

Emerging Technologies Institute  
EMTECH

Antonio Calzadilla

Proyecto 1. Introducción a Phyton

A 14 de septiembre del 2021

## Índice

Introducción.	.	.	.	.	.	.	.	.	3
Definición del Código	.	.	.	.	.	.	.	.	4
Solución al problema	.	.	.	.	.	.	.	.	7
Conclusión	.	.	.	.	.	.	.	.	9

## Introducción

Conocer los conceptos que integran las bases de Python es el punto esencial en el aprendizaje de este lenguaje de programación, pero para aprovechar totalmente su uso es indispensable también conocer su aplicación y alcance a fin de utilizarlo en nuestras actividades laborales.

En el presente proyecto se definen diferentes actividades a realizar, algunas de estas actividades fueron finalizadas en tanto que otras no fueron muy claras la forma de proceder por lo tanto no fue posible llegar a los objetivos propuestos, sin embargo, considero que se logro un buen entendimiento en el uso de las listas y sus funciones básica.

En los siguientes capítulos se revisará punto por punto como se fue desarrollando el proyecto hasta identificar áreas de oportunidad y logros en las conclusiones.

## Definición del código

### Listado de Productos más vendidos

A continuación, se muestra el código utilizado para generar el listado de los productos mas vendidos ordenados de mayor a menor para los primeros 15 productos. En color verde se explica el uso de cada una de las partes del código, detallando en forma muy general cual es el objetivo de cada parte.

```
[1] #mas vendido producto_venta
#En este punto se crearon dos lista a partir de lifestore_products, de la primer lista se sustrajo el primer elemento en la fila con el índice 0 a fin de mostrar el numero de articulo.
producto_ventas = []

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

[[1, 0], [2, 0], [3, 0], [4, 0], [5, 0], [6, 0], [7, 0], [8, 0], [9, 0], [10, 0], [11, 0], [12, 0], [13, 0], [14, 0], [15, 0], [16, 0], [17, 0], [18, 0], [19, 0], [20, 0], [21, 0], [22, 0], [23, 0], [24, 0], [25, 0], [26, 0], [27, 0], [28, 0], [29, 0], [30, 0], [31, 0], [32, 0], [33, 0], [34, 0], [35, 0], [36, 0], [37, 0], [38, 0], [39, 0], [40, 0], [41, 0], [42, 0], [43, 0], [44, 0], [45, 0], [46, 0], [47, 0], [48, 0], [49, 0], [50, 0], [51, 0], [52, 0], [53, 0], [54, 0], [55, 0], [56, 0], [57, 0], [58, 0], [59, 0], [60, 0], [61, 0], [62, 0], [63, 0], [64, 0], [65, 0], [66, 0], [67, 0], [68, 0], [69, 0], [70, 0], [71, 0], [72, 0], [73, 0], [74, 0], [75, 0], [76, 0], [77, 0], [78, 0], [79, 0], [80, 0], [81, 0], [82, 0], [83, 0], [84, 0], [85, 0], [86, 0], [87, 0], [88, 0], [89, 0], [90, 0], [91, 0], [92, 0], [93, 0], [94, 0], [95, 0], [96, 0], [97, 0], [98, 0], [99, 0], [100, 0]]

[3] # mas vendidos
#En este punto se creo una lista llamada ventas a partir de lifestore_sales, esta lista se busco empatarla con la creo se genero en el punto previo a fin de tener el numero de articulo y las ventas por cada articulo.
for ventas in lifestore_sales:
    lista_ventas = ventas[1]
    producto_ventas[lista_ventas - 1][1] += 1
print(producto_ventas)

[[1, 2], [2, 13], [3, 42], [4, 13], [5, 20], [6, 3], [7, 7], [8, 4], [9, 0], [10, 1], [11, 3], [12, 9], [13, 1], [14, 0], [15, 0], [16, 0], [17, 1], [18, 5], [19, 0], [20, 0], [21, 2], [22, 1], [23, 0], [24, 0], [25, 2], [26, 0], [27, 0], [28, 0], [29, 0], [30, 0], [31, 0], [32, 0], [33, 0], [34, 0], [35, 0], [36, 0], [37, 0], [38, 0], [39, 0], [40, 0], [41, 0], [42, 0], [43, 0], [44, 0], [45, 0], [46, 0], [47, 0], [48, 0], [49, 0], [50, 0], [51, 0], [52, 0], [53, 0], [54, 0], [55, 0], [56, 0], [57, 0], [58, 0], [59, 0], [60, 0], [61, 0], [62, 0], [63, 0], [64, 0], [65, 0], [66, 0], [67, 0], [68, 0], [69, 0], [70, 0], [71, 0], [72, 0], [73, 0], [74, 0], [75, 0], [76, 0], [77, 0], [78, 0], [79, 0], [80, 0], [81, 0], [82, 0], [83, 0], [84, 0], [85, 0], [86, 0], [87, 0], [88, 0], [89, 0], [90, 0], [91, 0], [92, 0], [93, 0], [94, 0], [95, 0], [96, 0], [97, 0], [98, 0], [99, 0], [100, 0]]

[4] #funcion de SORT
#Para ordenar de mayor a menor los productos mas vendidos se utilizo la funcion SORT adicionada con el reverse "True" y el key "segundo elemento", siendo este el que define el orden que usara la funcion SORT.
def segundo_elemento(l):
    return l[1]
producto_ventas.sort(reverse=True, key=segundo_elemento)

#producto mas vendido
#Para presentar en forma vertical la lista de productos se uso la funcion mover, de esta forma se presenta un listado ordenado en base al elemento dos, en este caso se solicito presentar unicamente los primeros 15 articulos.
mas_vendido = producto_ventas[:15]
for mv in mas_vendido:
    print(mv)

[54, 50]
[3, 42]
[5, 20]
[42, 18]
[57, 15]
[29, 14]
[2, 13]
[4, 13]
[47, 11]
[12, 9]
[48, 9]
[7, 7]
[31, 6]
[44, 6]
```

