

PROYECTO 1. INTRODUCCIÓN A PYTHON (LIFE – STORE)



EMTECH
Emerging Technologies Institute

Yonathan Andoni Martinez Medrano Ing. Mecatrónico Miércoles 1 de septiembre Del 2021

G2- Data Science Programación de ciencias de datos con Python

Contenido

Introducción.....	2
Objetivos.....	2
Definición del código.....	3
Puntos a realizar.....	3
Punto 1°	3
Generar un listado de los 50 productos con mas ventas.	4
Generar un listado de los 100 productos con mas búsquedas.....	5
Por categoría, generar un listado de los 50 productos con menos ventas.	5
Por categoría, generar un listado con los 100 productos con menos búsquedas.....	6
Punto 2°	6
Lista de 20 productos con mejor reseña, considerando los productos con devolución.	7
Lista de 20 productos con peor reseña, considerando los productos con devolución.	7
Punto 3°	8
Total de ingresos.....	8
Ventas promedio mensuales.....	8
Total anual.....	8
Meses con más ventas al año.....	8
Solución al problema	10
Interfaz grafica.....	11
Administrador.....	11
Login.....	11
Registro.....	12
Informacion.....	13
Descripción del programa de la interfaz grafica	15
Conclusión.....	20
Conclusión general.....	20
Conclusiones individuales.....	20

Introducción

En este proyecto final hare un código para un programa que dé solución a los puntos solicitados en el proyecto final de la primera parte de este programa que tengo el honor de ser parte. Dentro de este reporte se hará el análisis de cada punto a solucionar para tener una amplia vista del problema general y así poderle dar una solución más apropiada. Además, como punto importante haremos uso de las funciones vistas en el curso de la primera parte de EmTech para que sea productivo y de un uso apropiado a este recurso valiosos que me brindo Santander, así como herramientas avanzadas tomadas en otros cursos y de mi carrera para darle un estilo único a mi proyecto final, las cuales también explicare como es que se usaron dentro del programa como en la interfaz gráfica que propiamente realice desde cero.

Objetivos

- Diseñar un programa funcional cumpliendo todos los requisitos propuestos en este proyecto final.
- Hacer uso de las listas, operadores booleanos y/o lógicos, así como operadores aritméticos en el lenguaje de programación Python para la realización del programa.
- Hacer un análisis sistemático de los productos que menos se venden para que el usuario que vea la información resumida vea en orden de ventas y de ganancias como conclusión final decida la opción mas correcta para del valor de productos vendidos con el total de ganancias que se hicieron de esa categoría de productos.
- Por último, realizar una interfaz gráfica donde sea cómodo visualizar los datos resumidos, así como realizar un Login de inicio para que sea mas productivo el uso de Python como lenguaje para el análisis de estos datos.

Definición del código.

Objetivo: Código explicado describiendo sus procesos y variables.

Nota: antes de dar la explicación de mi código, tengo que hacer hincapié a que las indicaciones de los 3 puntos a realizar en el proyecto final como los datos que nos dan para darles solución tienen un error...

1. El primer error es que a continuación en los puntos podrán observar que se piden los 100 productos... los 50 productos... de tal a tal.
Los cuales no se podrán visualizar en la interfaz grafica ya que como se menciono al tutor nos dan un listado de productos en los cuales ni si quiera llega a los 100 productos que vende esta empresa llamada Life Store.
2. Como segundo error al hacer el análisis y filtros necesarios para sacar lo que piden los siguientes puntos se observara que no se llegara a lo que pide en cuestión de lo que se solicita, es decir, si se pide que hagamos un listado de los 50 productos con mayores ventas, al hacer el análisis y filtro de estos datos se observara que no llegara ni si quiera a la mitad hablando en general de los puntos que a continuación mostrare.

Puntos a realizar

Punto 1°

1

Productos más vendidos y productos rezagados:

- ⌚ Generar un listado de los 50 productos con mayores ventas y uno con los 100 productos con mayor búsquedas.
- ⌚ Por categoría, generar un listado con los 50 productos con menores ventas y uno con los 100 productos con menores búsquedas.

En el punto 1, se encuentran dos viñetas las cuales tienen dos requerimientos en cada una de ellas, es decir, el punto 1 tiene 4 puntos secundarios a realizar, los cuales son:

1. Generar un listado de los 50 productos con mas ventas.
2. Generar un listado de los 100 productos con mas búsquedas.
3. Por categoría, generar un listado de los 50 productos con menos ventas.
4. Por categoría, generar un listado con los 100 productos con menos búsquedas.

Como dicho al principio, después del primer error, menciono que hay ciertos errores, los cuales me puse a pensar a fondo y mejorar mi pensamiento en el análisis de datos y me puse a pensar de que tenia que hacer un código que sirviera a futuro, es decir, si la empresa llega a meter mas productos, el programa hará la misma secuencia y si llega a cambiar las ventas y el total de productos no importara porque hará el mismo proceso y arrojará otros valores dependiendo de su análisis...

Generar un listado de los 50 productos con mas ventas.

Para la solución para este punto y los demás es sencillo si lo vemos desde un punto de vista más amplio, es decir, tenemos un listado de productos vendidos, en los cuales, por obvias razones, también se tienen los reembolsos, los cuales se identifican con un “1” cuando se devuelven y un “0” cuando son vendidos, a continuación, daré la explicación a la solución de este punto.

- Con un ciclo “for” sacaremos los productos vendidos con la condición de que su identificación de que no es un reembolso sea un “0” y si cumple con ese requisito lo añado a una lista.

```
lista_productos_sales=[]
for i in range(0,len(lifestore_sales),1):
    if lifestore_sales[i][4] == 0:
        lista_productos_sales.append(lifestore_sales[i][1])
```

- En lo personal, hare un listado de los productos mas vendidos pero de mayor a menor, para eso tenemos que sacar los productos que no se repiten, es decir, si un producto se compra 80 veces, yo solo lo pondré 1 vez para que no se repita en mi lista de productos no repetidos y hacer un conteo de lo productos en otro ciclo for para tener: el producto “x” se vendió “y” veces pero en un orden de “y” veces siendo de mayor a menor en base a “y”, dicho esto es hacer simplemente otro ciclo for con un “if” para que se vaya añadiendo los productos que no se repiten.

```
lista_productos_sales_Norepetidos=[]
for i in range(0,len(lista_productos_sales),1):
    if lista_productos_sales[i]!=lista_productos_sales[i-1]:
        lista_productos_sales_Norepetidos.append(lista_productos_sales[i])
```

- Ya solo queda hacer un conteo de los productos de la lista de ventas con respecto a los productos que no se repiten, los guardamos en otra lista, después los organizamos en una lista el producto y las veces que se compró en otra lista y un plus que yo le doy es que deben de comprarse mínimo 4 veces el producto para entrar en otra lista de filtro de productos más vendidos.

```
lista_productos_sales_conteo=[]
for i in range(0,len(lista_productos_sales_Norepetidos),1):
    lista_productos_sales_conteo.append(lista_productos_sales.count(lista_productos_sales_Norepetidos[i]))

lista_Norepetidos_sales_conteo=[]
for i in range(0,len(lista_productos_sales_Norepetidos),1):
    lista_Norepetidos_sales_conteo.append([lista_productos_sales_Norepetidos[i],lista_productos_sales_conteo[i]])

lista_mayor_a_sales=[]
for i in range(0,len(lista_Norepetidos_sales_conteo),1):
    if lista_Norepetidos_sales_conteo[i][1]>=4:
        lista_mayor_a_sales.append(lista_Norepetidos_sales_conteo[i])
```

- Ya para finalizar organizamos los productos mas vendidos con la función lambda y volviendo a meter los mismos valores, pero ya organizados de mayores ventas a menor ventas.

```
lista_mayor_a_sales.sort(key=lambda x:x[1],reverse=True)
```

```
lista_top50_mayores_ventas=[]
for i in range(0,len(lista_mayor_a_sales),1):
    for j in range(0,len(lifestore_products),1):
        if lista_mayor_a_sales[i][0] == lifestore_products[j][0]:
            lista_top50_mayores_ventas.append(["El producto es: "+str(lifestore_products[j][1])+" y tuvo: "+str(lista_mayor_a_sales[i][1])+" ventas"])
---
```

Generar un listado de los 100 productos con mas búsquedas.

