



# FUNDAMENTOS DE INFORMATICA

INGENIERA SILVIA PATRICIA BARDELLI

PROFESORA ING. SILVIA PATRICIA BARDELLI

# CLASE NRO 8

Temas:

§ Ordenamiento de listas:

Método de Intercambio o Burbujeo

Método de Inserción

§ Aplicaciones de las listas

# MÉTODO DE INTERCAMBIO O BURBUJEO

- § Se basa en comparar cada elemento con el que tiene a su derecha.
- § Si es necesario, se los intercambia.
- § Luego se avanza a la siguiente pareja y se repite el proceso.

# MÉTODO DE INTERCAMBIO O BURBUJEO

1

2	7	5	1	6
---	---	---	---	---



# MÉTODO DE INTERCAMBIO O BURBUJEO

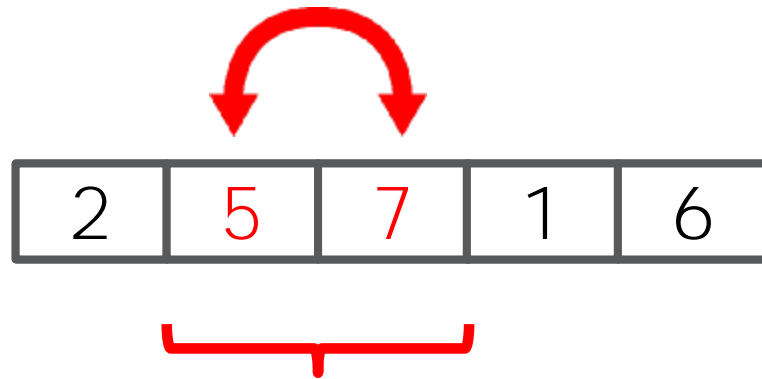
2

2	7	5	1	6
---	---	---	---	---



# MÉTODO DE INTERCAMBIO O BURBUJEO

3



# MÉTODO DE INTERCAMBIO O BURBUJEO

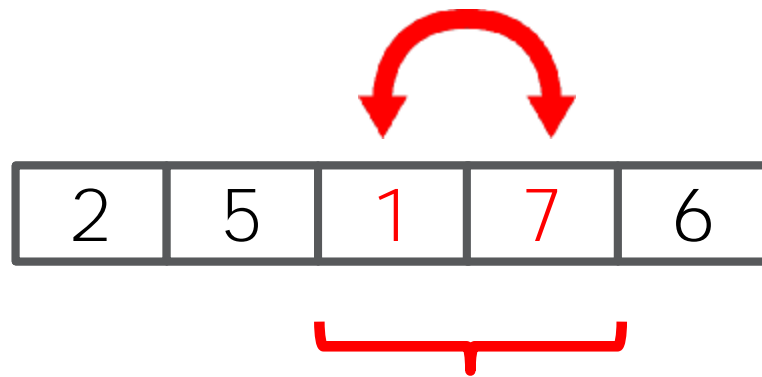
4

2	5	7	1	6
---	---	---	---	---



# MÉTODO DE INTERCAMBIO O BURBUJEO

5





# MÉTODO DE INTERCAMBIO O BURBUJEO

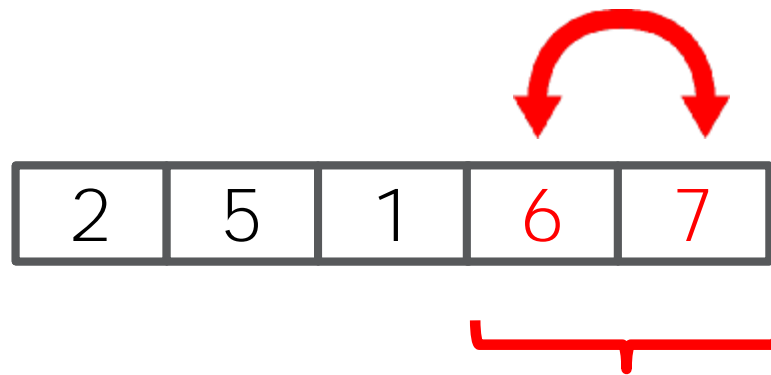
6

2	5	1	7	6
---	---	---	---	---



# MÉTODO DE INTERCAMBIO O BURBUJEO

7



# MÉTODO DE INTERCAMBIO O BURBUJEO

8

2	5	1	6	7
---	---	---	---	---

Llegamos al final.  
*¿Quedó ordenada la lista?*

# MÉTODO DE INTERCAMBIO O BURBUJEO

9

2	5	1	6	7
---	---	---	---	---



No. Volvemos a empezar.

