productos top 20 mas buscados
[[54, 263], [57, 107], [29, 60], [3, 55], [4, 41], [85, 35], [67, 32], [7, 31], [5, 30], [47, 30], [48, 27], [44, 25], [2, 24], [42,
23], [8, 20], [12, 15], [21, 15], [66, 15], [18, 11], [51, 11]]
```

## c. Listado de 5 Productos con Menores Ventas

Listado por categoría con los 5 productos con menores ventas

"""

*#Se empieza creando una lista donde guardaremos el id del producto y también su categoría*

cat\_ventas = []

for producto in lifestore\_products: *#tenemos id y categoría*

    lista\_interna = [producto[0], producto[3], 0]

    cat\_ventas.append(lista\_interna)

*#Ahora en la lista lifestores\_sales buscamos el id del producto que se encuentra en el segundo elemento de la lista*

for venta in lifestore\_sales:

    cat\_bus = venta[1] *#id\_product*

    cat\_ventas[cat\_bus-1][2] += 1 *#metemos las ventas a la lista anterior y*

*#tenemos id, categoría y ventas*

*#Aquí utilicé ayuda de un diccionario para buscar la categoría de acuerdo a la lista previamente creada de cat\_ventas*

fin\_cat\_ventas = {}

for cat in cat\_ventas:

    if cat[1] in fin\_cat\_ventas: *#de acuerdo la categoría -> cat[2] ventas y si no entonces no +=*

        fin\_cat\_ventas[cat[1]] += cat[2]

    else:

        fin\_cat\_ventas[cat[1]] = cat[2]

print("Regresa la Categoría y numero de venta de los productos(5) menos vendidos")

fin\_cat\_ventas = (list(map(list, fin\_cat\_ventas.items()))) *#lo convertimos a una lista*

print(sorted(fin\_cat\_ventas, key = lambda x: x[1])[:5])

Regresa la Categoría y numero de venta de los productos(5) menos vendidos

[['memorias usb', 1], ['pantallas', 2], ['bocinas', 2], ['audifonos', 5], ['tarjetas de video', 26]]

## d. Listado de 5 Productos con Menores Búsquedas

```
"""
Listado por categoría con los 5 productos con menores búsquedas
"""
#Se empieza creando una lista donde guardaremos el id del producto y también su categoría
cat_buscado = []
for producto in lifestore_products:
    lista_interna = [producto[0], producto[3], 0]
    cat_buscado.append(lista_interna)

#Ahora en la lista lifestores_searches buscamos el
#id del producto que se encuentra en el segundo elemento de la lista y lo agregamos a la
#lista de cat_buscado

for busqueda in lifestore_searches:
    cat_bus = busqueda[1]
    cat_buscado[cat_bus-1][2] += 1 #metemos las búsquedas a la lista anterior y
    #[tenemos id, categoría y búsquedas]

#Aquí utilicé ayuda de un diccionario para buscar la categoría de acuerdo a
#la lista previamente creada de cat_buscado

fin_cat_busqueda = {} #de acuerdo la categoría --> cat[2] búsquedas y si no entonces NO +=
for cat in cat_buscado:
    if cat[1] in fin_cat_busqueda:
        fin_cat_busqueda[cat[1]] += cat[2]
    else:
        fin_cat_busqueda[cat[1]] = cat[2]

print("Regresa categoría y numero de busqueda de los productos (100) menos buscados\n")
fin_cat_busqueda = (list(map(list, fin_cat_busqueda.items())) #lo convertimos en una lista y .items() que devuelve un
#objeto de vista.
print(sorted(fin_cat_busqueda, key = lambda x: x[1])[:5])
```

Regresa categoría y numero de busqueda de los productos (100) menos buscados

[['memorias usb', 0], ['bocinas', 9], ['pantallas', 56], ['audifonos', 64], ['tarjetas de video', 82]]

## e. Listado de 10 productos con las Mejores y Peores Reseñas

```
#####
Listado para 10 productos con las mejores reseñas y peores
#####

#Seccion 2 acomodar por rating
producto_rate = [] #Creamos una lista vacía donde meteremos nuestros resultados
for producto in lifestore_products:
    lista_interna = [producto[0], 0, 0] #Metemos id en el primer elemento
    producto_rate.append(lista_interna) #se mete a la lista antes creada

for rate in lifestore_sales:
    producto_rat = rate[1] #id_pr
    producto_rate[producto_rat-1][1] += rate[2] #score
    producto_rate[producto_rat-1][2] += 1 #pongo 1 en el [2]

