

Ejercicio_3_Listas

"""

Escribir un programa modular que haciendo uso de listas, permita:

El ingreso de una secuencia de valores, que termina con el valor 0.

1. Muestre los valores ingresados.
2. Muestre los valores hasta encontrar el 3er. valor impar ingresado inclusive.
3. Muestre los elementos que se encuentren en posiciones pares.
4. Muestre los elementos ordenados de menor a mayor, sin repetirlos.

En todos los casos, las salidas deben contener un título que indique lo que se esta mostrando y mostrar un valor por linea.

"""

```
def generar_lista():
    print("Ingreso de Valores - Finalice ingresando 0", end = "\n\n")
    lista = []
    valor = int(input("Valor: "))
    while (valor != 0):
        lista.append(valor)
        valor = int(input("Valor: "))
    return lista

def mostrar(titulo, lista):
    print("\n", titulo)
    for elemento in lista:
        print("\t", elemento)

def generar_3er_impar(lista):
    lista_3er_impar = []
    cant_imp = 0
    posicion = 0
    while ((cant_imp < 3) and (posicion < len(lista))):
        lista_3er_impar.append(lista[posicion])
        if (lista[posicion]%2 == 1):
            cant_imp += 1
        posicion += 1
    return lista_3er_impar

def generar_pos_pares(lista):
    lista_pos_pares = []
    for posicion in range(0, len(lista), 2):
        lista_pos_pares.append(lista[posicion])
    return lista_pos_pares

def generar_ord_sin_dup(lista):
    lista_ord = sorted(lista)
    lista_ord_sin_dup = []
    anterior = 0
```

Ejercicio_3_Listas

```
for elemento in lista_ord:
    if elemento != anterior:
        lista_ord_sin_dup.append(elemento)
        anterior = elemento
return lista_ord_sin_dup
```

#----- Bloque Principal -----#

```
l_original = generar_lista() #Punto 1
```

```
mostrar("Datos Ingresados", l_original) #Punto 2
```

```
l_3er_impar = generar_3er_impar(l_original) #Punto 3
mostrar("Datos hasta 3er. Valor Impar", l_3er_impar)
```

```
l_pos_pares = generar_pos_pares(l_original) #Punto 4
mostrar("Datos en posiciones pares", l_pos_pares)
```

```
l_ord_sin_dup = generar_ord_sin_dup(l_original) #Punto 5
mostrar("Datos ordenados sin duplicados", l_ord_sin_dup)
```