

Javier Romero Castro DNI: 44659463-H

Nara Menor Sierra DNI: 44479308-E

Esteban Puentes Silveira DNI: Y1883497H

ACTIVIDAD 1 - CDI

1.1. Introducción a los hilos en Java

1.1.1 Una aplicación que crea una instancia de un hilo debe aportar el código que va a ser ejecutado en ese hilo. Esto se puede hacer de dos maneras:

- *Con un objeto Runnable.* La interfaz Runnable define un único método, “run”, creado para contener el código ejecutado en el hilo. El objeto Runnable se pasa al constructor de la clase Thread.
- *Subclase Thread.* La clase thread implementa por si sola la interfaz Runnable, mientras que su método “run” no hace nada. Una aplicación puede ser una subclase de thread, aportando su propia implementación de la clase “run”.

1.1.2 Es preferible utilizar la interfaz Runnable ya que de esta manera podremos extender otra clase, sin embargo si utilizamos extend Thread no se podrán extender sobre más ya que es una limitación de java.

1.1.3 Thread.sleep(1000);

1.1.4 Se activan 3 hilos, el main y los dos hilos programados. Para conocer los hilos activos del momento se utiliza el método activeCount().

1.2. Creación de hilos múltiples

1.2.1 En el proyecto

1.2.2 En el proyecto

1.2.3 Asegurarnos de que todos los hilos terminan, esto se consigue con el método `join()`.

Si no nos aseguramos puede ocurrir que algún hilo no haya terminado, por lo que el recuento del tiempo sería erróneo.

1.2.4 El valor de los tiempos de cada uno de los hilos será menor que la global. La suma de estos valores permite comparar un programa con respecto a otro o incluso comprobar la velocidad de ese programa en otra máquina.

1.2.5 No se pueden distinguir, por lo que no se puede diferenciar entre el tiempo que tarda el programa en crear el hilo y su tiempo de ejecución.

1.2.6 Sí que cambia bastante porque tenemos que introducir un nuevo parámetro que determinará si se hará una operación matemática o simplemente se muestra algo por pantalla, y esto implica más tiempo ya que es necesario comprobarlo cada vez.

1.3. Control básico de múltiples hilos

1.3.1 El método `interrupt()` interrumpe la ejecución del hilo actual.

El hilo interrumpido debe ser compatible con su propia interrupción.

Se debe utilizar en con `try...catch` para que el hilo a interrumpir permite la excepción.

El método `stop()` obliga al hilo de detenerse, mientras que el método `suspend()` detiene el hilo momentáneamente y puede reanudarse en momentos futuros, si se reanuda. Es por eso, que ambos métodos no están recomendados y se usa `interrupt()`.

1.3.2 El método `interrupted()` es el método que interrumpe las acciones del hilo, mientras que el método `isInterrupted()` comprueba si el hilo está interrumpido o no, devuelve un booleano.

1.3.3 En el proyecto.