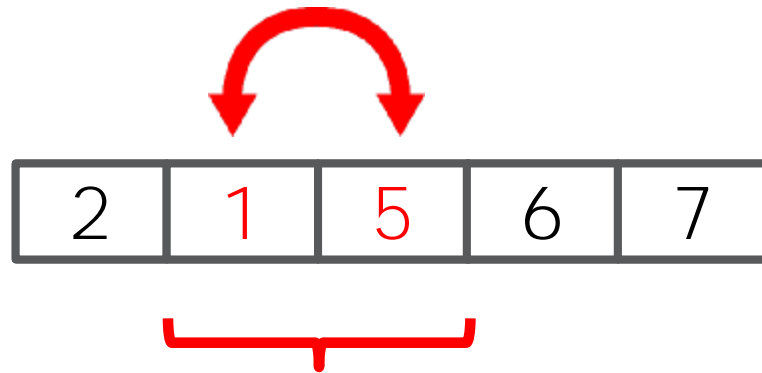
# MÉTODO DE INTERCAMBIO O BURBUJEO

10



# MÉTODO DE INTERCAMBIO O BURBUJEO

11



# MÉTODO DE INTERCAMBIO O BURBUJEO

12



# MÉTODO DE INTERCAMBIO O BURBUJEO

13





# MÉTODO DE INTERCAMBIO O BURBUJEO

14

2	1	5	6	7
---	---	---	---	---

Llegamos al final.  
*¿Quedó ordenada la lista?*

# MÉTODO DE INTERCAMBIO O BURBUJEO

15

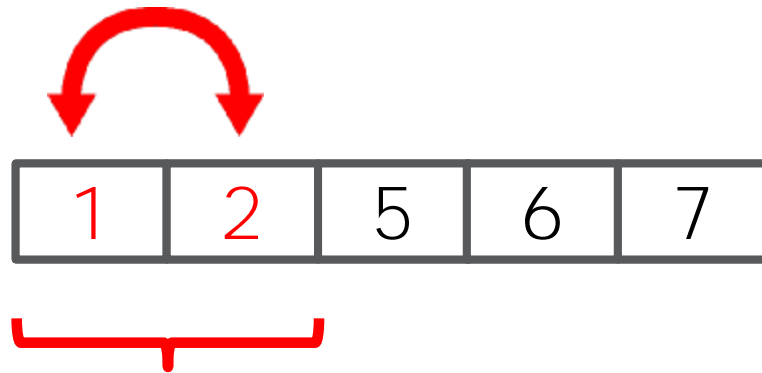
2	1	5	6	7
---	---	---	---	---



No. Volvemos a empezar.

# MÉTODO DE INTERCAMBIO O BURBUJEO

16



# MÉTODO DE INTERCAMBIO O BURBUJEO

17



# MÉTODO DE INTERCAMBIO O BURBUJEO

18



# MÉTODO DE INTERCAMBIO O BURBUJEO

19



# MÉTODO DE INTERCAMBIO O BURBUJEO

20

1	2	5	6	7
---	---	---	---	---

Llegamos al final.  
*¿Quedó ordenada la lista?*

# MÉTODO DE INTERCAMBIO O BURBUJEO

21

1	2	5	6	7
---	---	---	---	---





# MÉTODO DE INTERCAMBIO O BURBUJEO

22



# MÉTODO DE INTERCAMBIO O BURBUJEO

23



# MÉTODO DE INTERCAMBIO O BURBUJEO

24



# MÉTODO DE INTERCAMBIO O BURBUJEO

25

1	2	5	6	7
---	---	---	---	---

Se completó una pasada *sin realizar ningún intercambio.* **è** Está ordenada

# MÉTODO DE INTERCAMBIO O BURBUJEO

```
def metododeintercambio(v):  
    desordenado = True  
    while desordenado:  
        desordenado = False  
        for i in range(len(v)-1):  
            if v[i]>v[i+1]:  
                aux = v[i]  
                v[i] = v[i+1]  
                v[i+1] = aux  
                desordenado = True
```