fin_product_rate = {} #creamos un diccionario para poder juntarlo, como en los anteriores
for cat in producto_rate: #si rate
    if cat[1] in fin_product_rate: #score
        fin_product_rate[cat[0]] += cat[1] #id en score
    else:
        if cat[2] != 0: #score dif a 0 entonces
            fin_product_rate[str(cat[0])] = cat[1]/cat[2] #score sobre 1
        else: #el id del prod de acuerdo al score
            fin_product_rate[str(cat[0])] = cat[1] #y si es 0 entonces ponle el score
#se tomó en cuenta que no todos tienen siempre el mismo score, porque los califican diferentes personas

fin_product_rate = (list(map(list, fin_product_rate.items())))#ahora convertimos ese diccionario en lista

print("Regresa ID y rating de los productos (20) mas rating\n")
print(sorted(fin_product_rate, key = lambda x: -x[1])[:20])

print("Regresa ID y rating de los productos (20) menor rating\n")
print(sorted(fin_product_rate, key = lambda x: x[1])[:20])
```

Regresa ID y rating de los productos (20) mas rating

```
[['1', 5.0], ['6', 5.0], ['7', 5.0], ['8', 5.0], ['11', 5.0], ['21', 5.0], ['22', 5.0], ['25', 5.0], ['28', 5.0], ['40', 5.0], ['49', 5.0], ['50', 5.0], ['52', 5.0], ['60', 5.0], ['66', 5.0], ['67', 5.0], ['84', 5.0], ['85', 5.0], ['57', 4.866666666666666], ['3', 4.809523809523809]]
```

Regresa ID y rating de los productos (20) menor rating

```
[['9', 0], ['14', 0], ['15', 0], ['16', 0], ['19', 0], ['20', 0], ['23', 0], ['24', 0], ['26', 0], ['27', 0], ['30', 0], ['32', 0], ['34', 0], ['35', 0], ['36', 0], ['37', 0], ['38', 0], ['39', 0], ['41', 0], ['43', 0]]
```



## f. Fecha y venta de los productos (20) Más y Menos vendidos

```
#####
Fecha y venta de los productos (20) más y menos vendidos
#####

#Seccion 3 ventas

producto_ventas = [] #creamos una lista vacía
for producto in lifestore_sales:
    lista_interna = [producto[3], producto[1]] #metemos fecha y el id del producto
    producto_ventas.append(lista_interna)

for venta in lifestore_products:
    producto_vent = venta[0]
    producto_ventas[producto_vent-1][1] += venta[2] #precio de acuerdo al precio

#utilizamos un diccionario de nuevo para juntar la fecha con el precio
fin_product_venta = {}
for vent in producto_ventas:
    if vent[0] in fin_product_venta:
        fin_product_venta[vent[0]] += vent[1]
    else:
        fin_product_venta[vent[0]] = vent[1]

fin_product_venta = (list(map(list, fin_product_venta.items())))

print("*Regresa Fecha y venta de los productos (20) mas vendidos")
print(sorted(fin_product_venta, key = lambda x: -x[1])[:20])

print("*Regresa Fecha y venta de los productos (20) menos vendidos")
print(sorted(fin_product_venta, key = lambda x: x[1])[:20])
```

```
*Regresa Fecha y venta de los productos (20) mas vendidos
[['27/04/2020', 30542], ['08/03/2020', 21083], ['27/02/2020', 20837], ['24/04/2020', 18611], ['25/02/2020', 16528], ['14/04/2020', 16287], ['31/01/2020', 14987], ['21/07/2020', 14004], ['30/03/2020', 12115], ['06/04/2020', 11828], ['20/05/2020', 11512], ['05/05/2020', 10888], ['09/01/2020', 10773], ['14/05/2020', 9774], ['11/01/2020', 8536], ['10/11/2019', 8441], ['13/04/2020', 8070], ['24/03/2020', 7917], ['02/05/2020', 7485], ['11/06/2020', 6714]]
*Regresa Fecha y venta de los productos (20) menos vendidos
[['31/03/2020', 7], ['16/02/2020', 8], ['20/01/2020', 8], ['05/03/2020', 11], ['25/03/2020', 12], ['06/03/2020', 12], ['04/03/2020', 12], ['05/09/2020', 17], ['02/02/2020', 18], ['01/02/2020', 18], ['30/06/2020', 29], ['04/04/2020', 29], ['26/02/2020', 29], ['09/02/2020', 29], ['06/02/2020', 29], ['26/01/2020', 29], ['10/01/2020', 29], ['20/04/2020', 30], ['28/03/2020', 32], ['24/05/2020', 40]]
```

## g. Menú para ver los resultados Anteriores como Resumen

```
###Menú

#Como resumen para quien vaya a revisarlo puede usar esta parte
print( 'Hola, aquí podrás ver la información resumida. A continuación está el menú de opciones.')