Para este sub – punto es igual al anterior, sacamos los productos mas buscados y los metemos a una lista, procedemos a sacar de esa lista los productos que no se repiten para meterlos en otra lista y volver a hacer otra lista del conteo con respecto a los que no se repiten con la lista original de los productos mas buscados y volver a meterlos a otra lista, continuando, volvemos a hacer otra lista en la cual vamos a poner el producto y las veces que se buscó poniendo como condición que se tiene que buscar como mínimo 4 veces para que entre en el top de los 100 productos mas buscados, procedemos a organizar de mayor a menor búsqueda con la función lambda y simplemente hacemos un último ciclo for por estética en la presentación del producto.

```
lista_productos_searches=[]
for i in range(0,len(lifestore_searches),1):
    lista_productos_searches.append(lifestore_searches[i][1])

lista_productos_searches_Norepetidos=[]
for i in range(0,len(lista_productos_searches),1):
    if lista_productos_searches [i] != lista_productos_searches[i-1]:
        lista_productos_searches_Norepetidos.append(lista_productos_searches [i])

lista_productos_searches_conteo=[]
for i in range(0,len(lista_productos_searches_Norepetidos),1):
    lista_productos_searches_conteo.append(lista_productos_searches.count(lista_productos_searches_Norepetidos[i]))

lista_Norepetidos_searches_conteo=[]
for i in range(0,len(lista_productos_searches_Norepetidos),1):
    lista_Norepetidos_searches_conteo.append([lista_productos_searches_Norepetidos[i],lista_productos_searches_conteo[i]])

lista_mayor_a_searches=[]
for i in range(0,len(lista_Norepetidos_searches_conteo),1):
    if lista_Norepetidos_searches_conteo[i][1]>=4:
        lista_mayor_a_searches.append(lista_Norepetidos_searches_conteo[i])

lista_mayor_a_searches.sort(key=lambda x:x[1],reverse=True)

lista_top100_mayores_buscados=[]
for i in range(0,len(lista_mayor_a_searches),1):
    for j in range(0,len(lifestore_products),1):
        if lista_mayor_a_searches[i][0] == lifestore_products[j][0]:
            lista_top100_mayores_buscados.append(["El producto es: "+str(lifestore_products[j][1])+" Y se busco: "+str(lista_mayor_a_searches[i][1])+" veces"])
    """
```

Por categoría, generar un listado de los 50 productos con menos ventas.

La forma en como realice el proyecto es de la misma forma que con el primer punto, por eso es que lo explico detalladamente y el resto como es igual y para hacerlo más fácil de leer simplemente doy una explicación rápida de lo que se hace en el código como en esta ocasión:

- Hacemos una lista de los productos vendidos con la condición de que no se devuelvan.
- Hacemos otra lista con la lista anterior de los productos que no se repiten.
- Hacemos otra lista con la lista que no se repite, pero ahora contando los productos que se vendieron con cada producto que no se repite.
- Hacemos otra lista uniendo el producto y las veces que se vendió y dentro con la condición de que debe de ser menor a 5 ventas para entrar en esta lista.
- Usamos la función lambda para organizar de menor a mayor.
- Hacemos otra lista ya filtrada con la anterior y organizada de menor a mayor con la función lambda y poniéndola ya con el producto en nombre y las veces que se vendió en orden de menor a mayor con la condición de que menor sea =1 y mayor menor o igual sea mayor = 5 y que en la interfaz gráfica se vea mucho mejor.

```

lista_productos_menos_ventas=[]
for i in range(0,len(lifestore_sales),1):
    if lifestore_sales[i][4] == 0:
        lista_productos_menos_ventas.append(lifestore_sales[i][1])

lista_productos_menos_ventas_Norepetidos=[]
for i in range(0,len(lista_productos_menos_ventas),1):
    if lista_productos_menos_ventas[i] != lista_productos_menos_ventas[i-1]:
        lista_productos_menos_ventas_Norepetidos.append(lista_productos_menos_ventas[i])

lista_productos_menos_ventas_conteo=[]
for i in range(0,len(lista_productos_menos_ventas_Norepetidos),1):
    lista_productos_menos_ventas_conteo.append(lista_productos_menos_ventas_Norepetidos[i])

lista_Norepetidos_menos_ventas=[]
for i in range(0,len(lista_productos_menos_ventas_Norepetidos),1):
    lista_Norepetidos_menos_ventas.append([lista_productos_menos_ventas_Norepetidos[i],lista_productos_menos_ventas_conteo[i],lifestore_products[lista_productos_menos_ventas_Norepetidos[i]]])

lista_menor_a_menos_ventas=[]
for i in range(0,len(lista_Norepetidos_menos_ventas),1):
    if lista_Norepetidos_menos_ventas[i][1]<=5:
        lista_menor_a_menos_ventas.append(lista_Norepetidos_menos_ventas[i])

lista_menor_a_menos_ventas.sort(key=lambda x:x[2],reverse=False)

Pcategoria_50productos_menos_ventas=[]
for i in range(0,len(lista_menor_a_menos_ventas),1):
    for j in range(0,len(lifestore_products),1):
        if lista_menor_a_menos_ventas[i][0] == lifestore_products[j][0]:
            Pcategoria_50productos_menos_ventas.append(["El producto es: "+str(lifestore_products[j][1])+" Y se vendio: "+str(lista_menor_a_menos_ventas[i][1])+" veces"])

```

Por categoría, generar un listado con los 100 productos con menos búsquedas.

En este punto me puse a prueba para reutilizar código, decidí como ya tenía los de mayor búsqueda en puntos anteriores, simplemente puse la condición en un ciclo for que la menor búsqueda fuera de 1 a 10 búsquedas del producto, después usé la función lambda para organizar de 1 a 10 siendo 1 el mayor en la lista y 10 el que se mostrara al final.

```

lista_mayor_a_searches=[]
for i in range(0,len(lista_Norepetidos_searches_conteo),1):
    if lista_Norepetidos_searches_conteo[i][1]<=10:
        lista_mayor_a_searches.append(lista_Norepetidos_searches_conteo[i])

lista_mayor_a_searches.sort(key=lambda x:x[1],reverse=False)

lista_top100_menos_buscados=[]
for i in range(0,len(lista_mayor_a_searches),1):
    for j in range(0,len(lifestore_products),1):
        if lista_mayor_a_searches[i][0] == lifestore_products[j][0]:
            lista_top100_menos_buscados.append(["El producto es: "+str(lifestore_products[j][1])+" Y se busco: "+str(lista_mayor_a_searches[i][1])+" veces"])

```

Punto 2°

2 Productos por reseña en el servicio

- ⌚ Mostrar dos listados de 20 productos cada uno, un listado para productos con las mejores reseñas y otro para las peores, considerando los productos con devolución.

Como en su enunciado menciona que son dos listas las cuales debemos de tener en cuenta los productos que se devolvieron y la reseña.

1. Lista de 20 productos con mejor reseña, considerando los productos con devolución.
2. Lista de 20 productos con peor reseña, considerando los productos con devolución.

