

# Ejercicio 1

El ejercicio consiste en construir en Java una clase para manejo de números Quebrados que contará con los siguientes métodos:

```

class Quebrado implements Cloneable {

    /*----- Constructores ----- */

    public Quebrado(int n, int d) {
    }

    /* Cuando no se proporciona, el denominador es 1 por defecto. */
    Quebrado(int n) {
    }

    /*----- Métodos de consulta -----*/

    /* Devuelve el numerador */
    public int num() {
    }

    /* Devuelve el denominador */
    public int den() {
    }

    /* Los siguientes métodos calculan el producto, la suma, la resta y la división */
    public Quebrado por(Quebrado q) {
    }

    public Quebrado sum(Quebrado q) {
    }

    public Quebrado res(Quebrado q) {
    }

    public Quebrado div(Quebrado q) {
    }

    /* Decide si dos quebrados son "iguales". Nota: 2/3 es igual a 4/6. */
    public boolean equals(Quebrado q) {
    }

    /* Calcula el maximo comun divisor de 2 enteros. Se supone que N es mayor que D. */
    public static int mcd(int N, int D) {
    }

    /*----- Métodos de modificación -----*/
    /* Cambia los valores del quebrado actual para que pasen a ser numerador n y denominador d */
    public void set(int n, int d) {
    }

    /* Simplifica el quebrado actual para que numerador y denominador no sean divisibles
    * por una cantidad mayor que uno.
    */
    public void simplifica() {
    }

    public String toString() {
    }

    protected Object clone() {
    }

    /*----- Parte privada: atributos -----*/

```

```
private int n,d;  
} // Class Quebrado
```

Para el manejo de quebrados negativos se tomará como criterio almacenar internamente el signo siempre en el numerador. De este modo, el quebrado  $3/(-4)$  se almacenará siempre como  $(-3)/4$ .

**Guión:**

1. Construir la clase en Java y hacer algunas pruebas.
2. Diseñar pre y post condiciones para todos los métodos (salvo toString y clone).
3. Comprobar las pre y postcondiciones usando JML.
4. Añadir una invariante de clase: el denominador nunca puede tomar el valor cero.
5. Añadir invariante de bucle en el método mcd.
6. Se supone ahora que el método división debe lanzar una excepción propia llamada QuebradoException cuando el divisor es cero. Cambiar el programa y las aserciones de forma adecuada.