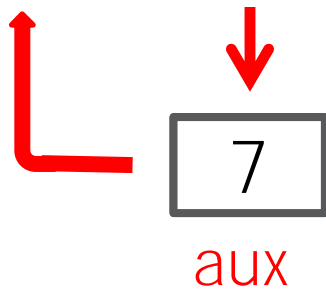
# MÉTODO DE INSERCIÓN

§Comienza a ordenar a partir del **segundo** elemento de la lista.

§Consiste en mover cada elemento del arreglo hacia la izquierda, haciéndolo retroceder hasta encontrar su ubicación definitiva.

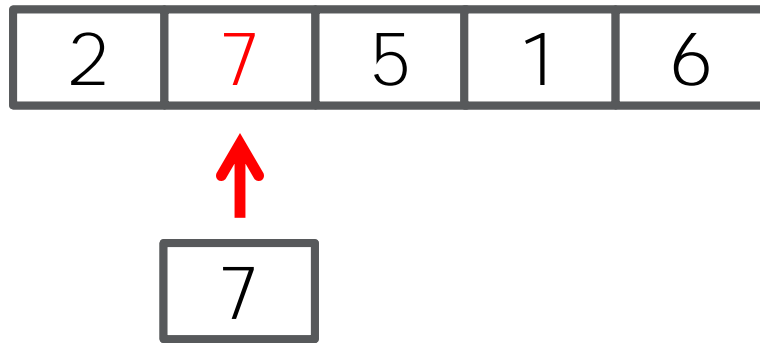
# MÉTODO DE INSERCIÓN

1



