

# Proyecto 1: Ordenador de Arreglos (Método Burbuja)

Curso: Python de 0 a Interfaces Gráficas

Proyecto de los Temas 1 al 5

## Índice

<b>1. Objetivo del Proyecto</b>	<b>2</b>
1.1. Temas Aplicados . . . . .	2
<b>2. Descripción del Programa</b>	<b>2</b>
<b>3. Guía de Desarrollo Paso a Paso</b>	<b>3</b>
3.1. Paso 1: El Algoritmo (La Herramienta) . . . . .	3
3.2. Paso 2: El Menú Principal (El Cerebro) . . . . .	3
3.3. Paso 3: Las Funciones de las Opciones . . . . .	4
3.3.1. Función: modo_manual() . . . . .	4
3.3.2. Función: modo_aleatorio() . . . . .	4
<b>4. Código Completo del Proyecto</b>	<b>6</b>

# 1. Objetivo del Proyecto

El objetivo de este primer proyecto es construir una aplicación de consola completa que combine todos los conceptos aprendidos desde el Tema 1 hasta el Tema 5.

Crearás una herramienta que le permite al usuario ordenar una lista de números usando el algoritmo clásico "Método Burbuja".

## 1.1. Temas Aplicados

- **Tema 1 (Fundamentos):** Uso de variables, `print()` para mostrar texto, `input()` para recibir datos del usuario, y `int()` para convertir el texto a números.
- **Tema 2 (Lógica):** Uso de Listas (arreglos) para guardar los números, `if/elif/else` para la lógica del menú, y bucles `for` (¡anidados!) para implementar el algoritmo de ordenamiento.
- **Tema 3 (Bucles while):** Uso de un bucle `while True` para crear un menú principal que se repita constantemente hasta que el usuario decida salir con `break`.
- **Tema 4 (Estructuras):** Aunque no usamos diccionarios aquí, reforzamos el uso avanzado de Listas.
- **Tema 5 (Funciones):** La parte más importante. Vamos a modularizar (separar) nuestro código en funciones. Tendremos una función para el menú, una para el algoritmo burbuja, y una para cada opción.
- **Librerías Externas:** Usaremos la librería `random` para generar los números aleatorios de la Opción 2.

# 2. Descripción del Programa

El programa debe ser un menú interactivo que le presente al usuario 3 opciones:

**Opción 1: Ingresar Manualmente.** El programa le preguntará al usuario cuántos números desea ingresar. Luego, le pedirá cada número uno por uno. Finalmente, le mostrará la lista desordenada y la lista ordenada.

**Opción 2: Generar Aleatoriamente.** El programa le preguntará al usuario cuántos números desea generar. También le pedirá un rango (un valor mínimo y un máximo). El programa generará la lista aleatoria, la mostrará desordenada y luego la mostrará ordenada.

**Opción 3: Salir.** El programa mostrará un mensaje de despedida y terminará.

### 3. Guía de Desarrollo Paso a Paso

Para construir esto, no escribimos todo de golpe. Lo hacemos por partes.

#### 3.1. Paso 1: El Algoritmo (La Herramienta)

Primero, creamos nuestra "herramienta" principal: la función que ordena.

- **¿Cómo funciona la Burbuja?** Es simple: recorre la lista comparando cada número con el siguiente (en pares).
- Si el número actual es **mayor** que el siguiente, los **intercambia**.
- Repite este proceso una y otra vez hasta que no haya más intercambios.
- Para lograr esto, necesitamos **dos bucles for anidados**:
  - El **bucle externo** (con *i*) solo se asegura de que el proceso se repita las veces necesarias.
  - El **bucle interno** (con *j*) es el que hace las comparaciones y los intercambios.

```
def bubble_sort(lista_desordenada):
    # Hacemos una copia para no destruir la original
    lista = lista_desordenada.copy()
    n = len(lista) # Total de elementos

    # Bucle externo (i)
    for i in range(n):
        # Bucle interno (j)
        # (n - i - 1) es una optimizacion
        for j in range(0, n - i - 1):

            # La comparacion clave:
            if lista[j] > lista[j + 1]:
                # El intercambio
                lista[j], lista[j + 1] = lista[j + 1], lista[j]

    return lista # Devolvemos la lista ordenada
```