Para ello haremos el uso de ciclos for y condiciones if para darle solución a este punto en sus dos incisos que a continuación explicare...

Como en los incisos pasados hacia varias listas para sacar los datos y meterlos a una nueva, haremos lo mismo en este punto ya que es una forma que me entiendo y siento que es la mejor forma de hacerlo para explicar y entender como es que se hace.

Lista de 20 productos con mejor reseña, considerando los productos con devolución.

Como las veces pasadas, haremos bastantes listas y meteremos una sobre otra para ordenar y filtrar la información solicitada de esta manera:

- Haremos una lista en la cual tendrá la condición de la reseña sea igual a 5 y que no sea un producto que se devolvió.
- Hacemos otra lista donde pongamos la lista anterior y tomemos los productos que no se repiten.
- Procedemos a hacer otra lista haciendo el conteo de los productos de la lista primera con la que no se repiten.
- Haciendo un plus, hacemos una lista para sacar las categorías de los productos con mejor reseña.
- Hacemos una lista para ordenar tanto el producto, las veces que se tuvo una reseña de 5 y la categoría la cual pertenece.
- Ordenamos la lista con la función lambda haciendo prioridad el conteo de las veces que el producto tuvo una reseña de 5.
- Ya para finalizar hacemos una lista para que se vea mejor la presentación de la información.

```
lista_mejor_reseña_sales=[]
for i in range(0,len(lifestore_sales),1):
    if lifestore_sales[i][2] >=5 and lifestore_sales[i][4] ==0:
        lista_mejor_reseña_sales.append(lifestore_sales[i][1])

lista_mejor_reseña_sales_Norepetidos=[]
for i in range(0,len(lista_mejor_reseña_sales),1):
    if lista_mejor_reseña_sales[i] != lista_mejor_reseña_sales[i-1]:
        lista_mejor_reseña_sales_Norepetidos.append(lista_mejor_reseña_sales[i])

lista_mejor_reseña_sales_conteo=[]
for i in range(0,len(lista_mejor_reseña_sales_Norepetidos),1):
    lista_mejor_reseña_sales_conteo.append(lista_mejor_reseña_sales.count(lista_mejor_reseña_sales_Norepetidos[i]))

lista_mejor_reseñas_Sales_categorias=[]
for i in range(0,len(lista_mejor_reseña_sales_Norepetidos),1):
    lista_mejor_reseñas_Sales_categorias.append(lifestore_products[lista_mejor_reseña_sales_Norepetidos[i]][3])

lista_mejor_reseña_Sales_orden=[]
for i in range(0,len(lista_mejor_reseña_sales_Norepetidos),1):
    lista_mejor_reseña_Sales_orden.append((lista_mejor_reseña_sales_Norepetidos[i],lista_mejor_reseña_sales_conteo[i],lista_mejor_reseñas_Sales_categorias[i]))

lista_mejor_reseña_Sales_orden.sort(key=lambda x:x[1],reverse=True)

lista_20_productos_mejor_reseñas=[]
for i in range(0,20,1):
    for j in range(0,len(lifestore_products),1):
        if lista_mejor_reseña_Sales_orden[i][0] == lifestore_products[j][0]:
            lista_20_productos_mejor_reseñas.append(["El producto: "+str(lifestore_products[j][1])+" tuvo "+str(lista_mejor_reseña_Sales_orden[i][1])+" reseñas de '5'"])
```

Lista de 20 productos con peor reseña, considerando los productos con devolución.

Como es el mismo caso, haremos un reciclaje de código:

- Hacemos una lista con un ciclo for que pondremos de condición if que la reseña sea menor o igual a 3 o que fuera un producto que se reembolso y también el conteo de las veces que tuvo una reseña mala.
- Hacemos otra lista sacando las categorías de la lista anterior.
- Hacemos una lista donde metamos el producto, las veces que tuvo una mala reseña y de que categoría es.
- Hacemos uso de la función lambda para ordenar las veces que se tuvo una mala reseña como prioridad en la lista pasada
- Y para finalizar hacemos una lista por estética de como se van a mostrar los productos con su nombre y las veces que se tuvo una mala reseña y su categoría.


```

lista_peor_reseña_sales=[]
lista_peor_reseña_sales_reseña=[]
for i in range(0,len(lifestore_sales),1):
    if lifestore_sales[i][2] <=3 or lifestore_sales[i][4] == 1:
        lista_peor_reseña_sales.append(lifestore_sales[i][1])
        lista_peor_reseña_sales_reseña.append(lifestore_sales[i][2])

lista_peor_reseña_sales_categoria=[]
for i in range(0,len(lista_peor_reseña_sales),1):
    lista_peor_reseña_sales_categoria.append(lifestore_products[lista_peor_reseña_sales[i]][3])

lista_peor_reseña_orden=[]
for i in range(0,len(lista_peor_reseña_sales),1):
    lista_peor_reseña_orden.append([lista_peor_reseña_sales[i],lista_peor_reseña_sales_reseña[i],lista_peor_reseña_sales_categoria[i]])

lista_peor_reseña_orden.sort(key=lambda x:x[1],reverse=True)

lista_top20_productos_peor_reseña=[]
for i in range(0,len(lista_peor_reseña_orden),1):
    for j in range(0,len(lifestore_products),1):
        if lista_peor_reseña_orden[i][0] == lifestore_products[j][0]:
            lista_top20_productos_peor_reseña.append("El producto es: "+str(lifestore_products[j][1])+" Y tuvo: "+str(lista_peor_reseña_orden[i][1])+" reseñas menores

```

Punto 3°



Total de ingresos y ventas promedio mensuales, total anual y meses con más ventas al año

En el siguiente link, puedes descargar el archivo para la ejecución del proyecto: <https://github.com/emtechinstitute/proyecto1>

En este punto también tiene un error, el cual los datos que nos dan son de casi todo un año, por tanto, el total de ingresos y el total anual es el mismo, también tiene otros dos puntos en los cuales tenemos que hacer un arreglo especial para sacar lo meses para hacer un promedio de ventas en los meses, así como decidir los meses con mas ventas, los puntos siguientes son:

- Total de ingresos.
- Ventas promedio mensuales.
- Total anual.
- Meses con más ventas al año.

Total de ingresos.

Ventas promedio mensuales.

Total anual.

Meses con más ventas al año.

Dentro de un ciclo for pondré una condición en la cual no debe de ser reembolso en el cual agregaremos dos valores en dos distintas listas y haremos la suma de del producto que se vendió ingresando en la lista de los productos, buscando el precio y así metiéndolo a la variable de total de ventas, en una lista sacamos el mes en el cual se vendió y lo guardamos, así como el nombre del producto y su precio para usarlo después. Si el producto es un reembolso se agrega el producto y su precio a una lista, y se hace el total de productos y sus precios con una suma.

Organizamos la lista con un sort(), después sacamos los meses que no se repiten para hacer el conteo de productos vendidos en cada mes, para volver a hacer otro ciclo for para destinar los productos con sus meses y precios en su mes que corresponde y meterlos en una lista describiendo, el mes y cuanto se vendió en ese mes y como un plus también hacemos el promedio de meses con reembolsos y sus precios de cada mes.

