

Universidad central de Venezuela

Alumno: Rubén Aparicio

Asignatura: Programación (0790)

Tarea N°5

Concepto a investigar:

Método de ordenamiento burbuja o bubblesort.

```
from random import sample
# Importamos un Método de la biblioteca random para generar listas aleatorias

lista = list(range(100)) # Creamos la lista base con números del 1 al 100

# Creamos una lista aleatoria con sample
#(8 elementos aleatorios de la lista base)
vectorbs = sample(lista,8)

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)

    bubblesort(vectorbs)
```

En este algoritmo se nos plantea el **método de burbuja o bubblesort** para el ordenamiento de elemento de una variable.

Es importante notar como se comienza importando el método `sample` para lograr definir posteriormente la cantidad de elementos de la lista que vamos a ordenar.

- Se procede a crear una lista base de números que van del 0 al 100. Esto nos da un amplio rango de valores a escoger para nuestro arreglo.

```
lista = list(range(100)) # Creamos la lista base con números del 1 al 100
```

- Con el método `sample` nos servirá para tomar una cierta cantidad de valores al azar de nuestra lista base, para crear una nueva lista. Con la línea "`vectorbs = sample(lista,8)`" se indica que la nueva lista, va a tomar una nuestra al azar de la lista base, en este caso 8 elementos

```
vectorbs = sample(lista,8)
```

- Luego pasamos a definir una función que se encargara de ordenar de mayor a menor los elementos de esa lista aleatoria.
- Usamos el ciclo "`for _ in vectorbs`" para recorrer cada elemento dentro de la lista y establecemos un contador para saber la cantidad de caracteres del vector.

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

- Luego recorreremos el vector desde su primera posición hasta la penúltima. Esto se debe a que por la estructura de este

algoritmos, se compara un valor con el de la posición siguiente; por lo que no podemos llegar hasta la última posición, ya que no habrían más valores con los cuales comparar, ocasionando que el programa nos arrojara un error.

```
for i in range(n-1):  
    # Le damos un rango n para que complete el proceso.  
    for j in range(0, n-i-1):
```

- Luego establecemos otro ciclo para comparar una posición del vector con la que le sigue. Y en caso de que el valor anterior sea de un mayor, reasignamos los valores de las posiciones dentro del vector. En este caso se hace uso de una asignación múltiple, una ventaja bastante útil de python.

```
if vectorbs[j] > vectorbs[j+1] :  
    vectorbs[j], vectorbs[j+1] = vectorbs[j+1], vectorbs[j]
```

Nota: esta doble reasignación se puede hacer de otra manera, en caso de que se busque una forma más general, para poder aplicarla en otros lenguajes donde no exista la posibilidad de realizar una asignación múltiple, como se muestra a continuación.

Definimos la variable respaldo para no perder el valor de la posición que estamos comparando, luego se procede a hacer la reasignación de cada posición en líneas de código distintas.

```
if vectorbs[j] > vectorbs[j+1] :  
    respaldo = vectorbs[j]  
    vectorbs[j] = vectorbs[j+1]  
    vectorbs[j+1] = respaldo
```

- Finalmente se presenta el vector ordenado por pantalla.

```
print ("El vector ordenado es: ",vectorbs)
```