print("\n\n")

print( 'Opciones: 1. Productos Más Vendidos, 2. Productos Más Buscados, 3.Productos con Menores Ventas, 4.Productos con Menores Búsquedas')
print( '5. Productos con las mejores reseñas, 6. Productos con las peores, 7.Fecha y venta de los productos más vendidos, 8.Fecha y venta de los productos menos vendidos')

print("\n\n")

#En el input pondrá la opción que quiera revisar de las opciones dadas
pregunta = input('Ingresa solicitud: ')

pregunta = int(pregunta)

#Aquí están todas las opciones necesarias de acuerdo a este documento y se pueden revisar 10 veces por lo que el código está repedito esas veces.
if pregunta == 1:
    print(sorted(producto_ventas_retorno, key = lambda x: -x[1])[:15])
elif pregunta ==2:
    print(sorted(producto_buscado, key = lambda x: -x[1])[:20])
elif pregunta ==3:
    print(sorted(fin_cat_ventas, key = lambda x: x[1])[:5])
elif pregunta==4:
    print(sorted(fin_cat_busqueda, key = lambda x: x[1])[:5])
elif pregunta==5:
    print(sorted(fin_product_rate, key = lambda x: -x[1])[:20])
elif pregunta==6:
    print(sorted(fin_product_rate, key = lambda x: x[1])[:20])
elif pregunta ==7:
    print(sorted(fin_product_venta, key = lambda x: -x[1])[:20])
elif pregunta ==8:
    print(sorted(fin_product_venta, key = lambda x: x[1])[:20])
elif pregunta >8:
    print('Lo sentimos, su opción no es válida ')

