

1. ¿Qué es un programa?

- Un programa informático es una secuencia de instrucciones, escritas para realizar una tarea específica en una computadora. Este dispositivo requiere programas para funcionar, por lo general, ejecutando las instrucciones del programa en un procesador central.

2. ¿Qué es un proceso?

- Un proceso se entiende por un programa en ejecución a través de sucesivas instrucciones.

3. En informática, ¿qué significado tiene la palabra “conurrencia”?

- Se refiere a las situaciones en las que dos o más procesos pueden coincidir en el acceso a un recurso compartido, requiere coordinarse en su ejecución.

4. ¿Para qué se utilizan los lenguajes concurrentes?

- Es la simultaneidad en la ejecución de múltiples tareas interactivas, es decir sirve para coordinar los recursos que comparten por todos los procesos.

5. ¿Qué ventajas e inconvenientes tienen la programación concurrente?

Ventajas

- Permite optimizar el uso de recursos en sistemas multiprocesador
- Fiable administración de los datos en sistemas con gran información
- Mejor aprovechamiento CPU
- Permite el desarrollo de aplicaciones que no son afectadas a tiempo real
- Permite compartir recursos entre tareas lentas y rápidas para que las lentas no retrasen a las rápidas
- Velocidad de ejecución
- Menor tiempo de respuesta

Inconvenientes

- Consumo de recursos cuando hay exceso de procesos
- Dificultad de desarrollo
- Dificultad de verificación
- Si se aplica mal lleva a resultados erróneos
- En programas las instrucciones van más lento

6. ¿Cómo define Dijkstra un programa concurrente?

7. ¿Qué son los semáforos?

- Es una variable especial que restringe y permite el acceso a los recursos compartidos en un entorno de multiprocesamiento.

8. ¿Cuál es el objetivo de la programación paralela en contraposición con la concurrente?

- Utilizar con los mismos recursos, que varios procesos se hagan simultáneamente, mientras que la concurrente no son simultáneamente.

9. ¿ Todos los programas paralelos suelen ser concurrentes? ¿y al revés?

- Si, pero al revés no.

10. En la programación paralela, ¿cómo se comunican dos procesos?

- En la programación paralela se comunican con el rendez-vous, semáforos, mensajes, etc.

11. En la programación concurrente, ¿cómo se comunican dos procesos?

- Comparten memoria, es decir los recursos.

12. ¿Qué son los hilos de un proceso?

- Son tareas pequeñas que ejecuta el sistema operativo.

13. ¿Qué diferencias existen entre los procesos y los hilos?

- El hilo está dentro de un proceso y que es más ligero que un proceso.

14. ¿En qué estados puede estar un proceso? Realiza un diagrama

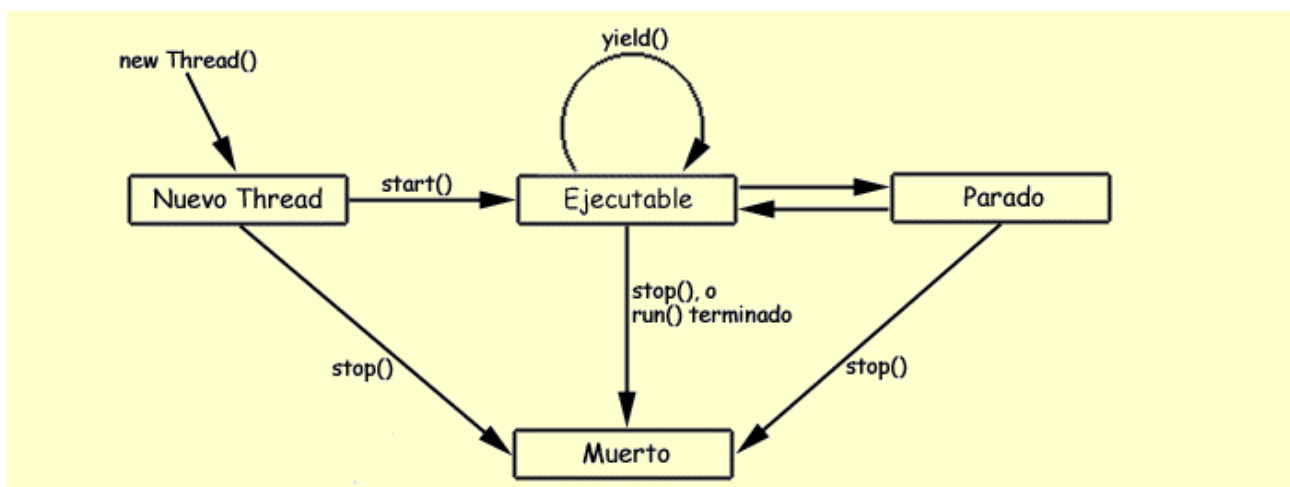
- Ejecución → cuando esta haciendo el uso del procesador
- Bloqueado, No puede ejecutarse hasta que se haya parado o bloqueado
- Listo → ha dejado disponible el procesador

