
INSTITUTO TECNOLÓCIO SUPERIOR DE JEREZ


INGENIERÍA EN SISTEMAS COMPUTACIONALES

TÓPICOS AVANZADOS DE PROGRAMACIÓN

4to SEMESTRE

I.S.C. SALVADOR ACEVEDO SANDOVAL

# “MAPA CONCEPTUAL: CICLO DE VIDA DE HILOS”

ALBAR DE LA TORRE GARCÍA

No. Control: 16070122

Correo: albar00@hotmail.com

JEREZ ZACATECAS

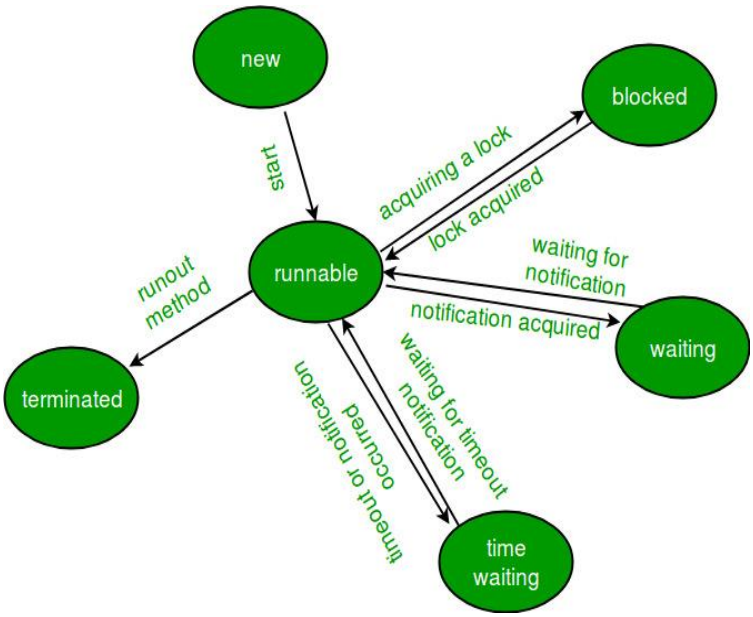
23 DE MARZO DEL 2018

# Ciclo De Vida Y Estados De Un Hilo En Java

Existe un hilo en Java en cualquier punto del tiempo en cualquiera de los siguientes estados. Un hilo se encuentra solo en uno de los estados mostrados en cualquier instante:

1. Nuevo
2. Runnable
3. Obstruido
4. Esperando
5. Espera temporizada
6. Terminado

El diagrama que se muestra a continuación representa varios estados de un hilo en cualquier instante de tiempo.




**Ciclo de vida de un hilo**

1. **Nuevo hilo:** cuando se crea un nuevo hilo, está en el nuevo estado. El subproceso aún no se ha iniciado cuando el subproceso está en este estado. Cuando un hilo se encuentra en el nuevo estado, su código aún no se ha ejecutado y no ha comenzado a ejecutarse.
2. **Estado ejecutable:** un hilo que está listo para ejecutarse se mueve al estado ejecutable. En este estado, un hilo podría estar ejecutándose o podría estar listo para ejecutarse en cualquier momento. Es responsabilidad del programador de hilos dar el hilo y el tiempo de ejecución. Un programa de subprocesos múltiples asigna una cantidad de tiempo fija a cada subproceso individual. Todos y cada uno de los subprocesos se ejecutan durante un breve período de tiempo y luego hacen una pausa y renuncian a la CPU a otro subproceso, para que otros subprocesos tengan la

oportunidad de ejecutarse. Cuando esto sucede, todos los subprocesos que están listos para ejecutarse, a la espera de que la CPU y el subproceso en ejecución se encuentren en estado ejecutable.

3. **Estado bloqueado / en espera:** cuando un hilo está temporalmente inactivo, está en uno de los siguientes estados:
    - Obstruido
    - Esperando

    Por ejemplo, cuando un hilo está esperando a que se complete la E / S, se encuentra en estado bloqueado. Es responsabilidad del programador de hilos reactivar y programar un hilo bloqueado / en espera. Un hilo en este estado no puede continuar su ejecución más hasta que se mueva a estado ejecutable. Cualquier hilo en estos estados no consume ningún ciclo de CPU.

    Un hilo está en el estado bloqueado cuando intenta acceder a una sección protegida de código que actualmente está bloqueada por algún otro hilo. Cuando la sección protegida está desbloqueada, el programa selecciona uno de los hilos bloqueados para esa sección y lo mueve al estado ejecutable. Mientras que, un hilo está en estado de espera cuando espera otro hilo en una condición. Cuando se cumple esta condición, se notifica al planificador y el hilo de espera se mueve a estado ejecutable.