# MÉTODO DE INSERCIÓN

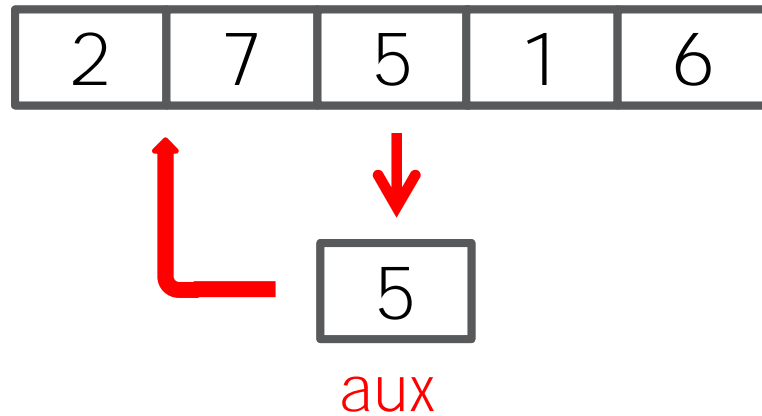
2





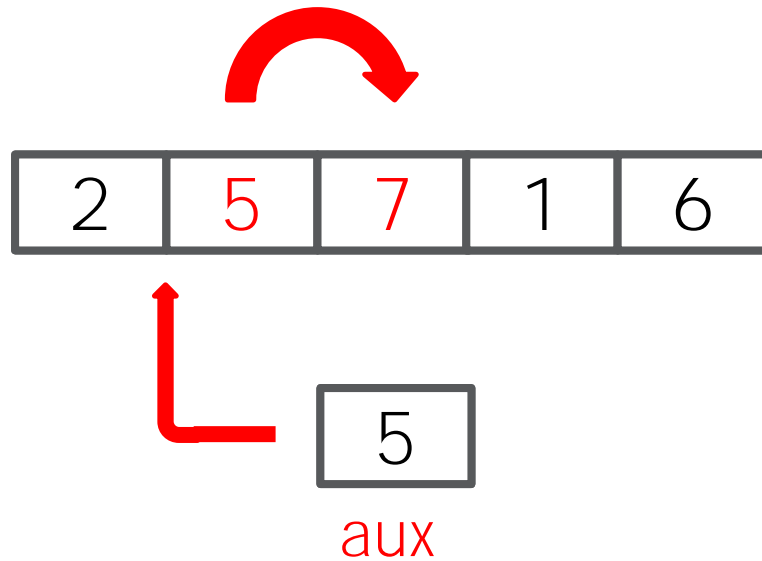
# MÉTODO DE INSERCIÓN

3



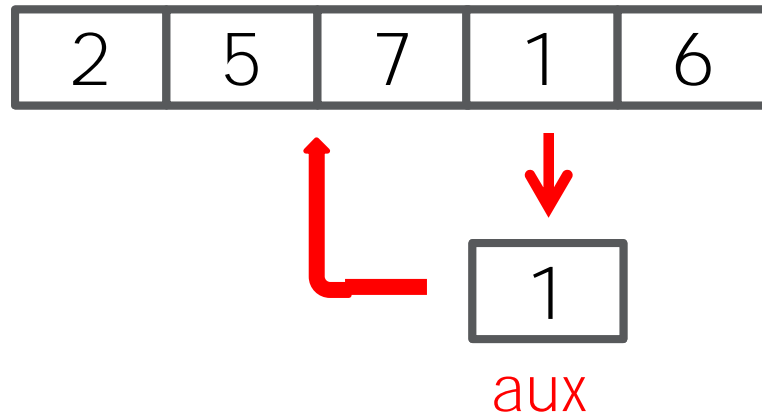
# MÉTODO DE INSERCIÓN

4



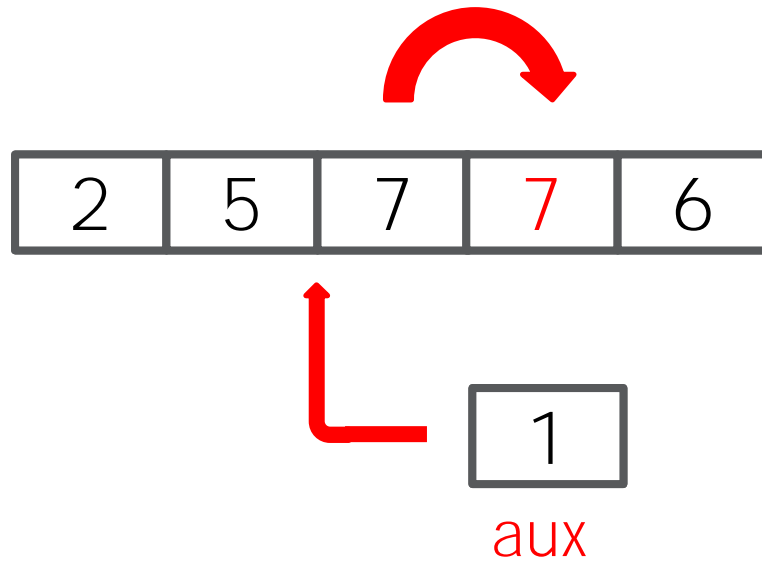
# MÉTODO DE INSERCIÓN

5



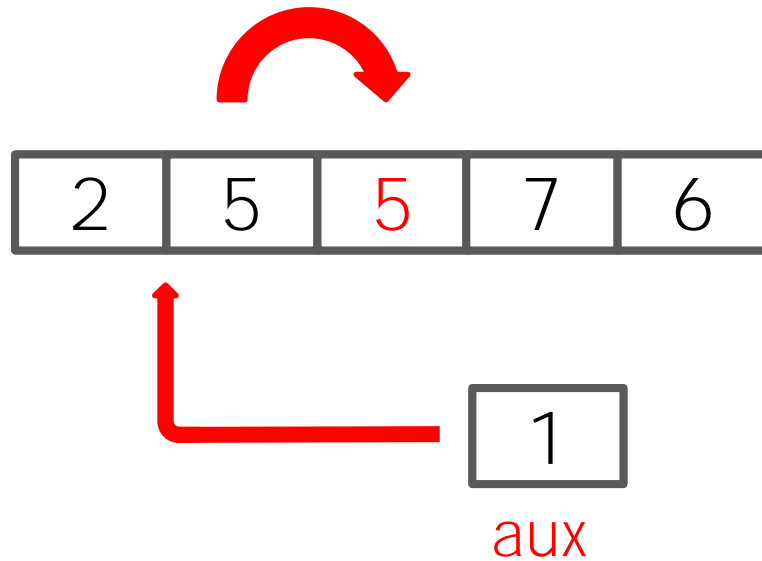
# MÉTODO DE INSERCIÓN

6