Con otra lista sacamos las ventas promedio de cada mes pero organizando las veces que ese mes se vendió algo para hacer el calculo de promedio de cada mes y para finalizar una última lista mostrando el cómo se vera la lista en la interfaz gráfica con unos arreglos con cadenas de caracteres.

```
total_ingresos_sales_ventas=[]
total_ingresos_sales_rembolsos=[]
total_rembolsos=0
total_ventas=0
meses_mas_ventas=[]
meses_mas_ventas_Norepetidos=[]
meses_mas_ventas_conteo=[]

for i in range(0,len(lifestore_sales),1):
    if lifestore_sales[i][4]==0:
        total_ingresos_sales_ventas.append([lifestore_sales[i][1],lifestore_products[lifestore_sales[i][1]][2]])
        total_ventas+=lifestore_products[lifestore_sales[i][1]][2]
        meses_mas_ventas.append(lifestore_sales[i][3][3:5])
    else:
        total_ingresos_sales_rembolsos.append([lifestore_sales[i][1],lifestore_products[lifestore_sales[i][1]][2]])
        total_rembolsos+=lifestore_products[lifestore_sales[i][1]][2]

meses_mas_ventas.sort()

for i in range(0,len(meses_mas_ventas),1):
    if meses_mas_ventas[i]!=meses_mas_ventas[i-1]:
        meses_mas_ventas_Norepetidos.append(meses_mas_ventas[i])

for i in range(0,len(meses_mas_ventas_Norepetidos),1):
    meses_mas_ventas_conteo.append(meses_mas_ventas.count(meses_mas_ventas_Norepetidos[i]))

for i in range(0,len(meses_mas_ventas_Norepetidos),1):
    if meses_mas_ventas_Norepetidos[i]=='01':
        meses_nombre.append('Enero')
    elif meses_mas_ventas_Norepetidos[i]=='02':
        meses_nombre.append('Febrero')
    elif meses_mas_ventas_Norepetidos[i]=='03':
        meses_nombre.append('Marzo')
    elif meses_mas_ventas_Norepetidos[i]=='04':
        meses_nombre.append('Abril')
    elif meses_mas_ventas_Norepetidos[i]=='05':
        meses_nombre.append('Mayo')
    elif meses_mas_ventas_Norepetidos[i]=='06':
        meses_nombre.append('Junio')
    elif meses_mas_ventas_Norepetidos[i]=='07':
        meses_nombre.append('Julio')
    elif meses_mas_ventas_Norepetidos[i]=='08':
        meses_nombre.append('Agosto')
    elif meses_mas_ventas_Norepetidos[i]=='09':
        meses_nombre.append('Septiembre')
    elif meses_mas_ventas_Norepetidos[i]=='10':
        meses_nombre.append('Octubre')
    elif meses_mas_ventas_Norepetidos[i]=='11':
        meses_nombre.append('Noviembre')
    elif meses_mas_ventas_Norepetidos[i]=='12':
        meses_nombre.append('Diciembre')

meses_mas_ventas_orden=[]
for i in range(0,len(meses_nombre),1):
    meses_mas_ventas_orden.append(['El mes de: '+meses_nombre[i],meses_mas_ventas_conteo[i], 'se vendio'])

promedio_rembolsos_total=total_rembolsos / len(total_ingresos_sales_rembolsos)

meses_ventas_promedio=[]
for i in range(0,len(meses_mas_ventas_Norepetidos),1):
    meses_ventas_promedio.append(str(round((total_ventas*meses_mas_ventas_conteo[i])/len(total_ingresos_sales_ventas),3)))
```

```

meses_nombres_ventas=[]
for i in range(0,len(meses_mas_ventas_Norepetidos),1):

    if meses_mas_ventas_Norepetidos[i]=='01':
        meses_nombres_ventas.append('Enero')
    elif meses_mas_ventas_Norepetidos[i]=='02':
        meses_nombres_ventas.append('Febrero')
    elif meses_mas_ventas_Norepetidos[i]=='03':
        meses_nombres_ventas.append('Marzo')
    elif meses_mas_ventas_Norepetidos[i]=='04':
        meses_nombres_ventas.append('Abril')
    elif meses_mas_ventas_Norepetidos[i]=='05':
        meses_nombres_ventas.append('Mayo')
    elif meses_mas_ventas_Norepetidos[i]=='06':
        meses_nombres_ventas.append('Junio')
    elif meses_mas_ventas_Norepetidos[i]=='07':
        meses_nombres_ventas.append('Julio')
    elif meses_mas_ventas_Norepetidos[i]=='08':
        meses_nombres_ventas.append('Agosto')
    elif meses_mas_ventas_Norepetidos[i]=='09':
        meses_nombres_ventas.append('Septiembre')
    elif meses_mas_ventas_Norepetidos[i]=='10':
        meses_nombres_ventas.append('Octubre')
    elif meses_mas_ventas_Norepetidos[i]=='11':
        meses_nombres_ventas.append('Noviembre')
    elif meses_mas_ventas_Norepetidos[i]=='12':
        meses_nombres_ventas.append('Diciembre')

meses_ventas_promedio_final=[]
for i in range(0,len(meses_nombres_ventas),1):
    meses_ventas_promedio_final.append(["El mes: "+str(meses_nombres_ventas[i])+" se vendio: "+str(meses_ventas_promedio[i])+" $"])

```

Solución al problema

1. Productos más vendidos y productos rezagados a partir del análisis de las categorías con menores ventas y categorías con menores búsquedas.
 - 1.1. Para este problema, hablando lo mas simplificado posible es visualizar el final de la lista que se mostrara en la interfaz gráfica, es decir, cada categoría en mi lista debe de contar con el producto, cuantas veces se vendió, y de que categoría es, así de sencillo es plantearse a lo que se quiere llegar, para ello, simplemente es hacer 3 listas con esos 3 requerimientos y después desglosarse..., es como hacer una pirámide, pero primero de la punta hacia abajo.
2. Productos por reseña en el servicio a partir del análisis de categorías con mayores ventas y categorías con mayores búsquedas.
 - 2.1. Hacemos lo mismo que punto pasado, comenzamos con el producto terminado, es decir la lista y el como se visualizara, y comenzamos a desglosar el como se ira haciendo todo, en este caso haremos uso de una lista la cual tendrá el producto cuantas veces tuvo una reseña de “x” y en que categoría se encuentra, pero para ello determinaremos todo eso haciendo filtros y condiciones en cada una de nuestras listas para que sea lo mas optima y que funcione a futuro.
3. Sugerir una estrategia de productos a retirar del mercado, así como sugerencia de cómo reducir la acumulación de inventario considerando los datos de ingresos y ventas mensuales.
 - 3.1. Para eso y con las conclusiones al final diré algo similar y es que para eso generalice, es decir, dependiendo las categorías y cuanto fue en sus ventas decidirá el dueño si es importante o no cierta categoría para que quite todos los productos de esa categoría y se centre en las categorías de mucha venta y de muchos ingresos, así como es bastante obvio, las categorías con menos ventas serán las de peores reseñas y eso lo note cuando hacia todo el programa que se repetía mucho cierta categoría.

Interfaz grafica

Para explicar la programación de la interfaz gráfica lo resumiré en 3 partes, las cuales son las ventanas de interfaz grafica y son las siguientes:

1. LifeStore – Login
2. LifeStore – Registro
3. LifeStore – Información

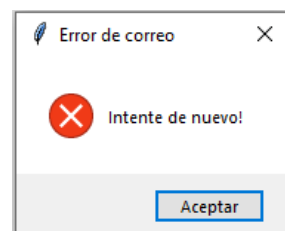
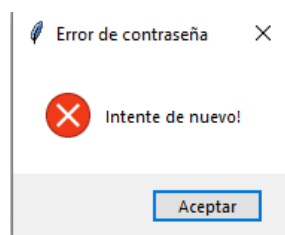
Arriba de cada una de las ventanas podrán observar los nombres ya mencionados, dentro de cada una de ellas podrán interactuar con las entradas y botones, así como si existe un problema con el correo o contraseña tanto en el Login como en el Registro le saltarán pequeñas ventanas que dirán cual es su error para poder corregirlos y continuar con el registro.