    Si un hilo actualmente en ejecución se mueve al estado bloqueado / en espera, otro hilo en el estado ejecutable es programado por el programador de hilos para ejecutarse. Es responsabilidad del planificador de hilos determinar qué hilo ejecutar.

4. **Tiempo de espera:** un hilo se encuentra en el estado de espera temporizada cuando llama a un método con un parámetro de tiempo de espera. Un hilo se encuentra en este estado hasta que se complete el tiempo de espera o hasta que se reciba una notificación. Por ejemplo, cuando un hilo llama a suspensión o espera condicional, se mueve al estado de espera temporizada.
5. **Estado cancelado :** un hilo termina por cualquiera de los siguientes motivos:
    - Porque existe normalmente Esto sucede cuando el código del hilo ha sido ejecutado por completo por el programa.
    - Porque se produjo algún evento erróneo inusual, como un error de segmentación o una excepción no controlada.

    Un hilo que se encuentra en estado terminado ya no consume ningún ciclo de CPU.

## Punto Muerto En Java Multihilo

La palabra clave synchronized se usa para hacer que la clase o método sea seguro para subprocesos, lo que significa que solo un subproceso puede tener un método sincronizado y usarlo, otros subprocesos tienen que esperar hasta que el bloqueo se libere y cualquiera de ellos adquiera ese bloqueo.

Es importante usarlo si nuestro programa se ejecuta en un entorno de subprocesos múltiples donde dos o más subprocesos se ejecutan simultáneamente. Pero a veces también causa un problema que se llama punto muerto. A continuación se muestra un ejemplo simple de condición de punto muerto.

**Figure -** 1

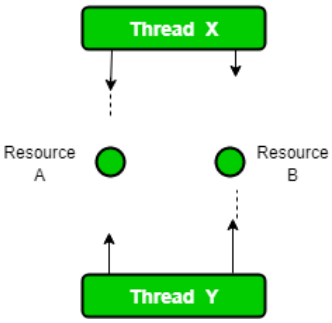
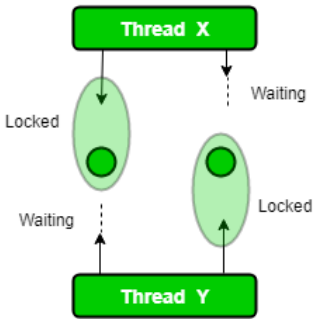



**Figure -** 2




## Evitar La Condición De Bloqueo Muerto

Podemos evitar la condición de bloqueo muerto al conocer sus posibilidades. Es un proceso muy complejo y no es fácil de atrapar. Pero aun así, si lo intentamos, podemos evitar esto. Hay algunos métodos por los cuales podemos evitar esta condición. No podemos eliminar completamente su posibilidad, pero podemos reducirla.

Evite las cerraduras anidadas: esta es la razón principal para el bloqueo muerto. Dead Lock ocurre principalmente cuando damos bloqueos a múltiples hilos. Evite dar bloqueo a múltiples hilos si ya se lo hemos dado a uno.

Evite bloqueos innecesarios: deberíamos haber bloqueado solo los miembros que se requieren. Tener el bloqueo innecesariamente puede conducir a un bloqueo muerto.

Uso de la unión de subprocesos: la condición de bloqueo inactivo aparece cuando un subproceso está esperando que otro finalice. Si se produce esta condición, podemos usar Thread.join con el tiempo máximo que cree que tomará la ejecución.