## Listado de productos más buscados

A continuación, se muestra el código utilizado para generar el listado de los productos más buscados ordenados de mayor a menor para los primeros 20 productos. En color verde se explica el uso de cada una de las partes del código, detallando en forma muy general cual es el objetivo de cada parte.

```
1 #Producto mas buscado
#En este punto se crearon dos lista a partir de lifestore_products, de la primer lista se sustrajo el primer elemento en la fila con el indice 0 a fin de mostrar el numero de articulo.
producto_ventasb = []

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

[[1, 0], [2, 0], [3, 0], [4, 0], [5, 0], [6, 0], [7, 0], [8, 0], [9, 0], [10, 0], [11, 0], [12, 0], [13, 0], [14, 0], [15, 0], [16, 0], [17, 0], [18, 0], [19, 0], [20, 0], [21, 0], [22, 0], [23, 0], [24, 0], [25, 0], [26, 0], [27, 0],

7 # busquedas por producto
#En este punto se creo una lista llamada busquedas a partir de lifestore_searches, esta lista se busco empararla con la que se genero en el punto previo a fin de tener el numero de articulo y las busquedas por cada articulo.
for busquedas in lifestore_searches:
    lista_busquedas = busquedas[1]
    producto_ventasb[lista_busquedas - 1][1] += 1
print(producto_ventasb)

[[1, 10], [2, 24], [3, 55], [4, 41], [5, 30], [6, 10], [7, 31], [8, 20], [9, 1], [10, 1], [11, 5], [12, 15], [13, 2], [14, 0], [15, 4], [16, 0], [17, 3], [18, 11], [19, 0], [20, 0], [21, 15], [22, 5], [23, 0], [24, 0], [25, 10], [26, 5],

8 #funcion de SORT
#Para ordenar de mayor a menor los productos mas buscado se utilizo la funcion SORT adicionada con el reverse "True" y el key "segundo elemento", siendo este el que define el orden que usara la funcion SORT.
def segundo_elemento(l):
    return l[1]
producto_ventasb.sort(reverse=True, key=segundo_elemento)

#producto mas buscado
#Para presentar en forma vertical la lista de productos se uso la funcion mover, de esta forma se presenta un listado ordenado en base al elemento dos, en este caso se solicito presentar unicamente los primeros 20 articulos.
mas_buscado = producto_ventasb[:20]
for mv in mas_buscado:
    print(mv)

[54, 263]
[57, 107]
[29, 60]
[3, 55]
[4, 41]
[85, 35]
[67, 33]
[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]
```

## Listado de productos con peores y mejores reseñas

A continuación, se muestra el código utilizado para generar el listado de los productos con más y menos devoluciones lo cual nos permitirá identificar los que tienen mejores y peores reseñas ordenados de mayor a menor para los primeros 10 productos con más y menos devoluciones. En color verde se explica el uso de cada una de las partes del código, detallando en forma muy general cual es el objetivo de cada parte.

