Ejercicio 1

Crea un programa que pida al usuario dos números *n1* y *n2*, posteriormente lanzará un hilo que mostrará los números existentes entre *n1* y *n2*. Haz que el hilo se suspenda de forma aleatoria entre 1 y 1000 milisegundos cada vez que muestre un número por pantalla. Una vez lanzado el hilo el programa principal mostrará el mensaje "El hilo se ha lanzado". Resuelve este ejercicio de las siguientes formas: en primer lugar, creando una clase que implemente la interfaz *Runnable*, luego mediante una clase anónima y finalmente mediante el uso de expresiones *lambda*.

Ejemplo de ejecución:

```
Introduce n1: 2
Introduce n2: 4
El hilo se ha lanzado
2
3
4
```

Ejercicio 2

Realiza un programa que pida al usuario un número *n*. El programa lanzará *n* hilos, el primero se llamará "*Hilo 1*", el siguiente "*Hilo 2*" y así sucesivamente. Cada hilo generará un número aleatorio entre 1 y 100 y mostrará los números primos que existan entre 1 y el número elegido aleatoriamente. Junto con el número primo, se debe mostrar el nombre del hilo. Se realizará una pausa aleatoria entre 500 y 1000 milisegundos tras mostrar cada número. Para crear los hilos utiliza una clase que implemente la interfaz *Runnable*.

Ejemplo de ejecución:

```
Introduce el número de hilos a crear: 2
Hilo 1: Mostrando primos hasta el 45
Hilo 2: Mostrando primos hasta el 72
Hilo 1: 2
Hilo 2: 2
Hilo 2: 3
Hilo 1: 3
Hilo 2: 5
Hilo 1: 5
Hilo 2: 7
…
```

Ejercicio 3

Modifica el programa anterior para que, tras lanzar los hilos, el programa principal muestre cada segundo el ID, nombre y estado de cada hilo, hasta que todos los hilos hayan finalizado.

Ejemplo de ejecución:

```
Introduce el número de hilos a crear: 2
12 Hilo 1 RUNNABLE
13 Hilo 2 BLOCKED
Hilo 1: Mostrando primos hasta el 81
Hilo 2: Mostrando primos hasta el 46
Hilo 1: 2
Hilo 2: 2
Hilo 2: 3
Hilo 1: 3
12 Hilo 1 TIMED_WAITING
13 Hilo 2 TIMED_WAITING
Hilo 1: 5
Hilo 2: 5
12 Hilo 1 TIMED_WAITING
13 Hilo 2 TIMED_WAITING
…
```