

## Informe Metodo de Ordenamiento BubbleSort

```
def bubblesort(vectorbs):
    """Esta función ordenara el vector que le pases como argumento con el Método de Bubble Sort"""

    # Imprimimos la lista obtenida al principio (Desordenada)
    print("El vector a ordenar es:",vectorbs)
    n = 0 # Establecemos un contador del largo del vector

    for _ in vectorbs:
        n += 1 #Contamos la cantidad de caracteres dentro del vector

    for i in range(n-1):
        # Le damos un rango n para que complete el proceso.
        for j in range(0, n-i-1):
            # Revisa la matriz de 0 hasta n-i-1
            if vectorbs[j] > vectorbs[j+1] :
                vectorbs[j], vectorbs[j+1] = vectorbs[j+1], vectorbs[j]
            # Se intercambian si el elemento encontrado es mayor
            # Luego pasa al siguiente
        print ("El vector ordenado es: ",vectorbs)
```

Primero mostramos en pantalla el vector a ordenar con todos sus elementos usando **print**, luego obtenemos el tamaño del vector y lo guardamos en la variable **n** para luego poder recorrer dicho vector con un **for** que se mueve con el iterable **i**, dentro de un rango **n-1** que representa los índices de los elementos dentro del vector.

Dentro de ese primer **for** creamos otro que se moverá con el iterable **j**, esta vez el rango va hasta **n-i-1** con **i** representando la cantidad de veces que ya se ha recorrido el vector ordenando sus elementos, para no volver a revisar partes del vector que ya deberían estar ordenadas.

El segundo **for** es para recorrer los elementos de la lista comparando el elemento actual con el que le sigue (**if vector[j] > vector[j+1]**) y si resulta ser mayor el elemento actual se cambian los valores dentro de la lista para ir ordenando.

Al final del recorrido deberíamos obtener una lista de elementos completamente ordenada sin ningún problema y la mostramos en pantalla con **print** nuevamente.