<pre>[10] #mejores resenas version II #En este punto se crearon dos lista a partir de lifestore_products, de la primer producto_ventasd = []  for producto in lifestore_products:     lista_interna = [producto[0], 0]     producto_ventasd.append(lista_interna)</pre>	<p>lista se sustrajo el primer elemento en la fila con el indice 0 a fin de mostrar el numero de articulo.</p>
<pre>• #En este punto se creo una lista llamada devolucion a partir de lifestore_sales, for devolucion in lifestore_sales:     producto_devuelto = devolucion[1]     devuelto=devolucion[4]     if devuelto==1:         validez=True     else:         validez=False     if validez== True:         producto_ventasd[producto_devuelto - 1][1] += 1</pre>	<p>esta lista se busco emparelarla con la que se genero en el punto previo a fin de tener el numero de articulo y las devoluciones por cada articulo.</p>
<pre>• #mejores resenas version II #En este punto se crearon dos lista a partir de lifestore_products, de la primer devolucion=(     0 ) for relacion in producto_ventasd:     devolucion += relacion[1] print(relacion) print(devolucion)</pre>	<p>lista se sustrajo el primer elemento en la fila con el indice 0 a fin de mostrar el numero de articulo.</p>
<pre>[36, 0] [37, 0] [38, 0] [39, 0] [40, 0] [41, 0] [42, 0] [43, 0] [44, 0] [45, 1] [46, 1]</pre>	
<pre>[13] #funcion de SORT #Para ordenar de mayor a menor los productos mas devueltos se utilizo la funcion def segundo_elemento(l):     return l[1] producto_ventasd.sort(reverse=True, key=segundo_elemento)</pre>	<p>SORT adicionada con el reverse "True" y el key "segundo elemento", siendo este el que define el orden que usara la funcion SORT.</p>
<pre>[14] #producto mas buscado #Para presentar en forma vertical la lista de productos se uso la funcion mover, menos_devuelto = producto_ventasd[10:] for mv in menos_devuelto:     print(mv)</pre>	<p>de esta forma se presenta un listado ordenado en base al elemento dos, en este caso se solicito presentar unicamente los primeros 10 articulos con mayores devoluciones.</p>
<pre>[31, 1] [2, 1] [17, 1] [29, 1] [45, 1] [46, 1] [54, 1] [1, 0] [3, 0] [4, 0]</pre>	
<pre>[16] #funcion de SORT #Para ordenar de menor a mayor los productos mas devueltos se utilizo la funcion def segundo_elemento(l):     return l[1] producto_ventasd.sort(reverse=False, key=segundo_elemento)</pre>	<p>SORT adicionada con el reverse "False" y el key "segundo elemento", siendo este el que define el orden que usara la funcion SORT.</p>
<pre>[17] #producto mas buscado #Para presentar en forma vertical la lista de productos se uso la funcion mover, menos_devuelto = producto_ventasd[10:] for mv in menos_devuelto:     print(mv)</pre>	<p>de esta forma se presenta un listado ordenado en base al elemento dos, en este caso se solicito presentar unicamente los primeros 10 articulos con menores devoluciones.</p>
<pre>[1, 0] [3, 0] [4, 0] [5, 0] [6, 0] [7, 0] [8, 0] [9, 0] [10, 0] [11, 0]</pre>	

