

Ejercicio 1

Después de haber comprobado el código y haber ejecutado y compilado todo con éxito esta es mi comprobación:

La clase main muestra como se crean tres objetos con la clase “Thread1” (Hilo1) y en cada clase hay un mensaje, un retardo y un contador de veces

The screenshot shows the IntelliJ IDEA interface. In the top navigation bar, there are tabs for Main.java, Main2.java, Thread2.java, and Thread1.java. The Main.java tab is active. The code editor displays the following Java code:

```
public class Main {
    public static void main(String[] args) {
        /*
         * Creación de los hilos
         */
        Thread1 t1 = new Thread1( mensaje: "Soy A", retardo: 100, veces: 10);
        Thread1 t2 = new Thread1( mensaje: "Soy B", retardo: 150, veces: 15);
        Thread1 t3 = new Thread1( mensaje: "Soy C", retardo: 300, veces: 5);
        /*
         * Inicialización de los hilos
         */
        t1.start();
        t2.start();
        t3.start();
        /*
         * Esperamos a que terminen los hilos
         */
        try {
            t1.join();
            t2.join();
        }
    }
}
```

The code is annotated with comments explaining the creation of threads and their initialization. The output window at the bottom shows the execution results:

```
Soy B
Soy C
Soy A
Soy A
Soy B
Soy A
Soy C
Soy A
```

The run configuration dropdown shows "Main". The status bar at the bottom right indicates "9:8 CRLF UTF-8 4 spaces".

Después de crear los objetos, estos se ejecutan cada uno de ellos nombrándolos y haciendo un .start (Ej: t1.start();).

Esto hace que se vaya a la clase Thread y específicamente al método run()

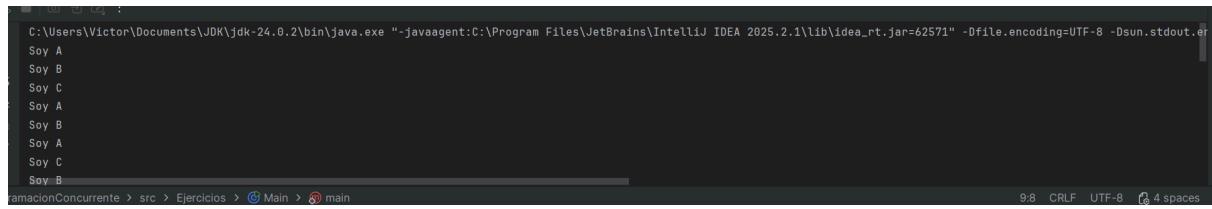
```
9 @↑    public void run() { no usages
0     |    for(int i = 1; i <= veces; i++) {
1     |        System.out.println(mensaje);
2     |        try {
3     |            /*
4     |             * Paraliza la ejecución del hilo
5     |             * [retardo] milisegundos
6     |            */
7     |            Thread.sleep(retardo);
8     |        } catch (InterruptedException e) {
9     |            e.printStackTrace();
0     |        }
1     |
2     }
3 }
```

En el método run() se crea un bucle if la cual se crea una variable “i” con un valor de 1 que mientras “i” es menor al número de veces que tiene asignado el objeto creado en el main (t1, t2 o t3) se enseñará el mensaje que tiene ligado el objeto, en el caso de “t1” se mostrará “Soy A” pero al ser hilos se mostrarán de manera aleatoria la cual se mostrará el mensaje de “t1”, “t2” y “t3” de manera aleatoria según el numero de veces que tenga asignado. Cada objeto tiene ligado un retardo que es la espera en milisegundos que tardará en mostrar ese mensaje, en el caso de “t1” tiene un retardo de 100 milisegundos.

```
Thread1 t1 = new Thread1( mensaje: "Soy A", retardo: 100, veces: 10);
```

A continuación se muestra que la respuesta es completamente aleatoria:

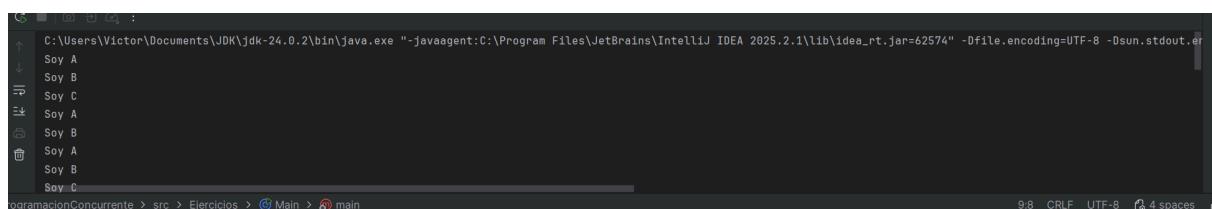
Ejemplo 1:



```
C:\Users\Victor\Documents\JDK\jdk-24.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=62571" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
Soy A
Soy B
Soy C
Soy A
Soy B
Soy C
Soy B
```

9:8 CRLF UTF-8 4 spaces

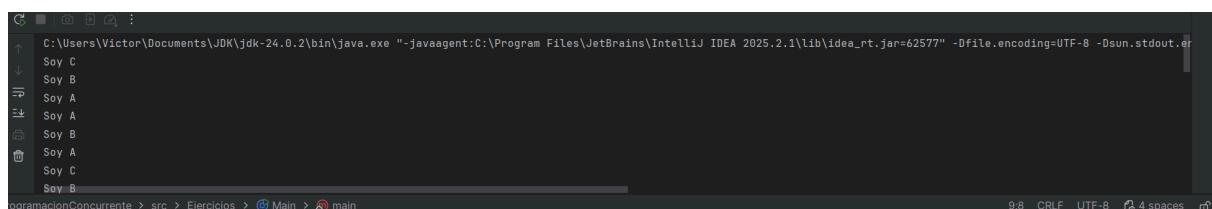
Ejemplo 2:



```
C:\Users\Victor\Documents\JDK\jdk-24.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=62574" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
Soy A
Soy B
Soy C
Soy A
Soy B
Soy A
Soy B
Soy C
Soy B
```

9:8 CRLF UTF-8 4 spaces

Ejemplo 3:



```
C:\Users\Victor\Documents\JDK\jdk-24.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2025.2.1\lib\idea_rt.jar=62577" -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8
Soy C
Soy B
Soy A
Soy A
Soy B
Soy A
Soy C
Soy B
```

9:8 CRLF UTF-8 4 spaces