

A dark blue vertical bar is positioned on the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the date. Below the banner, several thin, curved lines in dark blue and light grey sweep upwards from the bottom left corner.

15 de Septiembre de 2021

Reporte LifeStore

GitHub:

<https://github.com/NerfeP/REPORTE-01-PAZZI-EFREN>

EFREN PAZZI MANZANO
EMTECH INSTITUTE

INDICE

Introducción	2
Definición del código	2
Solución al problema	7
Conclusión	7

Introducción

Python es un lenguaje de programación muy popular ya que llega a ser muy sencillo utilizar y tienen una comunidad tan amplia que ha evolucionado a ser uno de los lenguajes mas poderosos.

Para el análisis de datos es una herramienta increíble de nuevo por su comunidad se crean bibliotecas desde el manejo de gráficos, hasta el codificar para inteligencia artificial

Si se quiere ser un científico de datos es indispensable saber manejar Python, por ello en el siguiente reporte es para desarrollar las habilidades necesarias en el manejo de bases de datos.

Definición del código

Para una mejor visualización se recomienda descargar el código del GitHub (<https://github.com/NerfeP/REPORTE-01-PAZZI-EFREN>)

Al iniciar el programa nos muestra el log in (Ilustración 1), la contraseña para Jimmy es

- Usuario: Jimmy
- Contraseña : EmtechRules

```
C:\Users\pazzi\Documents\ITSX\Emtech\Proyecto_final_primer_curso>py PROYECTO-01-PAZZI-EFREN.PY

Por favor Ingrese
Usuario: Jimmy
Contraseña: EmtechRules
```

Ilustración 1 Log in

Al tener las credenciales correctas se ingresa al menú del reporte (Ilustración 2)

```

Bienvenido Jimmy

El siguiente reporte es para la empresa Life Store
1.-Productos ventas y busquedas
2.-Reseñas
3.-Ventas
4.-Estrategia sugerida
5.-Salir del reporte

Ingresa tu eleccion:
```

Ilustración 2 Menú del reporte

Al ingresar 1 y después a ingresara a los productos más vendidos (ilustración 3)

```
Bienvenido Jimmy

El siguiente reporte es para la empresa Life Store
1.-Productos ventas y búsquedas
  a.-Ocultar 15 productos mas vendidos
-----
15 Productos con mas ventas
-----
id      Nombre del producto      No. Ventas
-----
54      SSD Kingston A400, 120GB  50
3       Procesador AMD Ryzen 5 2   42
5       Procesador Intel Core i3 20
42      Tarjeta Madre ASRock Mic 18
57      SSD Adata Ultimate SU800 15
29      Tarjeta Madre ASUS micro 14
4       Procesador AMD Ryzen 3 3 13
2       Procesador AMD Ryzen 5 3 13
47      SSD XPG SX8200 Pro, 256G 11
48      SSD Kingston A2000 NVMe, 9
12      Tarjeta de Video ASUS NV 9
7       Procesador Intel Core i7 7
44      Tarjeta Madre MSI ATX B4 6
31      Tarjeta Madre AORUS micr 6
18      Tarjeta de Video Gigabyt 5
-----
-----

      b.-Mostrar 20 productos mas buscados
      c.-Mostrar 5 productos menos vendidos por categoria
      d.-Mostrar 5 productos menos buscados por categoria
2.-Reseñas
3.-Ventas
4.-Estrategia sugerida
5.-Salir del reporte

Ingresa tu eleccion:
```

Ilustración 3 Productos más vendidos

Al volver a ingresar a se cerrar la tabla (ilustración 4)

```
Bienvenido Jimmy

El siguiente reporte es para la empresa Life Store
1.-Productos ventas y búsquedas
  a.-Mostrar 15 productos mas vendidos
  b.-Mostrar 20 productos mas buscados
  c.-Mostrar 5 productos menos vendidos por categoria
  d.-Mostrar 5 productos menos buscados por categoria
2.-Reseñas
3.-Ventas
4.-Estrategia sugerida
5.-Salir del reporte

Ingresa tu eleccion:
```

Ilustración 4 Productos más vendidos 2

Para esta tabla se hizo el siguiente código (ilustración 5)

```
lista_mas_vendidos = f_lista_iguales(lifestore_sales, 1)#obtener una lista de productos vendidos
nombres = lista_nombres(lifestore_products, 0, 1, lista_mas_vendidos, 0)#Obteber los nombres de los productos
for i in range(len(lista_mas_vendidos)):#Meter los nombres para imprimirlos
    pivote = lista_mas_vendidos[i][1]
    lista_mas_vendidos[i][1] = nombres[i]
    lista_mas_vendidos[i].append(pivote)
lista_mas_vendidos = orden(lista_mas_vendidos, 2)#Ordenar las ventas y cuales fueron los que mas se vendieron
mini_lista_1 = []
for i in range(1,16):
    mini_lista_1.append(lista_mas_vendidos[-i])#El ordenamiento es desde el mas pequeño
```

