

# Guía de Trabajos Prácticos

## Unidad 04: Ordenación

### Ejercicio 4.01

Implementá ordenamiento por selección

### Ejercicio 4.02

Implementá ordenamiento por inserción.

Una vez implementado haga los cambios necesarios para que ordene en orden decreciente

### Ejercicio 4.03

Implementá ordenamiento por burbuja

### Ejercicio 4.04

Implementá mergesort

### Ejercicio 4.05

Implementá quicksort

### Ejercicio 4.06

Implementá ordenación por inserción en una lista enlazada simple



### Ejercicio 4.07

Creá un programa que encuentre el número que falta en un array

Ejemplo:

input

a = {5,1,4,3,7,6}

output

2

### Ejercicio 4.08

Ordenar valores por su frecuencia

Ejemplo

Input 3, 4, 3, 3, 5, 2, 4, 6

Output 3, 4, 2, 5, 6

Pista: debe usar más de un array

### Ejercicio 4.09

Ordenar un array usando pilas

Ayuda: pseudocodigo

declarar pila1, pila2, temp

meter el array en pila1

mientras pila1 no esté vacía{

temp = pila1.pop()

mientras (pila2 no esté vacía y temp < pila2.arriba){

pila1.push(pila2.pop())

}

pila2.push(temp)]

}

El array ordenado está en pila2

### Ejercicio 4.10

Dado un conjunto de valores enteros en un array, escribí un programa que obtenga el par (o los pares) de números más cercanos entre sí.

### **Ejercicio 4.11**

Escribí una función que ordene una lista enlazada simple usando el método de la burbuja