#### 3.2. Paso 2: El Menú Principal (El Cerebro)

Ahora creamos la función `main()` que será el cerebroz contendrá el menú (Tema 3). Por ahora, solo pondremos mensajes.

```
def main():
    # El bucle 'while True' se repite para siempre
    while True:
        print("\n--- ORDENADOR DE ARREGLOS (BURBUJA) ---")
        print("1. Ingresar arreglo manualmente")
        print("2. Generar arreglo aleatorio")
```

```

print("3. Salir")

opcion = input("Elige una opcion: ")

if opcion == "1":
    print("...Aqui ira la funcion manual...")

elif opcion == "2":
    print("...Aqui ira la funcion aleatoria...")

elif opcion == "3":
    print("¡Adios!")
    break # 'break' rompe el bucle

else:
    print("Opcion no valida. Intenta de nuevo.")

```

### 3.3. Paso 3: Las Funciones de las Opciones

Ahora creamos las funciones para las opciones 1 y 2, y las conectamos al `main()`.

#### 3.3.1. Función: `modo_manual()`

```

def modo_manual():
    print("\n--- Modo Manual ---")
    mi_lista = [] # Lista vacia
    try:
        total_numeros = int(input("Cuantos numeros vas a ingresar?: "))

        for i in range(total_numeros):
            num = int(input(f"Ingresa el numero {i + 1}: "))
            mi_lista.append(num)

        print(f"\nTu lista original: {mi_lista}")

        # --- AQUÍ USAMOS NUESTRA HERRAMIENTA ---
        lista_ordenada = bubble_sort(mi_lista)

        print(f"Tu lista ordenada: {lista_ordenada}")

    except ValueError:
        print("Error: Debes ingresar solo numeros.")

```

#### 3.3.2. Función: `modo_aleatorio()`

¡No olvides poner `import random` al inicio de tu archivo!

```
import random # Poner al inicio del archivo
```

```

def modo_aleatorio():
    print("\n--- Modo Aleatorio ---")
    try:
        total_numeros = int(input("Cuantos numeros quieres generar?: "))
        minimo = int(input("Valor minimo del rango: "))
        maximo = int(input("Valor maximo del rango: "))

        mi_lista = []
        for _ in range(total_numeros):
            num_aleatorio = random.randint(minimo, maximo)
            mi_lista.append(num_aleatorio)

        print(f"\nLista generada (desordenada): {mi_lista}")

        # --- AQUI USAMOS NUESTRA HERRAMIENTA ---
        lista_ordenada = bubble_sort(mi_lista)

        print(f"Lista ordenada: {lista_ordenada}")

    except ValueError:
        print("Error: Debes ingresar solo numeros.")

```

## 4. Código Completo del Proyecto

Este es el archivo final, juntando todas las piezas.

```
import random # Para la opcion 2

# --- PASO 1: EL ALGORITMO (HERRAMIENTA) ---
def bubble_sort(lista_desordenada):
    """
    Recibe una lista y la ordena usando
    el metodo burbuja (de menor a mayor).
    """
    lista = lista_desordenada.copy()
    n = len(lista) # Numero de elementos

    # Bucle 'for' anidado
    for i in range(n):
        # Optimizacion: si no hay cambios, paramos
        swapped = False

        for j in range(0, n - i - 1):
            if lista[j] > lista[j + 1]:
                # Intercambio
                lista[j], lista[j + 1] = lista[j + 1], lista[j]
                swapped = True

        if not swapped:
            break # Si ya esta ordenado, salimos

    return lista # Devolvemos la lista ordenada

# --- PASO 3: FUNCIONES DE LAS OPCIONES ---
def modo_manual():
    """
    Pide al usuario que ingrese los datos uno por uno.
    """
    print("\n--- Modo Manual ---")
    mi_lista = [] # Lista vacia
    try:
        total_numeros = int(input("Cuantos numeros vas a ingresar?: "))

        for i in range(total_numeros):
            num = int(input(f"Ingresar el numero {i + 1}: "))
            mi_lista.append(num)

        print(f"\nTu lista original: {mi_lista}")

        # Llamamos a nuestra funcion de ordenar
        lista_ordenada = bubble_sort(mi_lista)
```