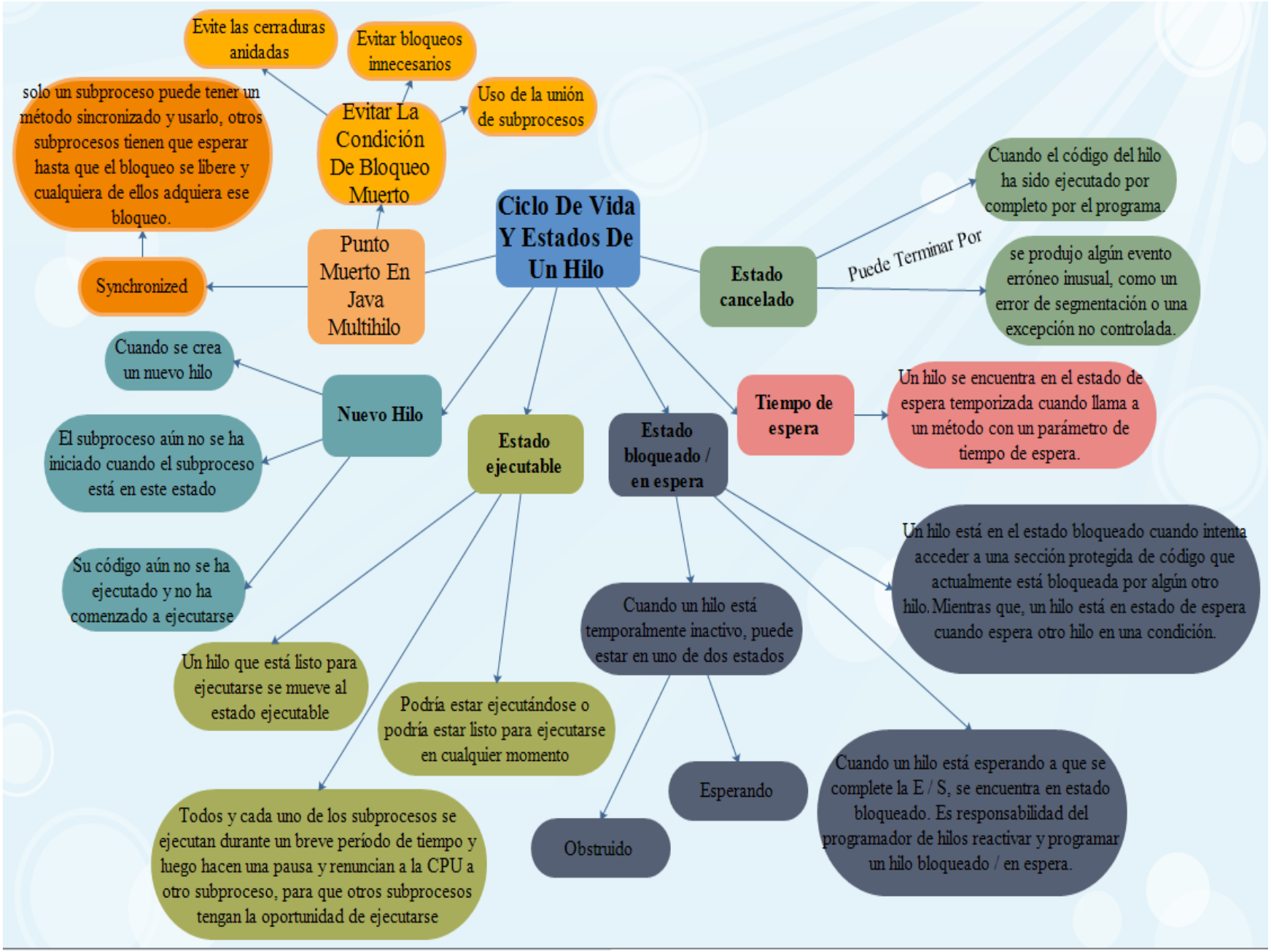
Puntos Importantes:

Si los hilos están esperando el uno al otro para terminar, entonces la condición se conoce como Interbloqueo.

La condición de punto muerto es una condición compleja que ocurre solo en caso de múltiples hilos.

La condición de interbloqueo puede romper nuestro código en tiempo de ejecución y puede destruir la lógica comercial.

Deberíamos evitar esta condición tanto como podamos.


Ciclo De Vida Y Estados De Un Hilo
Nuevo Hilo
Cuando se crea un nuevo hilo
El subproceso aún no se ha iniciado cuando el subproceso está en este estado
Su código aún no se ha ejecutado y no ha comenzado a ejecutarse
Estado ejecutable
Un hilo que está listo para ejecutarse se mueve al estado ejecutable
Podría estar ejecutándose o podría estar listo para ejecutarse en cualquier momento
Todos y cada uno de los subprocesos se ejecutan durante un breve período de tiempo y luego hacen una pausa y renuncian a la CPU a otro subproceso, para que otros subprocesos tengan la oportunidad de ejecutarse
Estado bloqueado / en espera
Un hilo está en el estado bloqueado cuando intenta acceder a una sección protegida de código que actualmente está bloqueada por algún otro hilo. Mientras que, un hilo está en estado de espera cuando espera otro hilo en una condición.
Cuando un hilo está temporalmente inactivo, puede estar en uno de dos estados
Esperando
Cuando un hilo está esperando a que se complete la E / S, se encuentra en estado bloqueado. Es responsabilidad del programador de hilos reactivar y programar un hilo bloqueado / en espera.
Obstruido
Tiempo de espera
Un hilo se encuentra en el estado de espera temporizada cuando llama a un método con un parámetro de tiempo de espera.
Estado cancelado
Puede Terminar Por
Cuando el código del hilo ha sido ejecutado por completo por el programa.
se produjo algún evento erróneo inusual, como un error de segmentación o una excepción no controlada.
Punto Muerto En Java Multihilo
Synchronized
solo un subproceso puede tener un método sincronizado y usarlo, otros subprocesos tienen que esperar hasta que el bloqueo se libere y cualquiera de ellos adquiera ese bloqueo.
Evitar La Condición De Bloqueo Muerto
Evite las cerraduras anidadas
Evitar bloqueos innecesarios
Uso de la unión de subprocesos