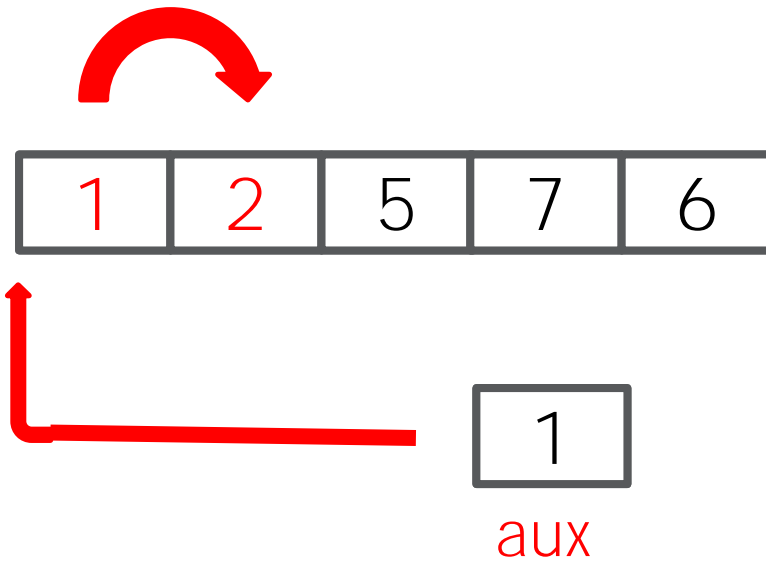
# MÉTODO DE INSERCIÓN

7



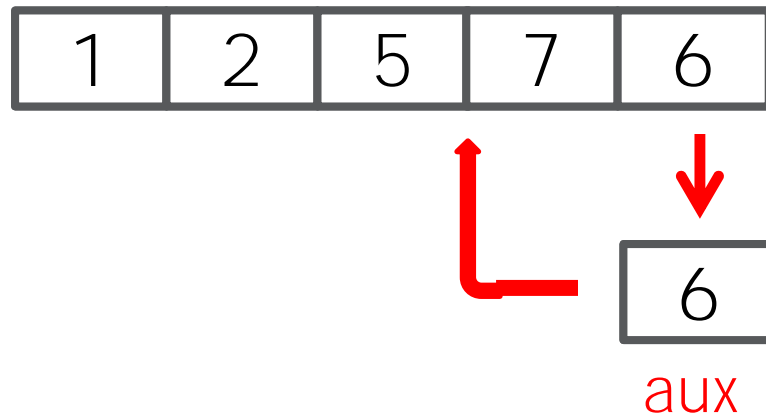
# MÉTODO DE INSERCIÓN

8



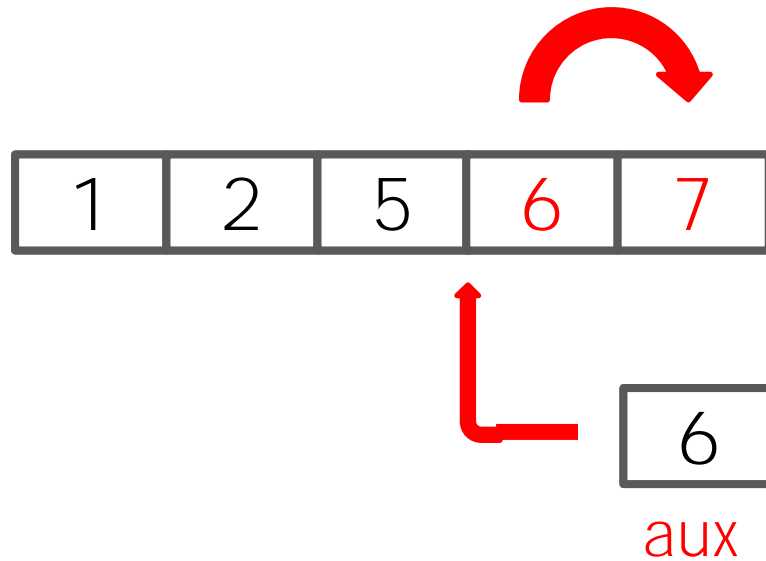
# MÉTODO DE INSERCIÓN

9



# MÉTODO DE INSERCIÓN

10





# MÉTODO DE INSERCIÓN

11

1	2	5	6	7
---	---	---	---	---

La lista quedó ordenada.

