

PRÁCTICA 05 2DAM 26/01/2024

Multihilos

Manuel Martín Santamaría

Índice

- ❖ [Pag 3] - [Indicaciones de entrega](#)
- ❖ [Pag 4] - [Actividades](#)
 - [Pag 4] - [Actividad 1](#)
 - [Pag 5] - [Actividad 2](#)
 - [Pag 5] - [Actividad 3](#)
 - [Pag 6] - [Actividad 11](#)

Indicaciones de entrega

Una vez realizada la tarea elaborarás un único fichero .zip donde figuren las respuestas correspondientes. El envío se realizará a través de la plataforma de la forma establecida para ello, y el archivo se nombrará siguiendo las siguientes pautas:

apellido1_apellido2_nombre_SIGxx_Tarea

Asegúrate que el nombre no contenga la letra ñ, tildes ni caracteres especiales extraños. Así por ejemplo la alumna Begoña Sánchez Mañas para la primera unidad del MP de DAW, debería nombrar esta tarea como...

sanchez_manas_begona_PSP_Tarea01

Cada ejercicio ha de desarrollarse en un documento distinto, siguiendo los nombres de ej1, ej2,...

Actividades

[Actividad 1]

¿Qué sucede?

Los hilos se ejecutan de forma concurrente, como se puede apreciar en la salida.

¿Es lo que esperabas?

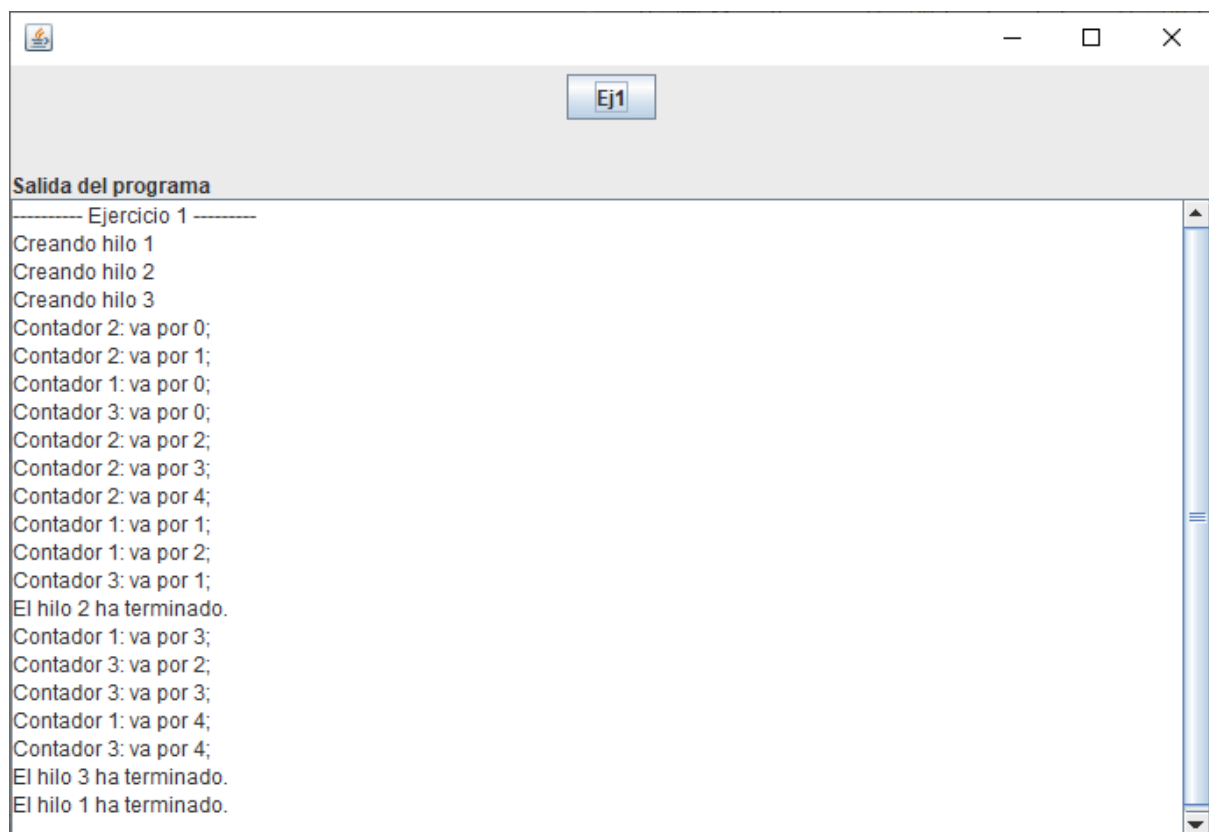
Si.

Haz una captura de pantalla.

