**P3 Semana 3 01/03/2018**

B) Si la variable utilizada simplemente se define como privada se creará una copia local de la variable para cada clase MiThread creada. Por lo que cada hilo sumará los números independientemente dando como resultado la suma alcanzada por cada hilo.

Por ejemplo, si sumo los números de 1 al 100 el resultado será 5050 en todos los hilos.

Sin embargo, si se define la variable como static podemos forzar a que esa variable sea un singleton y se comparta para todas las clases MiThread creadas.

Con esto conseguimos que todos los hilos accedan a la misma variabel y sumen de forma acumulativa por lo que el resultado total se incrementa conforme los hilos van sumando.

Si la clase MiThread extends Runnable, podríamos crear todas las instancias del hilo usando la misma instancia de la clase Runnable por lo que sí compartiríamos la misma clase.

C) Usando la clase ThreadLocal<Integer>, como esperábamos, el valor de las variables es único para cada clase por lo que nunca se comparte la variable entre los hilos.

D) El método Thread.interrupt() no interrumpe directamente la ejecución del hilo. Esto pasa porque en java la interrupción de un hilo únicamente puede hacerse desde el propio hilo. Al llamar a Thread.interrupt() desde el main simplemente pone a true el flag de interrupción del hilo. Una vez hecho esto desde el propio hilo que queremos parar debemos controlar el flag para parar el hilo como deseemos.