# MÉTODO DE INSERCIÓN

```
def metododeinsercion(v):  
    for i in range(1, len(v)): # empieza del 2º elemento  
        aux = v[i]  
        j = i  
        while j>0 and v[j-1]>aux:  
            v[j] = v[j-1]  
            j = j-1  
        v[j] = aux
```

# APLICACIONES DE LAS LISTAS

## Uso de listas como acumulador

# EJEMPLO

Una empresa cuenta con 50 vendedores, numerados del 1 al 50. Por cada venta realizada se ingresa el número de vendedor y el importe de la misma, donde el número de vendedor -1 indica el final de los datos.

Estos datos no están ordenados.  
Realizar un programa para imprimir el total de ventas por vendedor.

## EJEMPLO

¿Cuántos acumuladores se necesitan?

## EJEMPLO – PRIMERA PARTE

```
def leervendedor(maximo):  
    n = int(input("Número de vendedor? (-1 para  
terminar) "))  
    while (n!=-1 and (n<1 or n>maximo)):  
        print("*** Vendedor inválido ***")  
        n = int(input("Número de vendedor? (-1 para  
terminar) "))  
    return n
```

## EJEMPLO – SEGUNDA PARTE

*# Programa principal*

VENDEDORES = 50

# Creamos el vector y lo inicializamos con 0

ventas = [ ]

for i in range(VENDEDORES+1):

    ventas.append(0)

## EJEMPLO – TERCERA PARTE

*# Comenzamos la lectura y acumulación de datos*

```
vendedor = leervendedor(VENDEDORES)
while vendedor != -1:
    importe = int(input("Importe de la venta? "))
    ventas[vendedor] = ventas[vendedor]+importe
    vendedor = leervendedor(VENDEDORES)
```

*# Imprimir informe final*

```
for i in range(1, VENDEDORES+1):
    print("El vendedor", i, "vendió $", ventas[i])
```



# EJEMPLO – PROGRAMA COMPLETO

```
def leervendedor(maximo):  
    n = int(input("Número de vendedor (-1 para terminar) "))  
    while (n!=-1 and (n<1 or n>maximo)):  
        print("*** Vendedor inválido ***")  
        n = int(input("Número de vendedor (-1 para terminar) "))  
    return n
```

VENDEDORES = 50

*# Creamos el vector y lo inicializamos con 0*

ventas = []

for i in range(VENDEDORES+1):

ventas.append(0)

*# Comenzamos la lectura y acumulación de datos*

vendedor = leervendedor(VENDEDORES)

while vendedor!=-1:

importe = int(input("Importe de la venta? "))

ventas[vendedor] = ventas[vendedor]+importe

vendedor = leervendedor(VENDEDORES)

*# Imprimir informe final*

for i in range(1,VENDEDORES+1):

print("El vendedor",i,"vendió \$ ",ventas[i])

# ORDENAR POR BURBUJEO, 2 LISTAS EN FORMA PARALELA

```
def ordenarlistas(leg, nota):
```

```
    desordenado = True
```

```
    while desordenado:
```

```
        desordenado = False
```

```
        for i in range(len(nota)-1):
```

```
            # Intercambiamos las listas en paralelo para no perder la relación  
            entre sus elementos
```

```
            if nota[i]>nota[i+1]:
```

```
                aux = nota[i]
```

```
                nota[i] = nota[i+1]
```

```
                nota[i+1] = aux
```

```
                aux = leg[i]
```

```
                leg[i] = leg[i+1]
```

```
                leg[i+1] = aux
```

```
            desordenado = True
```

# Ejercitación

- Práctica 7 y 9 : Ejercicios Completa