Código:

```
// Ejecutar en paralelismo o de forma concurrente
// Se mezcla la salida
private static void ejecutaConcurrenemente() {
    hilo1.start();
    hilo2.start();
    hilo3.start();
}
```

Salida:



¿Se puede modificar el comportamiento de los hilos para que realice las acciones en el orden esperado?

Sí.

¿Cómo?

Para solucionarlo, podemos forzar la ejecución secuencial, por ejemplo:

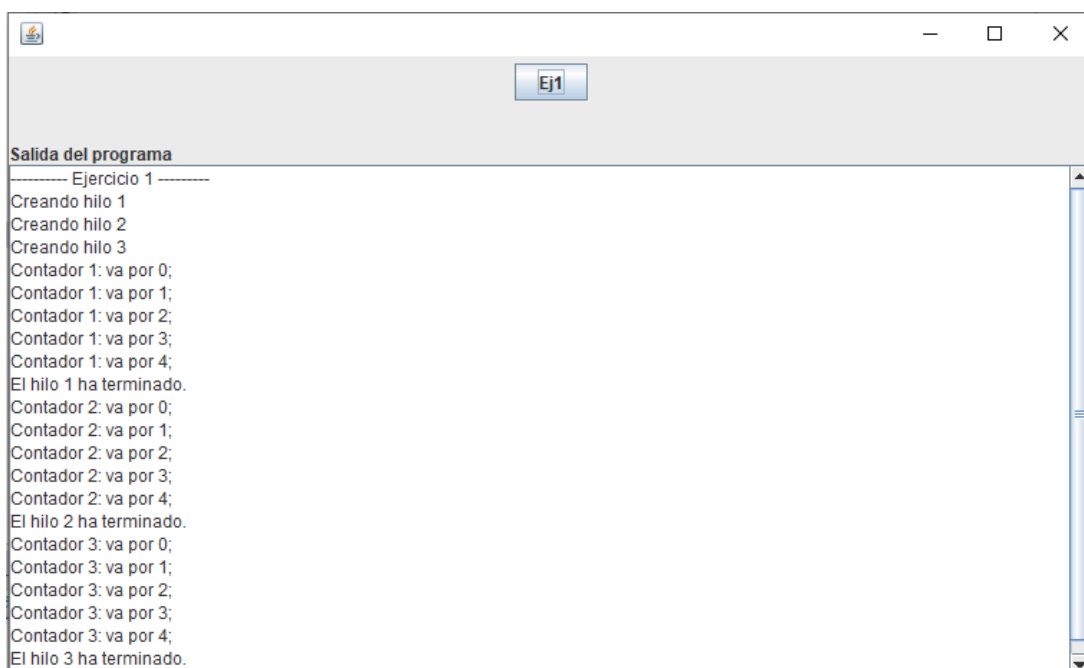
Código:

```
// Ejecutar secuencialmente
// Espera a que acabe uno para continuar con la ejecución del siguiente
private static void ejecutaSecuencialmente() {
    try {
        hilo1.start();
        hilo1.join();

        hilo2.start();
        hilo2.join();

        hilo3.start();
        hilo3.join();
    } catch (InterruptedException ie) {
        ie.printStackTrace();
    }
}
```

Salida:



```
Salida del programa
----- Ejercicio 1 -----
Creando hilo 1
Creando hilo 2
Creando hilo 3
Contador 1: va por 0;
Contador 1: va por 1;
Contador 1: va por 2;
Contador 1: va por 3;
Contador 1: va por 4;
El hilo 1 ha terminado.
Contador 2: va por 0;
Contador 2: va por 1;
Contador 2: va por 2;
Contador 2: va por 3;
Contador 2: va por 4;
El hilo 2 ha terminado.
Contador 3: va por 0;
Contador 3: va por 1;
Contador 3: va por 2;
Contador 3: va por 3;
Contador 3: va por 4;
El hilo 3 ha terminado.
```

[Actividad 2]