Para el punto de productos con menores ventas y productos con menores búsquedas por categoría se considero filtrar los productos que caían en determinada categoría y después contar las veces que aparecía cada producto en la fila de ventas y en la fila de búsquedas, sin embargo no se encontró la forma de unir la fila recientemente creada con la lista de `lifestore_searches` y `lifestore_sales` por lo cual no fue posible completar esta actividad.

```
# Se crea una lista que contenga el tercer elemento correspondiente a categorias
categorias=[]
for prod in lifestore_products:
    if prod[3] not in categorias:
        categorias.append(prod[3])
    print(categorias)
for categoria in categorias:
    lista_interna = [categoria[1], 0]
    categorias.append(lista_interna)
    print(producto_categorias)

[ ] #procesadores como categoria
producto_venta = []

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

[ ] #Aquí se filtro unicamente los productos cuya categoria fuera procesadores, lo malo de esta tecnica es que habria que estar generando código para cada una de las categorias lo cual consume mucho tiempo.
for categoria in lifestore_products:
    producto_procesadores = categoria[0]
    procesadores=categoria[3]
    if procesadores=='procesadores':
        validez=True
    else:
        validez=False
    if validez== True:
        producto_venta[producto_procesadores - 1][1] += 1

#En este punto se muestran los productos que son procesadores identificandose con un 1 en la segunda fila de la lista
sipprocesadores=(
    0
)
for relacion in producto_venta:
    sipprocesadores += relacion[1]
    print(relacion)
    print(sipprocesadores)

[1, 1]
[2, 1]
[3, 1]
[4, 1]
[5, 1]
[6, 1]
[7, 1]
[8, 1]
[9, 1]
```

## Solución al problema

Para disminuir el inventario acumulado y aumentar la búsqueda de sus productos lo cual se traduzca en un mayor volumen de ventas considero viable en primer lugar identificar el inventario en existencia el cual no logre reconocer en la base de datos ya que no se muestra un inventario inicial y/o un inventario final por lo tanto no es posible conocer la rotación de sus productos. El punto de devoluciones puede ayudar a complementar la rotación, pero antes que eso se tiene que revisar todos los puntos que no tuvieron ninguna venta y por lo tanto no tuvieron ninguna devolución.

Una vez que se identificó la rotación de inventarios es conveniente analizar las ventas para identificar los que tienen mayor venta y que según la investigación son los siguientes:

```
[54, 50]  
[3, 42]  
[5, 20]  
[42, 18]  
[57, 15]  
[29, 14]  
[2, 13]  
[4, 13]  
[47, 11]  
[12, 9]  
[48, 9]  
[7, 7]  
[31, 6]  
[44, 6]  
[18, 5]
```

Considerando los niveles de venta y teniendo los niveles de inventario sería posible determinar una razón de volumen de ventas de inventario lo cual nos ayudaría identificar aquellos productos que tuvieron mas o menos ventas tomando como punto de comparación los niveles de inventario a fin de saber cuales están teniendo una rotación lenta y que por lo tanto no es conveniente mantener en el almacén.

Con la lista de ventas máximas es posible observar que la mayoría de los productos con mayor nivel de ventas corresponden a procesadores y discos duros por lo tanto en estas categorías es donde la empresa podría buscar incrementar su inversión, ya que estos representan la mayor parte de sus ventas.

Sobre la ultima pregunta de la estrategia de productos, considero recomendable retirar las tarjetas de video, las tarjetas madre, los audífonos y las bocinas ya que son las que menos ventas tienen y aquí no importa si sus niveles de devolución son altos o bajos, estos productos simplemente no se venden, por lo tanto no es recomendable invertir en estos productos y lo que debe hacer la empresa es buscar disminuir a la brevedad el inventario existente en estos productos.



## Conclusión

Considero que no es posible dar una sugerencia con la información proporcionada no solo por que es insuficiente sino también porque no se logro completar cada uno de los puntos solicitados. Sin embargo, en forma general se puede observar que algunas categorías tuvieron más movimiento en el volumen de ventas mientras que otras no tuvieron ningún movimiento por lo que considero que este puede ser un punto de partida para generar ideas de una posible estrategia, así como se muestra en el punto de solución del problema.

Me hubiera gustado tener más casos prácticos durante el primer parte del curso para saber como obtener cada uno de los requerimientos de la practica final. Desafortunadamente me sentí perdido después de ver los videos y completar el 90% del curso y no saber que hacer con la práctica. Las tutorías fueron de gran ayuda para tener puntos de referencia, pero considero que hizo falta más ejercicios de cómo lidiar con situaciones como las que presentaba la práctica.