print("\n\n")
```

Hola, aquí podrás ver la información resumida. A continuación está el menú de opciones.

Opciones: 1. Productos Más Vendidos, 2. Productos Más Buscados, 3.Productos con Menores Ventas, 4.Productos con Menores Búsquedas  
5. Productos con las mejores reseñas, 6. Productos con las peores, 7.Fecha y venta de los productos más vendidos, 8.Fecha y venta de los productos menos vendidos

Ingresa solicitud: 1  
[[54, 49], [3, 42], [5, 20], [42, 18], [57, 15], [4, 13], [29, 13], [2, 12], [47, 11], [12, 9], [48, 9], [7, 7], [44, 6], [18, 5], [8, 4]]

Ingresa solicitud: 2  
[[54, 263], [57, 107], [29, 60], [3, 55], [4, 41], [85, 35], [67, 32], [7, 31], [5, 30], [47, 30], [48, 27], [44, 25], [2, 24], [42, 23], [8, 20], [12, 15], [21, 15], [66, 15], [18, 11], [51, 11]]

Ingresa solicitud: 3  
[['memorias usb', 1], ['pantallas', 2], ['bocinas', 2], ['audifonos', 5], ['tarjetas de video', 26]]

Ingresa solicitud: 4  
[['memorias usb', 0], ['bocinas', 9], ['pantallas', 56], ['audifonos', 64], ['tarjetas de video', 82]]

Ingresa solicitud: 5  
[['1', 5.0], ['6', 5.0], ['7', 5.0], ['8', 5.0], ['11', 5.0], ['21', 5.0], ['22', 5.0], ['25', 5.0], ['28', 5.0], ['40', 5.0], ['49', 5.0], ['50', 5.0], ['52', 5.0], ['60', 5.0], ['66', 5.0], ['67', 5.0], ['84', 5.0], ['85', 5.0], ['57', 4.866666666666667], ['3', 4.809523809523809]]

Ingresa solicitud: 6  
[['9', 0], ['14', 0], ['15', 0], ['16', 0], ['19', 0], ['20', 0], ['23', 0], ['24', 0], ['26', 0], ['27', 0], ['30', 0], ['32', 0], ['34', 0], ['35', 0], ['36', 0], ['37', 0], ['38', 0], ['39', 0], ['41', 0], ['43', 0]]  
Cut the selected cells

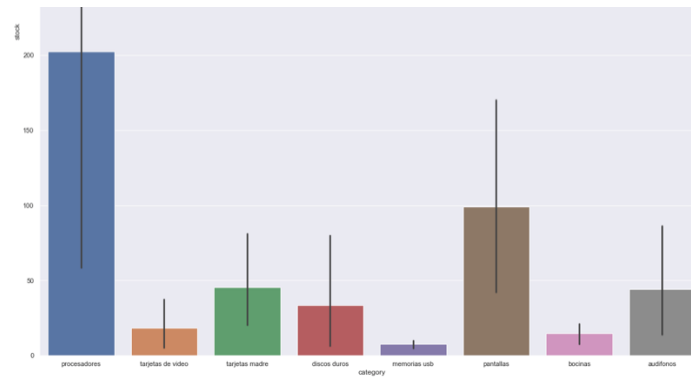
Ingresa solicitud: 7  
[['27/04/2020', 30542], ['08/03/2020', 21003], ['27/02/2020', 20037], ['24/04/2020', 18611], ['25/02/2020', 16520], ['14/04/2020', 16207], ['31/01/2020', 14907], ['21/07/2020', 14004], ['30/03/2020', 12115], ['06/04/2020', 11828], ['20/05/2020', 11512], ['05/05/2020', 10888], ['09/01/2020', 10773], ['14/05/2020', 9774], ['11/01/2020', 8536], ['10/11/2019', 8441], ['13/04/2020', 8070], ['24/03/2020', 7917], ['02/05/2020', 7485], ['11/06/2020', 6714]]

Ingresa solicitud: 8  
[['31/03/2020', 7], ['16/02/2020', 0], ['20/01/2020', 0], ['05/03/2020', 11], ['25/03/2020', 12], ['06/03/2020', 12], ['04/03/2020', 12], ['05/09/2020', 17], ['02/02/2020', 18], ['01/02/2020', 18], ['30/06/2020', 29], ['04/04/2020', 29], ['26/02/2020', 29], ['09/02/2020', 29], ['06/02/2020', 29], ['26/01/2020', 29], ['10/01/2020', 29], ['20/04/2020', 30], ['20/03/2020', 32], ['24/05/2020', 40]]

Ingresa solicitud: 9  
Lo sentimos, su opción no es válida

### 3. SOLUCIÓN AL PROBLEMA

Al observar los datos la recomendación de retirar un producto del mercado serían las pantallas ya que al observar la lista de productos más vendidos, estas son las que menos figuran en ella, y al observar en el stock, que podrán observar a continuación (*figura 1 en café*), en realidad tienen bastante comparado con los discos duros, las tarjetas madres y de video que venden más.



(figura 1)

Esta decisión se tomó ya que al quedarse en stock, las pantallas, además de que tienen bajas ventas están gastando en almacenaje.

Ya pasado por la categoría, retiraría por producto, aquellos productos que tienen menor rating ya que eso afecta a la categoría completa, empezaría por aquellos que tienen score 0 como "Procesador Intel Core i3-8100, S-1151, 3.60GHz, Quad-Core, 6MB Smart Cache (8va. Generación - Coffee Lake)", "Tarjeta de Video EVGA NVIDIA GeForce GT 710, 2GB 64-bit GDDR3, PCI Express 2.0" y " Tarjeta de Video EVGA NVIDIA GeForce GTX 1660 Ti SC Ultra Gaming, 6GB 192-bit GDDR6, PCI 3.0" y demás productos que demuestran un score en 0 para de esa forma reducir inventarios y aumentar en aquellos que tienen un score de 5 y más ventas como el disco duro: "SSD Kingston A400, 120GB, SATA III, 2.5", 7mm.

### 4. REFERENCIAS

- Creación de diccionario a partir de dos listas de datos | Interactive Chaos. (s. f.). Interactive Chaos. Recuperado 13 de septiembre de 2021, de <https://interactivechaos.com/es/forum/topic/creacion-de-diccionario-partir-de-dos-listas-de-datos>
- Diccionario en python, con ejemplos. (2017, 2 junio). Jarroba. <https://jarroba.com/diccionario-python-ejemplos/#:%7E:text=Los%20diccionarios%20en%20Python%20son,elementos%20con%20una%20misma%20clave>).
- Emtech, Fundamentos de Programación con Python, <https://escuela.emtech.digital/enrollments>
- Guzman, H. C. (s. f.). Función map() | Curso de Python | Hektor Profe. Hektor Docs. Recuperado 13 de septiembre de 2021, de <https://docs.hektorprofe.net/python/funcionalidades-avanzadas/funcion-map/>
- Python Dictionary items() Method. (s. f.). Schools. Recuperado 13 de septiembre de 2021, de [https://www.w3schools.com/python/ref\\_dictionary\\_items.asp](https://www.w3schools.com/python/ref_dictionary_items.asp)
- Python - Ejercicio 390: Combinar los Elementos de una Lista de Diccionarios con la Clase Counter. (2019, 22 diciembre). [Vídeo]. YouTube. <https://www.youtube.com/watch?v=fMarS2IHqQ8>
- **Tutorías de Jaime Saúl Alonso Sánchez**, Licenciado en Ingeniería en Sistemas Computacionales de la Universidad de Guanajuato.
- Shaik, H. K. (2020, 4 noviembre). Convertir lista en diccionario en Python. Geekflare. <https://geekflare.com/es/python-convert-list-to-dictionary/>