

# Día 2 · Algoritmos de búsqueda y ordenamiento

## Ejercicio 4: Búsqueda binaria de productos

**Versión básica** Crea una función que reciba un arreglo ordenado de códigos de productos y un código de búsqueda. La función debe devolver true si el código existe, false en caso contrario.

**Versión extendida** Implementa una búsqueda binaria que, además de detectar si el código existe, indique su posición y los pasos realizados. Añade visualización del rango evaluado en cada iteración.

### Definiciones clave

Concepto	Definición
Arreglo ordenado	Secuencia de elementos organizada (ascendente o descendente). Requisito de búsqueda binaria.
Búsqueda binaria	Algoritmo que divide el arreglo en mitades sucesivas, buscando eficientemente en $\log_2(n)$ pasos.
Comparador	Mecanismo de comparación entre valores, necesario para determinar si avanzar izquierda o derecha.

### Ejemplos

- Entrada básica:

```
js
```

```
buscarCodigo(["A01", "B02", "C03", "D04"], "C03"); // true
buscarCodigo(["A01", "B02", "C03", "D04"], "X99"); // false
```

- Entrada extendida: Resultado: "Código encontrado en posición 2. Iteraciones: 2"

# Ejercicio 5: Ordenamiento por burbuja optimizado

**Versión básica** Crea una función que ordene un arreglo de números enteros usando el algoritmo Bubble Sort.

**Versión extendida** Mejora la función para que se detenga si no hay intercambios en una iteración (optimización). Añade reporte del número de pasadas realizadas y visualiza el arreglo paso a paso.

## Definiciones clave

Concepto	Definición
Bubble Sort	Algoritmo de ordenamiento que compara pares adyacentes y los intercambia si están en orden incorrecto.
Optimización por parada temprana	Técnica que finaliza el algoritmo si se detecta que el arreglo ya está ordenado antes de completar todas las pasadas.
Intercambio (Swap)	Acción de cambiar dos elementos de lugar en un arreglo.

## Ejemplos

- Entrada básica:

```
js
ordenar([5, 3, 8, 1]); // Salida: [1, 3, 5, 8]
```

- Entrada extendida:

```
js
ordenarOptimizado([2, 4, 3, 1]); // Salida: [1, 2, 3, 4]
Reporte: "Pasadas: 3 – Intercambios totales: 5"
```

# Ejercicio 6: Detección de duplicados

**Versión básica** Implementa una función que reciba un arreglo de enteros y retorne true si hay duplicados o false si todos los valores son únicos.

**Versión extendida** Además de detectar duplicados, retorna un arreglo con los valores duplicados encontrados y el número de repeticiones. Usa estructuras como Map o Set.

## Definiciones clave

Concepto	Definición
Duplicado	Valor que aparece más de una vez en una colección de datos.
Set (Conjunto)	Estructura que permite almacenar elementos únicos, ideal para detección de repetidos.
Map (Diccionario)	Estructura que asocia claves con valores, útil para contar frecuencia de apariciones.

## Ejemplos

- Entrada básica:

```
js
tieneDuplicados([1, 2, 3, 4, 2]); // true
tieneDuplicados([5, 6, 7]);      // false
```

- Entrada extendida:

```
js
detectarDuplicados([2, 3, 2, 4, 3, 2]);
// Salida: ["2", "3"]
// Reporte: {"2": 3, "3": 2}
```