### Diagrama de 6 estados



15.¿En qué estados puede estar un hilo?Realiza un diagrama

- Nuevo Thread → Desde este estado solamente puede arrancarse llamando al método start(), o detenerse definitivamente, llamando al método stop()
- Ejecutable → creara los recursos del sistema necesarios
- Parado → para esperar a que se cumpla una determinada condición
- Muerto → Un hilo de ejecución se puede morir de dos formas: por causas naturales o porque lo maten (constop()). Un hilo muere normalmente cuando concluye de forma habitual su método run()
- isAlive → El interfaz de programación de la clase Thread incluye el método isAlive(), que devuelve true si el hilo ha sido arrancado (con start()) y no ha sido detenido (con stop()). Por ello, si el método isAlive() devuelve false, sabemos que estamos ante un Nuevo Thread o ante un thread Muerto. Si devuelve true, se sabe que el hilo se encuentra en estado Ejecutable o Parado. No se puede diferenciar entre Nuevo Thread y Muerto, ni entre un hilo Ejecutable



16.¿Todos los hilos comparten los mismos recursos?

- Si como la memoria, los archivos abiertos, la situación de autenticación

17.¿Cómo podemos crear un hilo en Java? Pon un ejemplo

- Crear un hilo en java es una tarea muy sencilla. Basta heredar de la clase Thread y definir el método run(). Luego se instancia esta clase y se llama al método start() para que arranque el hilo.

```
public MiHilo extends Thread
{
    public void run(){

    }
};
MiHilo elHilo = new MiHilo();
elHilo.start();
System.out.println("Yo sigo a lo mio");
```

18.¿Qué método debemos implementar para definir lo que tiene que ejecutar el hilo?

- Se lo decimos en el método `run()` y metemos el código que queremos que haga el hilo.

19.¿Qué ocurre cuando ejecuto el método `start()`?

- Es cuando el hilo se ejecuta. también hereda de `Thread`

20.¿Para qué sirve el método “`SetPriority()`”?

- Cambia la prioridad del hilo, para ejecutar el hilo que quieras .

21.¿Qué otros métodos existen para la clase “`Thread`” de Java? Realiza una tabla donde pongas el método y su función.

<code>start()</code>	usado para iniciar el cuerpo de la thread definido por el método <code>run()</code> .
<code>sleep()</code>	pone a dormir una thread por un tiempo mínimo especificado.
<code>join()</code>	usado para esperar por el término de la thread sobre la cual el método es invocado, por ejemplo por término de método <code>run()</code>
<code>yield()</code>	Mueve a la thread desde el estado de corriendo al final de la cola de procesos en espera por la CPU
<code>stop()</code>	el cual detiene la ejecución de la thread no importando consideración alguna
<code>suspend()</code>	el cual para temporalmente la ejecución de una thread.
<code>resume()</code>	reactiva una thread suspendida

22.¿Qué es la interfaz “`Runnable`”?¿Cuándo se usa?

- Solamente declara una función miembro denominada *run*, que han de definir las clases que implementen esta interface .