

## PRÁCTICA 2. ANÁLISIS Y DISEÑO DE ALGORITMOS. DIVIDE Y VENCERÁS

María Peinado Toledo. Doble Grado Ingeniería Informática y Matemáticas

08/11/2021

// ALUMNO: María Peinado Toledo 77666802B

// GRUPO: Grupo D

//

```
public class OrdenacionRapida extends Ordenacion {

    public static <T extends Comparable<? super T>> void ordenar(T v[]) {
        ordRapidaRec(v, 0, v.length-1);
    }

    // Debe ordenar ascendentemente los primeros @n elementos del vector @v con
    // una implementación recursiva del método de ordenación rápida.
    public static <T extends Comparable<? super T>> void ordRapidaRec(T v[], int izq, int der) {
        if (izq < der) {
            int p = partir(v, v[izq], izq, der);
            ordRapidaRec(v, izq, p-1);
            ordRapidaRec(v, p + 1, der);
        }
    }

    public static <T extends Comparable<? super T>> int partir(T v[], T pivote, int izq, int der) {
        int i = izq;
        int j = der;

        while (i < j){
            while(v[j].compareTo(pivote) > 0 && j > 0){
                j--;
            }
            while(v[i].compareTo(pivote) <= 0 && i < v.length-1){
                i++;
            }
            if (i < j){
                Ordenacion.intercambiar(v,i,j);
            }
        }
        Ordenacion.intercambiar(v,izq,j);

        return j;
    }
}
```

En resumen, lo que hacen estos métodos, es seleccionar un pivote con el cual van a comparar los elementos a izquierda y derecha de él, poniendo a la izquierda los menores y a la derecha los mayores e iguales. Una vez el array está dividido se vuelve a hacer el mismo proceso con los arrays a izquierda y derecha del pivote y así se consigue la completa ordenación de él.

```
// ALUMNO: María Peinado Toledo 77666802B
// GRUPO: Grupo D
////////////////////////////////////

public class OrdenacionRapidaBarajada extends OrdenacionRapida {

    // Implementación de QuickSort con reordenación aleatoria inicial (para comparar tiempos experimentalmente)
    public static <T extends Comparable<? super T>> void ordenar(T v[]) {
        barajar(v);
        ordRapidaRec(v, 0, v.length - 1);
    }

    // reordena aleatoriamente los datos de un vector
    private static <T> void barajar(T v[]) {
        for (int i = v.length - 1; i > 0; i--) {
            int j = aleat.nextInt(i+1);
            intercambiar(v, i, j);
        }
    }
}
}
```

Esta clase es casi igual que la anterior simplemente que antes de ordenar el array se baraja inicialmente con un for que recorre el array y un aleatorio.