|  |
| --- |
| Tema 2 |
|  |
| Programación de servicios y procesos 2ºDAM  Rodrigo Tapiador Cano |

[1. Actividades de Comprobación 3](#_Toc190171186)

[1.1. ¿Qué es un thread o hilo? 3](#_Toc190171187)

[1.2. ¿Cuál de las siguientes no es una característica de los hilos de ejecución? 3](#_Toc190171188)

[1.3. ¿Qué clase permite crear hilos de ejecución en Java? 3](#_Toc190171189)

[2. Métodos y Interfaces en Java para Hilos de Ejecución 3](#_Toc190171190)

[2.1. ¿Qué método permite iniciar la ejecución de un hilo en Java? 3](#_Toc190171191)

[2.2. ¿Qué interfaz permite crear hilos de ejecución? 3](#_Toc190171192)

[2.3. ¿Qué permite Runnable que no permite Thread? 3](#_Toc190171193)

[3. Clases Relacionadas con la Programación de Hilos 4](#_Toc190171194)

[3.1. ¿Qué clase del paquete java.lang representa una tarea programable? 4](#_Toc190171195)

[3.2. ¿Qué descripción es la correcta para la interfaz ExecutorService? 4](#_Toc190171196)

[4. Sincronización y Gestión de Hilos 4](#_Toc190171197)

[4.1. ¿Qué clase permite limitar el número de hilos que acceden a una sección crítica? 4](#_Toc190171198)

[4.2. ¿Qué clase proporciona una estructura de datos sincronizada equivalente a un ArrayList? 4](#_Toc190171199)

[5. Manejo y Dependencias entre Hilos 5](#_Toc190171200)

[5.1. ¿Qué método permite detener un hilo hasta que otro termine su trabajo? 5](#_Toc190171201)

[5.2. Dados dos segmentos de código cuyas salidas se asignan a las mismas variables. ¿Qué tipo de dependencia se estaría produciendo? 5](#_Toc190171202)

[6. Problemas de Concurrencia y Sincronización 5](#_Toc190171203)

[6.1. ¿Qué tipo de problema de concurrencia sucede cuando dos hilos tienen diferentes valores para una misma variable? 5](#_Toc190171204)

[6.2. ¿Cómo se puede recuperar la ejecución de un hilo detenido con el método wait? 5](#_Toc190171205)

## Actividades de Comprobación¿Qué es un thread o hilo?

Respuesta: c) Una pequeña unidad de computación que se ejecuta dentro del contexto de un proceso.

Justificación: Los hilos (threads) son subprocesos que se ejecutan dentro del espacio de memoria del proceso principal y comparten recursos con él.

Referencia: [threads y procesos](https://codigofacilito.com/articulos/threads-procesos)

## ¿Cuál de las siguientes no es una característica de los hilos de ejecución?

Respuesta: a) Se puede ejecutar de manera independiente a los procesos del sistema.

Justificación: Los hilos no son independientes de los procesos; siempre dependen de un proceso para existir.

Referencia: [threads y procesos](https://codigofacilito.com/articulos/threads-procesos)

## ¿Qué clase permite crear hilos de ejecución en Java?

Respuesta: b) La clase Thread.

Justificación: La clase Thread es la clase principal en Java para crear y manejar hilos.

Referencia: [hilos](https://docs.oracle.com/javase/7/docs/api/java/lang/Thread.html)

# Métodos y Interfaces en Java para Hilos de Ejecución

## ¿Qué método permite iniciar la ejecución de un hilo en Java?

Respuesta: b) start.

Justificación: El método start es el que inicia la ejecución de un hilo llamando internamente al método run.

Referencia: [thread.start](https://docs.oracle.com/javase/tutorial/essential/concurrency/runthread.html)

## ¿Qué interfaz permite crear hilos de ejecución?

Respuesta: a) La interfaz Runnable.

Justificación: La interfaz Runnable se utiliza para definir una tarea que puede ser ejecutada por un hilo.

Referencia: [Runnable](https://docs.oracle.com/javase/8/docs/api/java/lang/Runnable.html)

## ¿Qué permite Runnable que no permite Thread?

Respuesta: c) Crear múltiples hilos basándose en un único objeto.

Justificación: La interfaz Runnable permite reutilizar la misma instancia de una tarea en múltiples hilos.

Referencia: [Runnable vs Thread](https://docs.oracle.com/javase/tutorial/essential/concurrency/runthread.html)

# Clases Relacionadas con la Programación de Hilos

## ¿Qué clase del paquete java.lang representa una tarea programable?

Respuesta: c) TimerTask.

Justificación: La clase TimerTask se utiliza para representar una tarea que puede ser programada para ejecución futura.

Referencia: [TimerTask](https://docs.oracle.com/javase/7/docs/api/java/util/TimerTask.html)

## ¿Qué descripción es la correcta para la interfaz ExecutorService?

Respuesta: a) Subinterface de Executor, permite gestionar tareas asíncronas.

Justificación: ExecutorService es una subinterfaz de Executor que proporciona métodos para gestionar y controlar la ejecución de tareas asíncronas.

Referencia: [ExecutorService](https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/ExecutorService.html)

# Sincronización y Gestión de Hilos

## ¿Qué clase permite limitar el número de hilos que acceden a una sección crítica?

Respuesta: c) Semaphore.

Justificación: Semaphore se utiliza para controlar el acceso a un recurso compartido por varios hilos, limitando el acceso a cierto número de hilos. La clase synchronized por otra parte, limita el acceso a una sección critica de la memoria.

Referencia: [Semaphore](https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/Semaphore.html)

## ¿Qué clase proporciona una estructura de datos sincronizada equivalente a un ArrayList?

Respuesta: c) CopyOnWriteArrayList.

Justificación: CopyOnWriteArrayList es una implementación de List que es segura para la concurrencia.

Referencia: [CopyOnWriteArrayList](https://docs.oracle.com/javase/7/docs/api/java/util/concurrent/CopyOnWriteArrayList.html)

# Manejo y Dependencias entre Hilos

## ¿Qué método permite detener un hilo hasta que otro termine su trabajo?

Respuesta: b) join.

Justificación: El método join se utiliza para hacer que un hilo espere a que otro hilo termine su ejecución.

Referencia: [join](https://docs.oracle.com/javase/tutorial/essential/concurrency/join.html)

## Dados dos segmentos de código cuyas salidas se asignan a las mismas variables. ¿Qué tipo de dependencia se estaría produciendo?

Respuesta: a) Dependencia de flujo.

Justificación: La dependencia de flujo ocurre cuando una instrucción depende del resultado de una instrucción anterior.

Referencia: [Dependencies](https://en.wikipedia.org/wiki/Data_dependency)

# Problemas de Concurrencia y Sincronización

## ¿Qué tipo de problema de concurrencia sucede cuando dos hilos tienen diferentes valores para una misma variable?

Respuesta: b) Condición de carrera.

Justificación: La condición de carrera ocurre cuando el resultado de la ejecución depende del orden de ejecución de los hilos.

Referencia: [Condición de carrera](https://keepcoding.io/blog/que-es-condicion-de-carrera-programacion/)

## ¿Cómo se puede recuperar la ejecución de un hilo detenido con el método wait?

Respuesta: c) Con el método notifyAll.

Justificación: El método notifyAll se utiliza para despertar a todos los hilos que están esperando en el mismo monitor.

Referencia: [wait and notify](https://docs.oracle.com/javase/tutorial/essential/concurrency/guardmeth.html)