

UNIDAD 2. PROGRAMACIÓN MULTITHREAD

PARTE 1. CUESTIONARIO

Crea un documento en Writer y nombralo como “CuestionesUD2_NombreApellidos”. Copia todas las preguntas y responde de manera razonada. Debes entregar el cuestionario en formato PDF en la tarea de Aules correspondiente.

- 1. ¿Qué es un programa?**
- 2. ¿Qué es un proceso?**
- 3. En informática, ¿qué significado tiene la palabra “conurrencia”?**
- 4. ¿Para qué se utilizan los lenguajes concurrentes?**
- 5. ¿Qué ventajas e inconvenientes tiene la programación concurrente?**
- 6. ¿Cómo define Dijkstra un programa concurrente?**
- 7. ¿Qué son los semáforos?**
- 8. ¿Cuál es el objetivo de la programación paralela en contraposición con la concurrente?**
- 9. ¿Todos los programas paralelos suelen ser concurrentes? ¿y al revés?**
- 10. En la programación paralela, ¿cómo se comunican dos procesos?**
- 11. En la programación concurrente, ¿cómo se comunican dos procesos?**
- 12. ¿Qué son los hilos de un proceso?**
- 13. ¿Qué diferencias existen entre los procesos y los hilos?**
- 14. ¿En qué estados puede estar un proceso? Realiza un diagrama**
- 15. ¿En qué estados puede estar un hilo? Realiza un diagrama**
- 16. ¿Todos los hilos comparten los mismos recursos?**
- 17. ¿Cómo podemos crear un hilo en Java? Pon un ejemplo**
- 18. ¿Qué método debemos implementar para definir lo que tiene que ejecutar el hilo?**
- 19. ¿Qué ocurre cuando ejecuto el método start()?**
- 20. ¿Para qué sirve el método “SetPriority()”?**
- 21. ¿Qué otros métodos existen para la clase “Thread” de Java? Realiza una tabla donde pongas el método y su función.**
- 22. ¿Qué es la interfaz “Runnable”? ¿Cuándo se usa?**

PARTE 2. EJERCICIOS

Realiza los siguientes ejercicios en Java. Crea un documento en Writer y nombralo como "EjerciciosUD2_NombreyApellidos". En este documento debes añadir para cada ejercicio una captura de pantalla del código y una captura de pantalla del resultado obtenido. Debes entregar el documento en formato PDF en la tarea de Aules correspondiente.

E1. Crea una clase llamada "PrimerHilo" cuya funcionalidad sea mostrar por pantalla un mismo mensaje 5 veces. Este número se le pasará desde el constructor.

Por pantalla debe salir lo siguiente:

```
En el hilo...0
En el hilo...1
En el hilo...2
En el hilo...3
En el hilo...4
```

E2. Crea una clase llamada "TresHilos". En el método main() se deben crear 3 hilos. La misión de un hilo, será visualizar un mensaje donde se muestre el nombre del hilo que se está ejecutando (*para ello, se debe usar el método getName() de la clase "Thread"*) y el contenido de un contador con valor "5".

Debe mostrar por pantalla el siguiente resultado:

```
Creando Hilo:Hilo 1
Creando Hilo:Hilo 2
Creando Hilo:Hilo 3
      Hilo:Hilo 1C=0
      Hilo:Hilo 1C=1
      Hilo:Hilo 1C=2
      Hilo:Hilo 1C=3
      Hilo:Hilo 1C=4
      Hilo:Hilo 2C=0
      Hilo:Hilo 2C=1
      Hilo:Hilo 2C=2
      Hilo:Hilo 2C=3
      Hilo:Hilo 2C=4
3 Hilos iniciados...
      Hilo:Hilo 3C=0
      Hilo:Hilo 3C=1
      Hilo:Hilo 3C=2
      Hilo:Hilo 3C=3
      Hilo:Hilo 3C=4
```

E2_1. Puede ocurrir que la ejecución del programa anterior no siempre muestre la misma salida ¿por qué?

EJ3. Realiza el mismo programa que para el Ejercicio 2, pero ahora debes crear 5 hilos y poner el contador con valor "2". Para ello crea una clase llamada "CincoHilos".

EJ4. En muchas ocasiones, dentro del método `run()` se puede ver un bucle infinito de forma que el hilo no termina nunca. ¿Cómo podríamos detener un hilo? ¿Qué opciones serían posibles?

EJ5. Aplica al Ejercicio 1 (PrimerHilo) la detención del hilo según tu respuesta en el ejercicio 3. Nombra a la nueva clase como “PrimerHiloStop”.

EJ6. Basándote en el Ejercicio 2 y en la clase “TresHilos”, realiza lo siguiente: define por un lado la clase hilo y por otro lado la clase que usa el hilo (principal). De esta manera, tendrás dos clases:

→ Clase del hilo: nombrala como “TresHilos_v2”

→ Clase que va a usar el hilo: nombrala como “UsaTresHilos_v2”

Para que funcione correctamente, primero debes compilar “TresHilos_v2”, después “UsaTresHilos_v2” y finalmente ejecutar “UsaTresHilos_v2”.

El resultado mostrado por pantalla debe ser el mismo que el del Ejercicio 1.