

FINALIZACIÓN E INTERRUPCIÓN DE HILOS

El método **interrupt()** se utiliza para provocar la interrupción de un hilo, lo que implica enviar una señal en un instante específico. Esta señal puede ser aprovechada mediante código diseñado para responder a la interrupción, permitiendo al hilo realizar acciones definidas, como finalizar un bucle, entre otras posibles operaciones.

```
public void run() {  
    //Infinite loop until interruption  
    while(!Thread.interrupted()) {  
        //Stuff to do  
    }  
}
```

```
public void run() {  
    for(int i = 0; i<5 && !Thread.interrupted(); i++) {  
        //Stuff to do  
    }  
}
```

Para probar esto, crea un programa en Java con las siguientes características:

- ☐ Crea una clase que herede de la clase Thread.
 - El nombre de la clase será “*NombreEstudianteHilo*”.
 - El método run imprimirá cada dos segundos:
 - El nombre del hilo.
 - El número de la iteración.
 - El hilo termina después de 5 iteraciones.
 - El hilo termina si el método “interrupt” se ha llamado.
 - Cuando el hilo finalice, se imprimirá por pantalla si la terminación fue la normal o la forzada.
- ☐ En la clase principal (main):
 - Lanza tres hilos con distinta prioridad con el nombre “*NombreEstudianteHilo*”. Cada uno con un nombre diferente de la serie de televisión “El equipo A”.
 - Crea los hilos con distinta prioridad.
 - Llama al método **isAlive()** para comprobar si un hilo está vivo o no en diferentes momentos.
- ☐ Contesta a la siguiente pregunta:
 - ¿hay algún problema cuando interrumpimos un hilo usando el método interrupt()?

- Elabora otro fragmento de código similar para la finalización de un hilo empleando un indicador (**flag**). Un indicador es una variable booleana que experimenta un cambio de estado y se encuentra en la condición de un bucle 'while'. De este modo, cuando el indicador se modifica, se detiene la ejecución del bucle.

```
public void run() {  
    while(!flag) {  
        //Stuff to do  
    }  
    if (flag)  
        System.out.println( Thread.currentThread().getName() +  
                             " thread manually terminated!!");  
}
```

Para cambiar el valor del **flag** se suele implementar un método sincronizado que cambia su valor.

```
//METHOD  
public synchronized void terminate(){  
    this.flag = true;  
}
```