Administrador

Dentro de esta parte se encuentra la parte del administrador, la cual si teclean en el Login el correo: 123 y la contraseña: 123, los dejara entrar directamente a la ventada de Información, pero los invito a interactuar con el botón de Regístrate para que le salte la ventana de Registro y puedan ingresar un nuevo correo y contraseña y al darle al botón Ingresar los mande directamente al Login y tengan que ingresar los datos que pusieron anteriormente y vean que el código es 100% funcional sin Lachs o bucles infinitos...

Login

En la parte del Login podrán visualizar la siguiente imagen; la cual tendrá dos entradas y dos botones, las primeras dos entradas son para poner el correo y la contraseña, la cual tendrán dos opciones, crear una nueva cuenta y dándole click en el botón Regístrate que les abrirá otra ventana y registrarse o la opción de meterse como administrados la cual tendrá el valor en el correo de “123” y en la contraseña también



el valor de “123”, si por algo dentro de esta primera ventana tecleamos valores


incorrectos en el correo o en la contraseña les saldrán ventanas de información del error el cual tienen y como pueden resolverlo y son los que se muestran a continuación, cabe aclarar que estas mismas ventanas son las que se mostraran en la ventana de Registro solo con un mensaje diferente, con la finalidad de que sea más fácil de decir cual es el error al usuario sin que haya algún conflicto, ya que sin estas ventanas el usuario al dar varias veces click en ingresar y no poder, llegara a un punto en el cual dirá que el programa no funciona.



Registro

Si quisieran probar el boton de registrate les saltaria una ventana igual a esta, la cual deben de cumplir ciertos requisistos, son los siguientes:

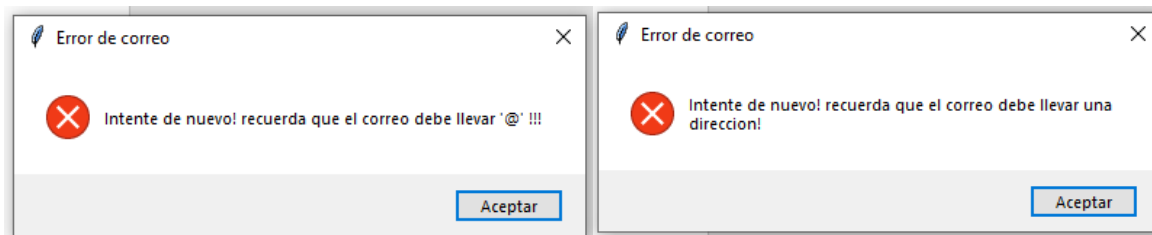
- El correo debe de llevar “@”.
- El correo debe de llevar una direccion:
 - Gmail.com
 - Hotmal.com
 - Outlook.com
- Deben de estar llenos los dos campos para que se pueda registrar los datos en una lista de datos.



The screenshot shows a window titled "LifeStore - Registro". Inside, the text "Life-Store" is displayed in a large, bold font, followed by "-Registro-" in a smaller font. Below this, there are two input fields: "Correo:" and "Contraseña:". To the right of the "Contraseña:" field is a button labeled "Ingresar".

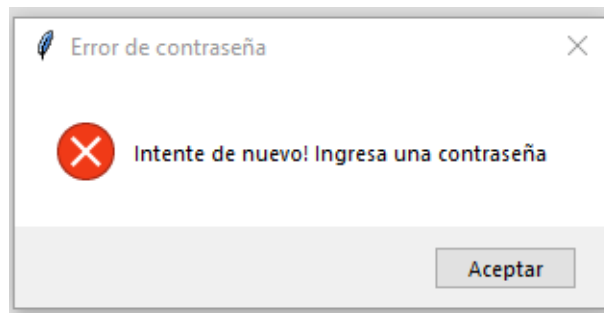
Ya que de lo contrario al presionar el boton Ingresar les arrojara unas pequeñas ventanas diciendo cual es el error y como pueden solucionarlo, son las siguientes:

Error de correo:



Two side-by-side screenshots of error messages. Both windows are titled "Error de correo". The left window shows a red circle with a white 'X' and the text "Intente de nuevo! recuerda que el correo debe llevar '@' !!!". The right window shows a red circle with a white 'X' and the text "Intente de nuevo! recuerda que el correo debe llevar una direccion!". Both windows have an "Aceptar" button at the bottom right.

Error de contraseña:



A screenshot of an error message window titled "Error de contraseña". It features a red circle with a white 'X' and the text "Intente de nuevo! Ingresa una contraseña". An "Aceptar" button is located at the bottom right.

Una ves que se haya hecho una nueva cuenta con esos datos y sin errores, al presionar el boton de Ingresar, automaticamente se cerrara la ventana de registro y deben de ingresar los nuevos datos de correo como de contraseña creados en la ventana de Login para iniciar sesion ya con un nuevo usuario con sus datos guardados en una lista flotante...

Cabe aclarar que ser correcto tambien los datos en el Login les abraira otra ventana de Informacion la cual podran salir y volver a ingresar ya sea con sus nuevos datos o con el correo y contraseña de administrados, que anteriormente se menciono.

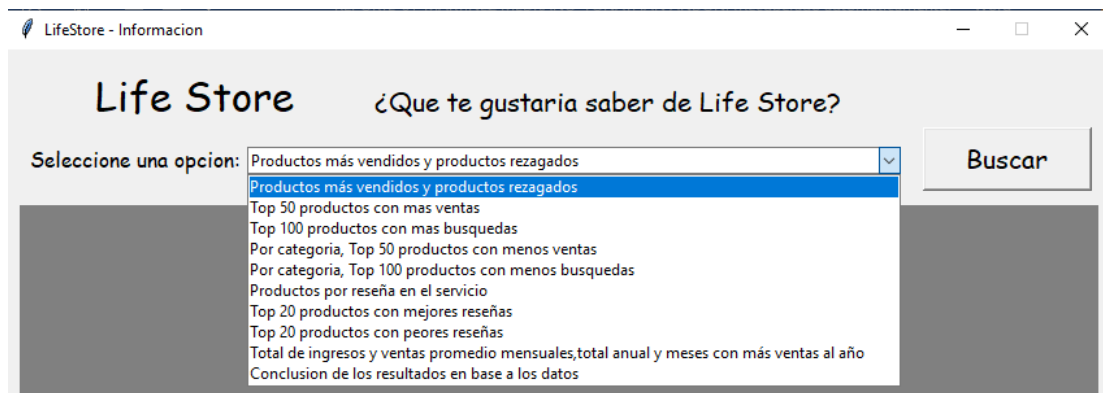
Informacion

Al ingresar correctamente los datos en la ventana principal de Login y darle click al boton Ingresar, se podra ver esta nueva ventana la cual podran seleccionar la opcion de datos que quieran visualizar con una lista desplegable las opciones la cual podran visualizar los puntos solicitados para esta empresa Life – Store con los datos que nos daban.