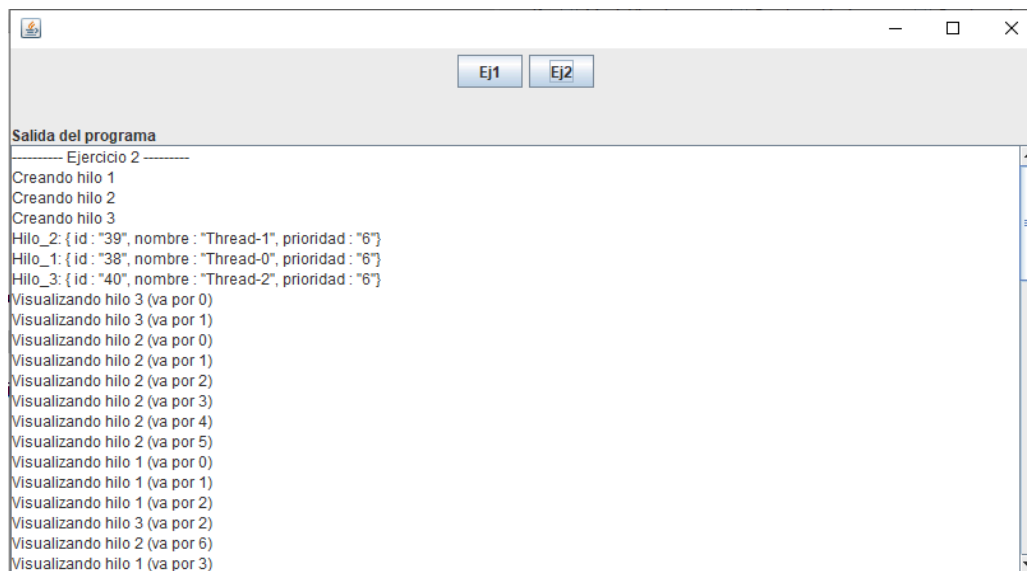
¿Es el resultado el mismo en todas las ejecuciones?

Cada ejecución cambia el orden de la salida al ejecutarse de manera concurrente.

Código:

```
// Ejecutar en paralelismo o de forma concurrente
// Se mezcla la salida
private static void ejecutaConcurrentemente() {
    hilo1.start();
    hilo2.start();
    hilo3.start();
}
```

Salida:



¿Se puede apreciar diferencia en el orden de los mensajes visualizados?

Establecer diferentes prioridades a los hilos no parece estar afectando aparentemente al orden en la que se ejecutan, posiblemente debido a la existencia de diferentes colas en la cpu, ya que es un procesador i7 de decimotercera generación.

Código:

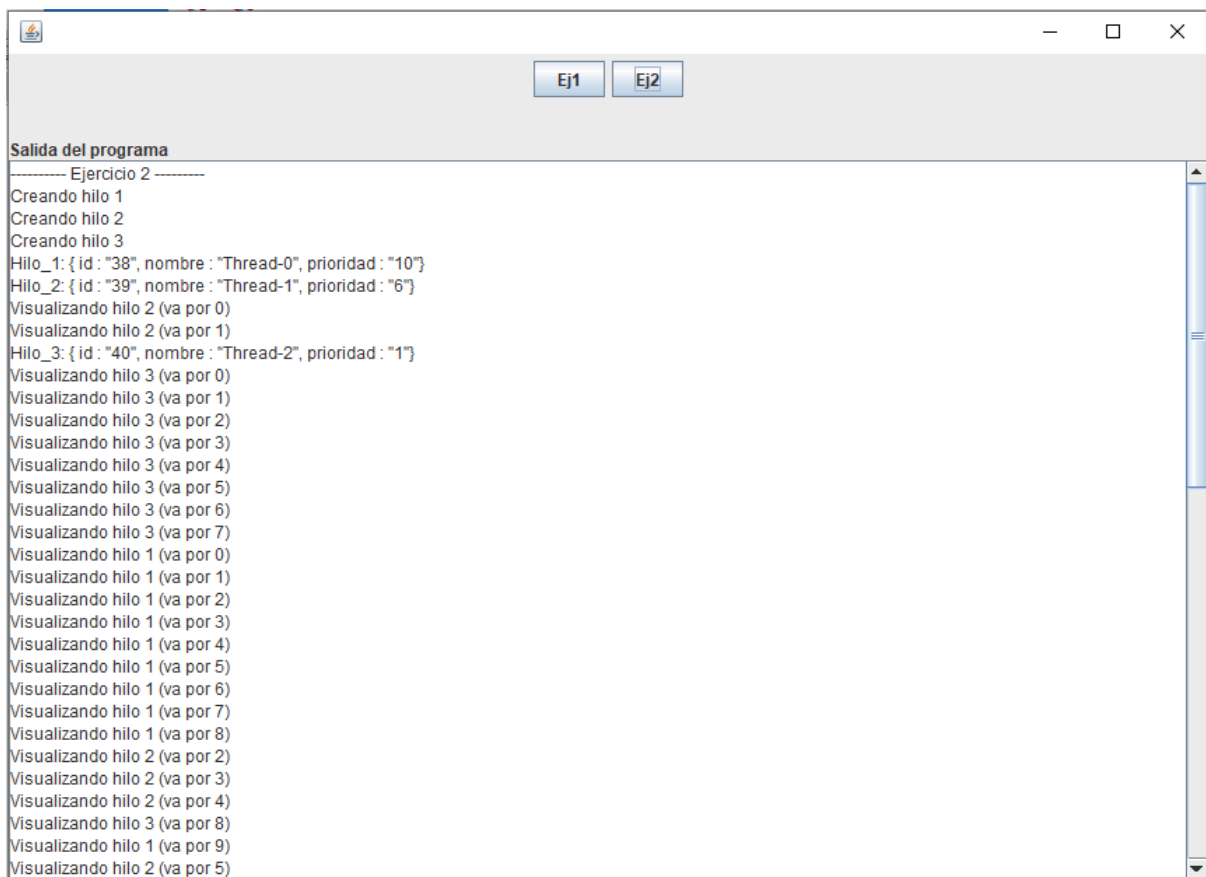
```
// Cambiando las prioridades sigue ejecutandose de manera concurrente
// Imagino que por las características de la cpu
private static void cambiaPrioridades() {

    hilo1.setPriority(newPriority:10);
    hilo2.setPriority(newPriority:6);
    hilo3.setPriority(newPriority:1);

    hilo1.start();
    hilo2.start();
    hilo3.start();

}
```

