

T2.3. Gestión de hilos

Con los hilos se pueden efectuar diversas operaciones que sean de utilidad al programador (y al administrador de sistemas a veces).

Por ejemplo, un hilo puede tener un nombre. Si queremos asignar un nombre a un hilo podemos usar el método `setName("Nombre que sea")`. También podemos obtener un objeto que represente el hilo de ejecución con `currentThread` que nos devolverá un objeto de la clase `Thread`.

Otra operación de utilidad al gestionar hilos es indicar la prioridad que queremos darle a un hilo. En realidad esta prioridad es indicativa, el sistema operativo no está obligado a respetarla aunque por lo general lo hacen. Se puede indicar la prioridad con `setPriority(10)`. La máxima prioridad posible es `MAX_PRIORITY`, y la mínima es `MIN_PRIORITY`.

Cuando lanzamos una operación también podemos usar el método `Thread.sleep(numero)` y poner nuestro hilo «a dormir».

Cuando se trabaja con prioridades en hilos **no hay garantías de que un hilo termine cuando esperemos**.

Podemos terminar un hilo de ejecución llamando al método `join`. Este método devuelve el control al hilo principal que lanzó el hilo secundario con la posibilidad de elegir un tiempo de espera en milisegundos.

El siguiente programa ilustra el uso de estos métodos:

```
class Calculador implements Runnable{
    @Override
    public void run() {
        int num=0;
        while(num<5){
            System.out.println("Calculando...");
            try {
                long tiempo=(long) (1000*Math.random()*10);
                if (tiempo>8000){
                    Thread hilo=Thread.currentThread();
                    System.out.println(
                        "Terminando hilo:" +
                        hilo.getName()
                    );
                    hilo.join();
                }
                Thread.sleep(tiempo);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            System.out.println("Calculado y reiniciando.");
            num++;
        }
        Thread hilo=Thread.currentThread();
        String miNombre=hilo.getName();
        System.out.println("Hilo terminado:"+miNombre);
    }
}

public class LanzadorHilos {
    public static void main(String[] args) {
        Calculador vHilos[]=new Calculador[5];
        for (int i=0; i<5;i++){
            vHilos[i]=new Calculador();
            Thread hilo=new Thread(vHilos[i]);
            hilo.setName("Hilo "+i);
            if (i==0){
```

```
        hilo.setPriority(Thread.MAX_PRIORITY);
    }
    hilo.start();
}
}
```