Pantalla de inicio :



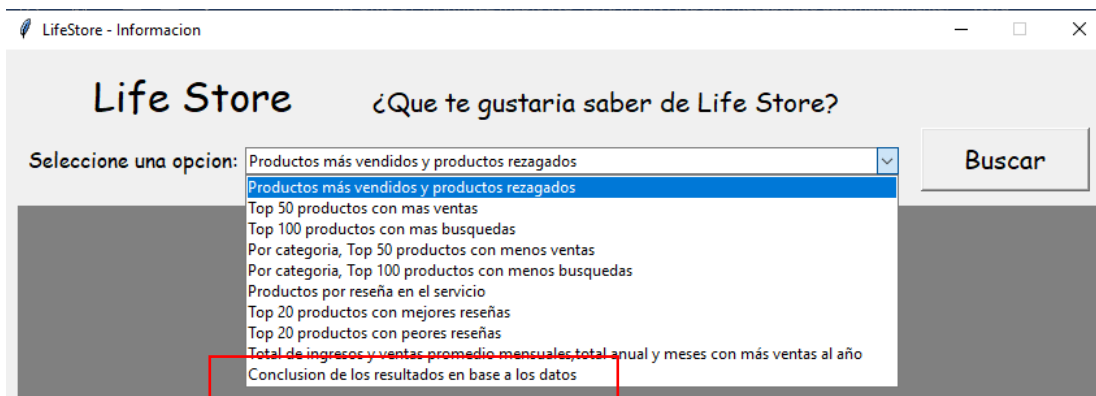
Lista desplegable en la pantalla inicio:

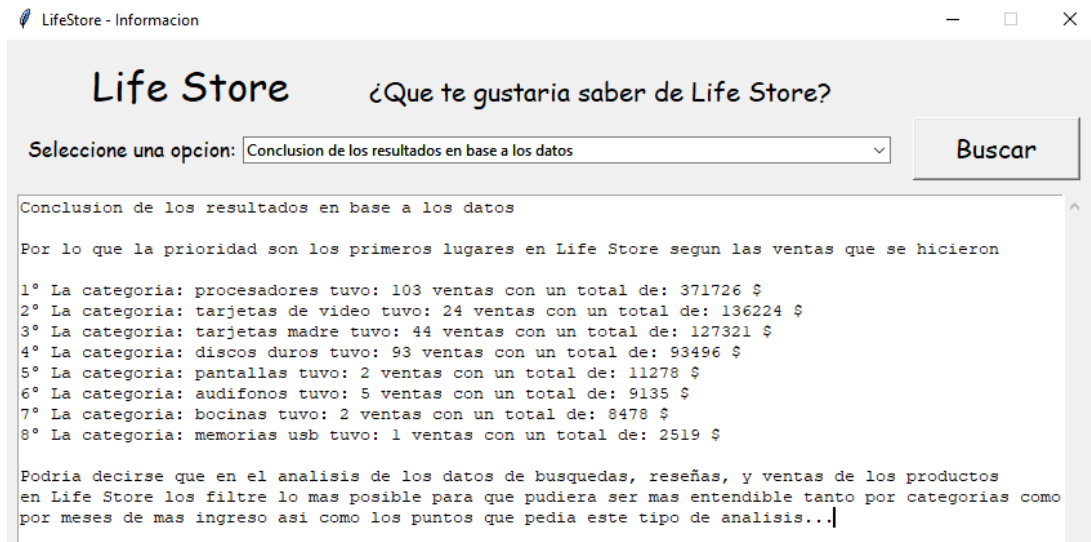


En cada una de las opciones que elija y al darle al botón buscar en la parte inferior se podrá visualizar algo similar a esto:



En la última opción fueron unas conclusiones generales que se podrán visualizar en el programa aunque se le agregan mas valores y productos nuevos, con la finalidad de que los valores de cada categoría cambien y se ordenen y filtren siempre y esto con el objetivo de que la persona, dueña pueda ver que categoría, cuantos productos se vendieron y cuanto genero durante todo ese tiempo y pueda ver donde se mueve más su mercancía y a que categorías centrarse y a cuales no.





Descripción del programa de la interfaz grafica

Al principio del programa como es costumbre, debemos de importar unas librerías, tanto de los datos que se mostraran dentro de la interfaz grafica como el de la interfaz gráfica.

```
from tkinter import *
from tkinter import ttk
import tkinter as tk
from tkinter import messagebox
from Codigo import *
```

```
ventana=tk.Tk()
ventana.geometry("350x300")
ventana.title("LifeStore - Login")
ventana.resizable(0,0)
```

Después tenemos que hacer la ventana principal, indicándole el tamaño y el título en la parte superior.

Creamos unas variables globales que se usaran dentro de todo el programa, así como unas listas donde se guardaran las listas de las contraseñas, correos y también una lista de opciones donde se visualizaran las opciones en la lista desplegable:

```
global correos
global contraseñas
global combo
global lista_desplegable

correos=['123']
contraseñas=['123']
opciones=['Productos más vendidos y productos rezagados',
          'Top 50 productos con mas ventas',
          'Top 100 productos con mas busquedas',
          'Por categoria, Top 50 productos con menos ventas',
          'Por categoria, Top 100 productos con menos busquedas',
          'Productos por reseña en el servicio',
          'Top 20 productos con mejores reseñas',
          'Top 20 productos con peores reseñas',
          'Total de ingresos y ventas promedio mensuales, total anual y meses con más ventas al año',
          'Conclusion de los resultados en base a los datos']
```

Nota: podemos observar que la lista de correos y contraseñas tenemos dos valores iguales que son los de administrador para que puedan visualizar la información sin necesidad de crear una cuenta.

Continuamos dentro de la ventana creando etiquetas visuales, ventanas para insertar datos y botones que corresponden al Login de usuario, con la finalidad de que al momento de ingresar los datos en las ventanas para insertar datos tengan un texto y al darle click al cualquier botón pueda llamar a una función y guardar los datos de las ventanas para insertar datos.

```
#Etiquetas-----
Label(text="Life-Store",font=("Comic Sans MS",23)).place(x=90,y=10)
Label(text="Correo: ",font=("Comic Sans MS",12)).place(x=48,y=80)
Label(text="Contraseña: ",font=("Comic Sans MS",12)).place(x=15,y=110)
Label(text="¿No tienes cuenta?",font=("Comic Sans MS",12)).place(x=100,y=200)

#Entradas-----
correo_entrada=StringVar()
Entry(ventana,textvariable=correo_entrada,width=35).place(x=115,y=87)
contraseña_entrada=StringVar()
Entry(ventana,textvariable=contraseña_entrada,width=35).place(x=115,y=117)

#Botones-----
boton_entrar=Button(ventana,width=17,text=("Ingresar"),font=("Comic Sans MS",12),command=Entrar).place(x=132,y=147)
boton_registrarse=Button(ventana,width=17,text=("Regístrate"),font=("Comic Sans MS",12),command=Registro).place(x=80,y=230)

ventana.mainloop()
```

Como dicho anteriormente, existen dos caminos los cuales el usuario puede tomar, el registrarse con una nueva cuenta o ingresar con la cuenta de administrador, si el usuario decidiera crear una cuenta dándole click al botón de Regístrate, les abrirá una ventana (llama a una función llamada: Registro) y la cual su programación consiste en hacer una nueva ventana con etiquetas, botones y ventanas para ingresar datos:

```
def Registro ():

    def guardar_datos ():
        for i in range(0,len(correo.get()),1):
            if correo.get()[i]=='@':
                if len(correo.get()[i:]) == '@gmail.com' or correo.get()[i:] == '@hotmail.com' or correo.get()[i:] == '@outlook.com':
                    if len(contraseña.get())!= 0:
                        correos.append(correo.get())
                        contraseñas.append(contraseña.get())
                        ventana_registro.destroy()
                        return None
                    messagebox.showerror("Error de contraseña","Intente de nuevo! Ingrese una contraseña")
                    return None
                messagebox.showerror("Error de correo","Intente de nuevo! recuerda que el correo debe llevar una direccion!")
                return None
            messagebox.showerror("Error de correo","Intente de nuevo! recuerda que el correo debe llevar '@' !!!")
            return None

    ventana_registro=Toplevel()
    ventana_registro.geometry("350x200")
    ventana_registro.title("LifeStore - Registro")
    ventana_registro.resizable(0,0)

    Label(ventana_registro,text="Life-Store",font=("Comic Sans MS",23)).place(x=90,y=10)
    Label(ventana_registro,text="-Registro-",font=("Comic Sans MS",15)).place(x=120,y=50)
    Label(ventana_registro,text="Correo: ",font=("Comic Sans MS",12)).place(x=48,y=80)
    Label(ventana_registro,text="Contraseña: ",font=("Comic Sans MS",12)).place(x=15,y=110)

    correo=StringVar()
    Entry(ventana_registro,width=35,textvariable=correo).place(x=115,y=87)

    contraseña=StringVar()
    Entry(ventana_registro,width=35,textvariable=contraseña).place(x=115,y=117)

    Button(ventana_registro,width=17,text="Ingresar",font=("Comic Sans MS",12),command=guardar_datos).place(x=130,y=147)
```

Podemos observar que es la llamada de una función dentro de otra llamada de otra función la cual tienen tareas distintas ya que cumpliendo la principal de crear lo visual y darle click al botón Ingresar esta primero llamada a la otra función de Guardardatos y al finalizar cerrara esa ventana para volver a la ventana principal de Login.