Ilustración 5 Código para la primera tabla

Es muy sencillo, se creo una función para agrupar todos los elementos iguales en una lista, así se cuentan todas las ventas. (Ilustración 6)

```
def f_lista_iguales(lista, indice_comparar):#Funcion para contar los indices iguales, recuerda que a
    lista_iguales_1 = []
    contador = 0
    for i in range(len(lista)):
        contador += 1
        if i < len(lista)-1:
            if lista[i][indice_comparar] != lista[i+1][indice_comparar]:
                lista_iguales_1.append([lista[i][indice_comparar], contador])
                contador = 0
            else:
                continue
        else:
            lista_iguales_1.append([lista[i][indice_comparar], contador])
    return lista_iguales_1
```

Ilustración 6 funcion Lista de iguales

Después con otra función sacamos los nombres de los productos y la metemos en la lista de las ventas contadas hecha previamente

```
def lista_nombres(lista_de_nombres, indice_comparar_a, indice_nombre, lista_sin_nombres, indice_comparar_b):
    contador = 0 #Contador necesario para encontrar los nombres de cada producto vendido
    nombres = []
    for i in lista_de_nombres:#Buscar los nombres de los productos vendidos, para visualizarlos mejor
        if i[indice_comparar_a] == lista_sin_nombres[contador][indice_comparar_b]:
            nombres.append(i[indice_nombre])
            contador += 1
        if len(lista_sin_nombres) == len(nombres):
            break
    return nombres
```

Ilustración 7 Lista de nombres

Ordenamos porque queremos de mayor a menores ventas

```
def orden(lista, indice): #Inicia una funcion para Quicksort. Los parametros son la lista a ordenar
    izquierda = []
    centro = []
    derecha = []
    if len(lista) > 1:
        pivote = lista[0][indice]
        for i in lista:
            if i[indice] < pivote:
                izquierda.append(i)
            elif i[indice] == pivote:
                centro.append(i)
            elif i[indice] > pivote:
                derecha.append(i)
        return orden(izquierda, indice) + centro + orden(derecha, indice)
    else:
        return lista
```

Ilustración 8 Función para ordenar, Quicksort

Se acomodan los índices como queremos mostrarlos, y lo guardamos en una lista para poder imprimirla después (Ir al final de la explicación del código).

Todas las listas fueron realizadas de una manera parecida, pero con ligeras diferencias por los índices que pedían. (Ver el código en GitHub para mejor análisis)

Para el menú se hizo un conjunto de decisiones para mostrar correctamente los datos y en la secuencia que se pedía (Ilustración 9 y 8)

```

import_borrar()
print("{>25} {}".format("Bienvenido", usuario))
print("\n\tEl siguiente reporte es para la empresa Life Store")
print("\t\t1.-Productos ventas y busquedas")
if opciones[0]:
    if opciones[1]:
        print("\t\t\t1.-Ocultar 15 productos mas vendidos")
        titulo = '15 Productos con mas ventas'
        cabeza = ['id', 'Nombre del producto', 'No. Ventas']
        imp_tab(titulo, cabeza, mini_lista_1)
        imp_linea_vacia(1)
    else:
        print("\t\t\t2.-Mostrar 15 productos mas vendidos")
    if opciones[2]:
        print("\t\t\t3.-Ocultar 20 productos mas buscados")
        titulo = '20 Productos con mas busquedas'
        cabeza = ['id', 'Nombre del producto', 'No. Busquedas']
        imp_tab(titulo, cabeza, mini_lista_2)
        imp_linea_vacia(1)
    else:
        print("\t\t\t4.-Mostrar 20 productos mas buscados")
    if opciones[3]:
        print("\t\t\t5.-Ocultar 5 productos menos vendidos por categoria")
        for i in range(len(categorias)): #Imprimir todas las tablas de los productos menos vendidos
            mini_lista_3 = []
            for j in range(len(lista_ventas_categorias[i])):
                mini_lista_3.append(lista_ventas_categorias[i][j])
            if j >= 5:
                break
            titulo = '5 Productos menos vendidos de ' + categorias[i][0]
            cabeza = ['id', 'Nombre del producto', 'No. Ventas']
            imp_tab(titulo, cabeza, mini_lista_3)
            imp_linea_vacia(1)
    else:
        print("\t\t\t6.-Mostrar 5 productos menos vendidos por categoria")
    if opciones[4]:
        print("\t\t\t7.-Ocultar 5 productos menos buscados por categoria")
        for i in range(len(categorias)): #Imprimir todas las tablas de los productos menos vendidos
            mini_lista_4 = []
            for j in range(len(lista_cat_menos_buscados[i])):
                mini_lista_4.append(lista_cat_menos_buscados[i][j])
            if j >= 5:

```

Ilustración 9 Código de los textos en el menú

```

opcion = input('\n\t\tIngresa tu eleccion: ')
if opcion == '1':
    for i in range(len(opciones)):
        opciones[i] = False
    opciones[0] = True
elif opcion == '2':
    for i in range(len(opciones)):
        opciones[i] = False
    opciones[5] = True
elif opcion == '3':
    for i in range(len(opciones)):
        opciones[i] = False
    opciones[8] = True
elif opcion == '4':
    for i in range(len(opciones)):
        opciones[i] = False
    opciones[11] = True
elif opcion == '5':
    break

elif opciones[0] and opcion == 'a' and not opciones[1]:
    opciones[1] = True
elif opciones[0] and opcion == 'a' and opciones[1]:
    opciones[1] = False
elif opciones[0] and opcion == 'b' and not opciones[2]:
    opciones[2] = True
elif opciones[0] and opcion == 'b' and opciones[2]:
    opciones[2] = False
elif opciones[0] and opcion == 'c' and not opciones[3]:
    opciones[3] = True
elif opciones[0] and opcion == 'c' and opciones[3]:
    opciones[3] = False
elif opciones[0] and opcion == 'd' and not opciones[4]:
    opciones[4] = True
elif opciones[0] and opcion == 'd' and opciones[4]:
    opciones[4] = False

elif opciones[5] and opcion == 'a' and not opciones[6]:
    opciones[6] = True
elif opciones[5] and opcion == 'a' and opciones[6]:
    opciones[6] = False
elif opciones[5] and opcion == 'b' and not opciones[7]:
    opciones[7] = True
elif opciones[5] and opcion == 'b' and opciones[7]:
    opciones[7] = False

elif opciones[8] and opcion == 'a' and not opciones[9]:
    opciones[9] = True
elif opciones[8] and opcion == 'a' and opciones[9]:
    opciones[9] = False
elif opciones[8] and opcion == 'b' and not opciones[10]:
    opciones[10] = True
elif opciones[8] and opcion == 'b' and opciones[10]:
    opciones[10] = False

else:
    imp_borrar()

```

Ilustración 10 Decisiones en las opciones del menú

Hay que puntualizar que para mostrar las tablas se creó una función llamada `imp_tab()`

```
def imp_tab(titulo, cabeza, lista):#funcion para imprimir listas
    ancho_total = 0
    for i in cabeza:
        ancho_total += len(i) + 2
    imp_linea(2,ancho_total)
    formato = '{:^'+ str(ancho_total) + '}'
    print(formato.format(titulo))
    imp_linea(1,ancho_total)
    formato = ''
    for i in range(len(cabeza)):
        formato += '[a[' + str(i) + ']:<' + str(len(cabeza[i]) + 2) + ']' + str(len(cabeza[i])) + ']'
    print(formato.format(a = cabeza))
    imp_linea(1,ancho_total)
    lista_a_str = []
    for i in lista:#Se creara un lista que todos sus elementos sean strings para que se pueda imprimir con el formato
        mini_lista = []
        for j in i:
            mini_lista.append(str(j))
        lista_a_str.append(mini_lista)
    for i in lista_a_str:
        print(formato.format(a = i))
    imp_linea(2,ancho_total)
```

Ilustración 11 Función para imprimir las tablas

Solución del problema

Dentro del código se llega la siguiente solución

4.-Estrategia sugerida

Se recomienda;

- 1.-Los 4 primeros productos de la lista 2.b sean retirados del inventario
- 2.-Retirar los productos con 0 ventas en las listas 1.c
- 3.-Retirar todos productos con 0 búsquedas en las listas 1.d

1. Los 4 primeros productos de la lista 2.b sean retirados del inventario
2. Retirar los productos con 0 ventas en las listas 1.c
3. Retirar todos productos con 0 búsquedas en las listas 1.d

Conclusión

El reporte se realizó con las todas las consignas correctas

La estrategia, aunque cortas, son concisas y en relación a los datos expuestos a lo largo de reporte

El tiempo para una persona sin experiencia en el manejo de datos y programación es muy justo, pero suficiente considerando un código sencillo y sin el uso de herramientas mas complejas para el análisis.