Salida:

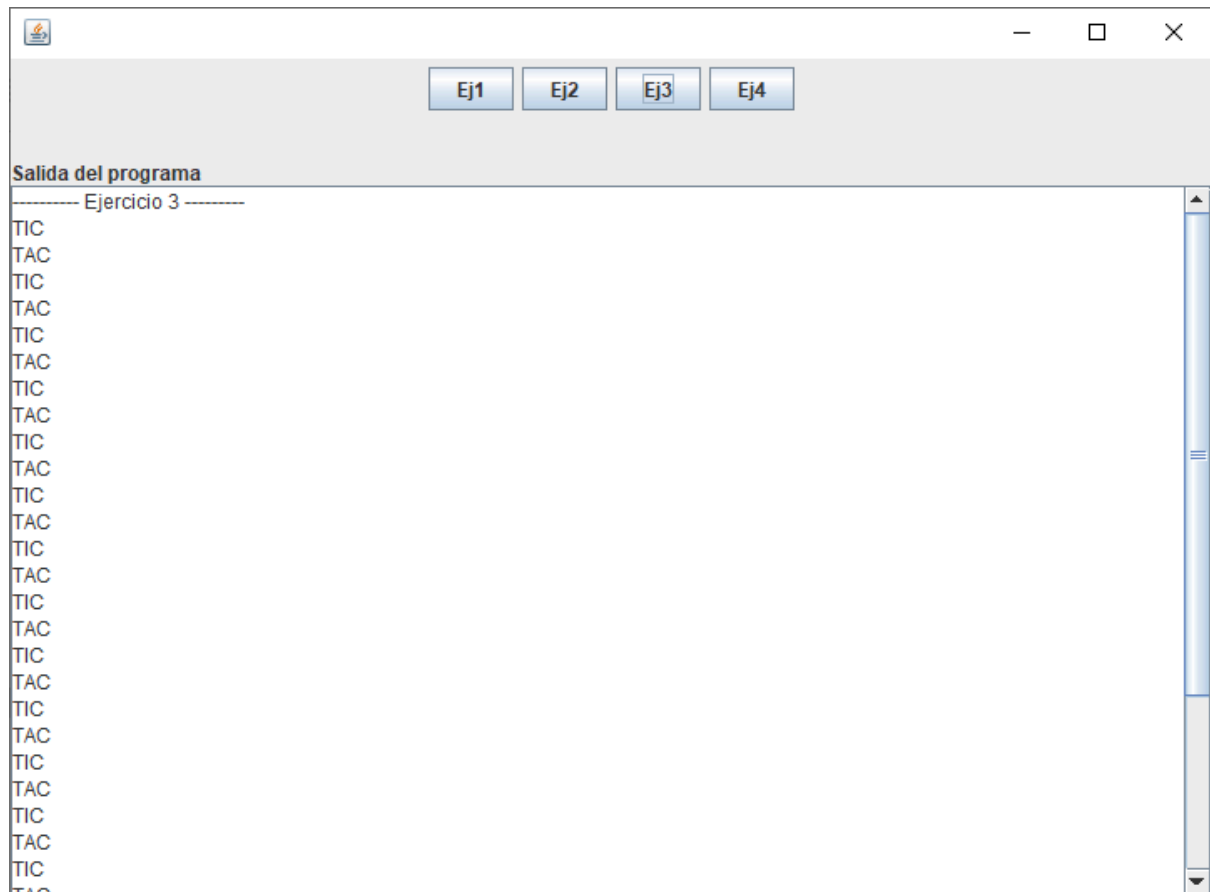


[Actividad 3]

¿Se visualizan los textos TIC y TAC de forma ordenada (es decir TICTAC TICTAC ...)?

SI

Salida



[Actividad 11]

¿Cómo podrías resolver el ejercicio de TIC TAC usando el modelo productor consumidor y los métodos sincronizados?

Empleando una cola Queue<String> en la cual, los productores de Tic y Tac vayan introduciendo strings y el consumidor vaya consumiendo de la cola siempre y cuando haya elementos en la cola.