Al volver de nuevo a la ventana de Login podrán ingresar la cuenta creada o la de administrador, después al darle click al botón Ingresar, llamará la función Entrar y hará la siguiente programación, cabe destacar que es como el caso pasado, es una llamada en otra llamada de función, en este caso solamente mostrare lo que se ejecutara primero para que se pueda entender mejor, dentro de esta llamada de función se verificaran si los datos que tenemos en la lista donde se guardaron los datos creados o el de administrador son los mismos con un ciclo for y varias condiciones if, si esto no se cumple saldrán ventanas informativas que dirán el problema y como podrán solucionarlo, si todo esta correcto se creara otra ventana la cual es la de Información, con varias etiquetas, botones y unas lista desplegable la cual se visualiza en la ventana de Información y se mostrara los datos de la lista desplegable con la lista ya creada al principio de la explicación.

```
def Entrar ():
    for i in range(0,len(correos),1):
        if correo_entrada.get() == correos[i]:
            for j in range(0,len(contraseñas),1):
                if contraseña_entrada.get() == contraseñas[i]:
                    ventana_informacion=Toplevel()
                    ventana_informacion.geometry("850x700")
                    ventana_informacion.title("LifeStore - Informacion")
                    ventana_informacion.resizable(0,0)

                    frame=Frame(ventana_informacion,width=830,height=568,bg='gray')
                    frame.place(x=10,y=120)

                    lista_desplegable=StringVar()

                    Label(ventana_informacion,text="Life Store",font=("Comic Sans MS",23)).place(x=65,y=10)
                    Label(ventana_informacion,text="¿Que te gustaria saber de Life Store?",font=("Comic Sans MS",15)).place(x=280,y=23)
                    Label(ventana_informacion,text="Seleccione una opcion: ",font=("Comic Sans MS",12)).place(x=15,y=70)

                    lista_desplegable = ttk.Combobox(ventana_informacion,width=80,value=opciones,state='readonly')
                    lista_desplegable.place(x=185,y=75)
                    lista_desplegable.current(0)

                    scrollbar=tk.Scrollbar(frame)
                    texto_solicitado=tk.Text(frame,width=101,height=37)

                    Button(ventana_informacion,text="Buscar",width=10,font=("Comic Sans MS",15),command=main).place(x=705,y=60)

                    return None
    messagebox.showerror("Error de contraseña","Intente de nuevo!")
    return None
messagebox.showerror("Error de correo","Intente de nuevo!")
```

Continuando a mostrar los datos, al darle click al botón Buscar, este llamara a la función main, la cual primero borrara todo, después con condiciones if – elif y con la lista desplegable el valor que en formato cadena que tiene preguntara en cada uno de las condiciones para ver cual se cumple y dentro de alguna de estas condiciones se mostrara la lista de la opción que se desea buscar, si se vuelve a cambiar el valor en formato cadena de la lista desplegable y al darle click de nuevo al botón Buscar, volverá a llamar a la función main para entrar en el ciclo y primero borrar lo que tenia pasado, y preguntar en cada condición cual se eligió para imprimir todo y poder visualizarlo en la interfaz gráfica utilizando el scroll.

