

## Algoritmo de búsqueda binaria

El algoritmo de búsqueda binaria nos permite buscar rápidamente un elemento en un array ordenado. Su rendimiento es del orden de  $\log N$ , en lugar de  $N$  (en el caso de búsqueda secuencial).

La versión iterativa de este algoritmo es:

```
public class BusquedaBinaria {

    public static void main(String[] args) {
        int[] lista = { 5, 7, 10, 20, 35, 37, 40, 42 };
        int x = 42;

        System.out.println("Resultado: " + busqueda(lista, x));
    }

    /**
     * Método que aplica el algoritmo de búsqueda binaria en un array de
     * enteros.
     *
     * @param lista
     *         Array con valores enteros.
     * @param x
     *         Número que hay que buscar en el array lista[].
     * @return Posición del elemento x en el array lista[] ó -1 si no se
     *         encuentra.
     */
    public static int busqueda(int[] lista, int x) {

        int inicio = 0;
        int fin = lista.length - 1;
        int medio;

        while (inicio <= fin) {
            medio = (inicio + fin) / 2;

            if (lista[medio] < x) {
                inicio = medio + 1;
            } else if (lista[medio] > x) {
                fin = medio - 1;
            } else {
                return medio;
            }
        }

        // Si inicio > fin entonces el elemento x no está en el array.
        return -1;
    }
}
```

La versión recursiva de este algoritmo es:

```
public class BusquedaBinariaRecursiva {

    public static void main(String[] args) {
        int[] lista = { 2, 4, 6, 8, 10, 12, 14, 16, 18, 20 };

        int x = 4;

        System.out.println("Resultado: "
            + busqueda(lista, 0, lista.length - 1, x));
    }

    /**
     * Método que aplica el algoritmo de búsqueda binaria en un array de
     * enteros, de manera recursiva.
     *
     * @param lista
     *         Array con valores enteros.
     * @param inicio
     *         Buscamos x desde la posición "inicio".
     * @param fin
     *         Buscamos x hasta la posición "fin".
     * @param x
     *         Número que hay que buscar en el array lista[].
     * @return Posición del elemento x en el array lista[] ó -1 si no se
     *         encuentra.
     */
    public static int busqueda(int[] lista, int inicio, int fin, int x) {

        if (inicio > fin) {
            // Caso base: El valor x no está en el array.
            return -1;
        }

        // Se calcula la posición central entre los dos índices de búsqueda.
        int medio = (inicio + fin) / 2;

        if (lista[medio] > x) {
            // Caso recursivo: si el valor es menor que la posición que se ha
            // mirado entonces hay que seguir buscando por la derecha del
            array.
            return busqueda(lista, inicio, medio - 1, x);
        } else if (lista[medio] < x) {
            // Caso recursivo: si el valor es mayor que la posición que se ha
            // mirado entonces hay que seguir buscando por la izquierda del
            // array.
            return busqueda(lista, medio + 1, fin, x);
        } else {
            // Caso base: Se ha encontrado el elemento x en la posición pos.
            return medio;
        }
    }
}
```