Código para mostrar los datos en la interfaz grafica haciendo uso de la lista desplegable y del botón Buscar.

```
def main():

    texto_solicitado.delete("1.0","end")

    if lista_desplegable.get() == 'Productos más vendidos y productos rezagados':
        scroll.pack(side=tk.RIGHT,fill=tk.Y)
        texto_solicitado.pack(side=tk.LEFT,fill=tk.Y)

        scroll.config(command=texto_solicitado.yview)
        texto_solicitado.config(yscrollcommand=scroll.set)

        texto_solicitado.insert(tk.INSERT,"Top 50 productos con mas ventas\n\n")
        for i in range(0,len(lista_top50_mayores_ventas),1):
            texto_solicitado.insert(tk.INSERT,chr(i+1)+"° "+str(lista_top50_mayores_ventas[i])+"\n")

        texto_solicitado.insert(tk.INSERT,"\nTop 100 productos con mas busquedas\n\n")
        for i in range(0,len(lista_top100_mayores_buscados),1):
            texto_solicitado.insert(tk.INSERT,chr(i+1)+"° "+str(lista_top100_mayores_buscados[i])+"\n")

        texto_solicitado.insert(tk.INSERT,"\nPor categoria, Top 50 productos con menos ventas\n\n")
        for i in range(0,len(Pcategoria_50productos_menos_ventas),1):
            texto_solicitado.insert(tk.INSERT,chr(i+1)+"° "+str(Pcategoria_50productos_menos_ventas[i])+"\n")

        texto_solicitado.insert(tk.INSERT,"\nPor categoria, Top 100 productos con menos busquedas\n\n")
        for i in range(0,len(lista_top100_menos_buscados),1):
            texto_solicitado.insert(tk.INSERT,chr(i+1)+"° "+str(lista_top100_menos_buscados[i])+"\n")

    elif lista_desplegable.get() == 'Top 50 productos con mas ventas':
        scroll.pack(side=tk.RIGHT,fill=tk.Y)
        texto_solicitado.pack(side=tk.LEFT,fill=tk.Y)

        scroll.config(command=texto_solicitado.yview)
        texto_solicitado.config(yscrollcommand=scroll.set)

        texto_solicitado.insert(tk.INSERT,"Top 50 productos con mas ventas\n\n")
        for i in range(0,len(lista_top50_mayores_ventas),1):
            texto_solicitado.insert(tk.INSERT,chr(i+1)+"° "+str(lista_top50_mayores_ventas[i])+"\n")

    elif lista_desplegable.get() == 'Top 100 productos con mas busquedas':
        scroll.pack(side=tk.RIGHT,fill=tk.Y)
        texto_solicitado.pack(side=tk.LEFT,fill=tk.Y)

        scroll.config(command=texto_solicitado.yview)
        texto_solicitado.config(yscrollcommand=scroll.set)

        texto_solicitado.insert(tk.INSERT,"Top 100 productos con mas busquedas\n\n")
        for i in range(0,len(lista_top100_mayores_buscados),1):
            texto_solicitado.insert(tk.INSERT,chr(i+1)+"° "+str(lista_top100_mayores_buscados[i])+"\n")

    elif lista_desplegable.get() == 'Por categoria, Top 50 productos con menos ventas':
        scroll.pack(side=tk.RIGHT,fill=tk.Y)
        texto_solicitado.pack(side=tk.LEFT,fill=tk.Y)

        scroll.config(command=texto_solicitado.yview)
        texto_solicitado.config(yscrollcommand=scroll.set)

        texto_solicitado.insert(tk.INSERT,"Por categoria, Top 50 productos con menos ventas\n\n")
        for i in range(0,len(Pcategoria_50productos_menos_ventas),1):
            texto_solicitado.insert(tk.INSERT,chr(i+1)+"° "+str(Pcategoria_50productos_menos_ventas[i])+"\n")
```

```

elif lista_desplegable.get() == 'Por categoria, Top 100 productos con menos búsquedas':
    scroll.pack(side=tk.RIGHT, fill=tk.Y)
    texto_solicitado.pack(side=tk.LEFT, fill=tk.Y)

    scroll.config(command=texto_solicitado.yview)
    texto_solicitado.config(yscrollcommand=scroll.set)

    texto_solicitado.insert(tk.INSERT, "Por categoria, Top 100 productos con menos búsquedas\n\n")
    for i in range(0, len(lista_top100_menos_buscados), 1):
        texto_solicitado.insert(tk.INSERT, str(i+1) + "° " + str(lista_top100_menos_buscados[i]) + "\n")

elif lista_desplegable.get() == 'Productos por reseña en el servicio':
    scroll.pack(side=tk.RIGHT, fill=tk.Y)
    texto_solicitado.pack(side=tk.LEFT, fill=tk.Y)

    scroll.config(command=texto_solicitado.yview)
    texto_solicitado.config(yscrollcommand=scroll.set)

    texto_solicitado.insert(tk.INSERT, "Top 20 productos con mejores reseñas\n\n")
    for i in range(0, len(lista_20_productos_mejor_reseñas), 1):
        texto_solicitado.insert(tk.INSERT, str(i+1) + "° " + str(lista_20_productos_mejor_reseñas[i]) + "\n")

    texto_solicitado.insert(tk.INSERT, "\nTop 20 productos con peores reseñas\n\n")
    for i in range(0, len(lista_top20_productos_peor_reseña), 1):
        texto_solicitado.insert(tk.INSERT, str(i+1) + "° " + str(lista_top20_productos_peor_reseña[i]) + "\n")

elif lista_desplegable.get() == 'Top 20 productos con mejores reseñas':
    scroll.pack(side=tk.RIGHT, fill=tk.Y)
    texto_solicitado.pack(side=tk.LEFT, fill=tk.Y)

    scroll.config(command=texto_solicitado.yview)
    texto_solicitado.config(yscrollcommand=scroll.set)

    texto_solicitado.insert(tk.INSERT, "Top 20 productos con mejores reseñas\n\n")
    for i in range(0, len(lista_20_productos_mejor_reseñas), 1):
        texto_solicitado.insert(tk.INSERT, str(i+1) + "° " + str(lista_20_productos_mejor_reseñas[i]) + "\n")

elif lista_desplegable.get() == 'Top 20 productos con peores reseñas':
    scroll.pack(side=tk.RIGHT, fill=tk.Y)
    texto_solicitado.pack(side=tk.LEFT, fill=tk.Y)

    scroll.config(command=texto_solicitado.yview)
    texto_solicitado.config(yscrollcommand=scroll.set)

    texto_solicitado.insert(tk.INSERT, "Top 20 productos con peores reseñas\n\n")
    for i in range(0, len(lista_top20_productos_peor_reseña), 1):
        texto_solicitado.insert(tk.INSERT, str(i+1) + "° " + str(lista_top20_productos_peor_reseña[i]) + "\n")

elif lista_desplegable.get() == 'Total de ingresos y ventas promedio mensuales, total anual y meses con más ventas al año':
    scroll.pack(side=tk.RIGHT, fill=tk.Y)
    texto_solicitado.pack(side=tk.LEFT, fill=tk.Y)

    scroll.config(command=texto_solicitado.yview)
    texto_solicitado.config(yscrollcommand=scroll.set)

    texto_solicitado.insert(tk.INSERT, "Total de ingresos y ventas promedio mensuales, total anual y meses con más ventas al año\n\n")
    texto_solicitado.insert(tk.INSERT, "El total de ventas es: " + str(total_ventas) + " $ \n\n")
    texto_solicitado.insert(tk.INSERT, "El promedio mensual de reembolsos es: " + str(round(promedio_rembolsos_total, 3)) + " $ \n\n")
    texto_solicitado.insert(tk.INSERT, "El promedio de ventas cada mes se muestra en la siguiente lista: \n\n")

    for i in range(0, len(meses_ventas_promedio_final), 1):
        texto_solicitado.insert(tk.INSERT, str(i+1) + "° " + str(meses_ventas_promedio_final[i]) + "\n")

    texto_solicitado.insert(tk.INSERT, "\nLos meses con mas ventas y cuantas ventas tuvieron se muestran en la siguiente lista: \n\n")

    for i in range(0, len(meses_mas_ventas_orden), 1):
        texto_solicitado.insert(tk.INSERT, str(i+1) + "° " + str(meses_mas_ventas_orden[i]) + "\n")

elif lista_desplegable.get() == 'Conclusion de los resultados en base a los datos':
    scroll.pack(side=tk.RIGHT, fill=tk.Y)
    texto_solicitado.pack(side=tk.LEFT, fill=tk.Y)

    scroll.config(command=texto_solicitado.yview)
    texto_solicitado.config(yscrollcommand=scroll.set)

    texto_solicitado.insert(tk.INSERT, "Conclusion de los resultados en base a los datos\n\n")
    texto_solicitado.insert(tk.INSERT, "\nPor lo que la prioridad son los primeros lugares en Life Store segun las ventas que se hicieron\n\n")
    for i in range(0, len(lista_conclusion_completa), 1):
        texto_solicitado.insert(tk.INSERT, str(i+1) + "° " + str(lista_conclusion_completa[i]) + "\n")

    texto_solicitado.insert(tk.INSERT, "\nPodría decirse que en el analisis de los datos de búsquedas, reseñas, y ventas de los productos")
    texto_solicitado.insert(tk.INSERT, "\nen Life Store los filtre lo mas posible para que pudiera ser mas entendible tanto por categorias como")
    texto_solicitado.insert(tk.INSERT, "\npor meses de mas ingreso así como los puntos que pedia este tipo de analisis...")

```

Conclusión

Conclusión general

- Al realizar este tipo de ejercicios, fue nuevo para mí en ciertos aspectos, el trabajar con un nuevo lenguaje, creando una interfaz gráfica con este lenguaje y usando este mismo orientada a la ciencia de datos y no a sistemas periféricos o programas físicos orientados a mi carrera, fue un reto para mí pero que siempre tuvo una solución adecuada para cada uno de los puntos que se solicitaban, dicho esto una vez más le agradezco a Santander por darme esta oportunidad y experiencia de lograr lo que me propongo y los retos nuevos que son estos para mí, en resumen, puedo decir que la complejidad de este tipo de ejercicio ya una vez terminándolo me abrió la cabeza para pensar la funcionalidad que tiene la programación en la ciencia de datos.

Conclusiones individuales

- Los puntos que se solicitaron fueron previamente analizados y filtrados para mostrarlos en la interfaz gráfica.
- Aunque no fue un programa muy óptimo, use las herramientas necesarias para entender lo que el programa EmTech me brindó y lo puse a prueba.
- Las conclusiones de dicho análisis fueron bastante resumidas y muy fáciles de entender (se pueden ver en la ventana de Información, abriendo la lista desplegable y seleccionar la última parte de conclusiones y dándole click al botón buscar) ya que el encargado o dueño de dicha empresa verá la categoría, cuántas veces se vendió y cuánto fue su monto total de ingresos, ordenando el monto total de ingresos de mayor a menor, siendo la prioridad el cuándo deja cierta categoría que el número de ventas de esa categoría.