



ESCUELA DE
INGENIERÍA EN CIENCIAS Y SISTEMAS
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



Día, Fecha:

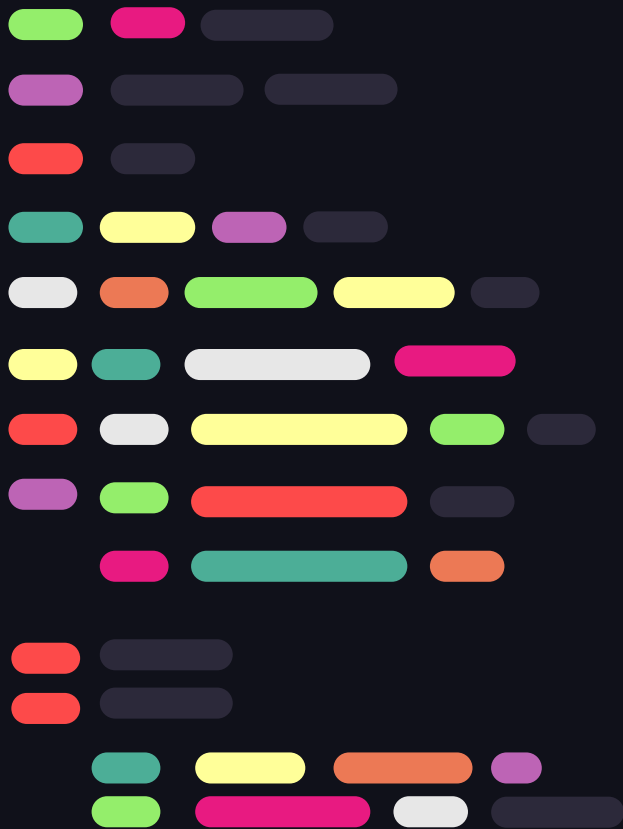
Martes, 11/06/2024

Hora de inicio:

13:00

Introducción a la Programación y Computación 1 [A]

Josué Rodolfo Morales Castillo



{ ..



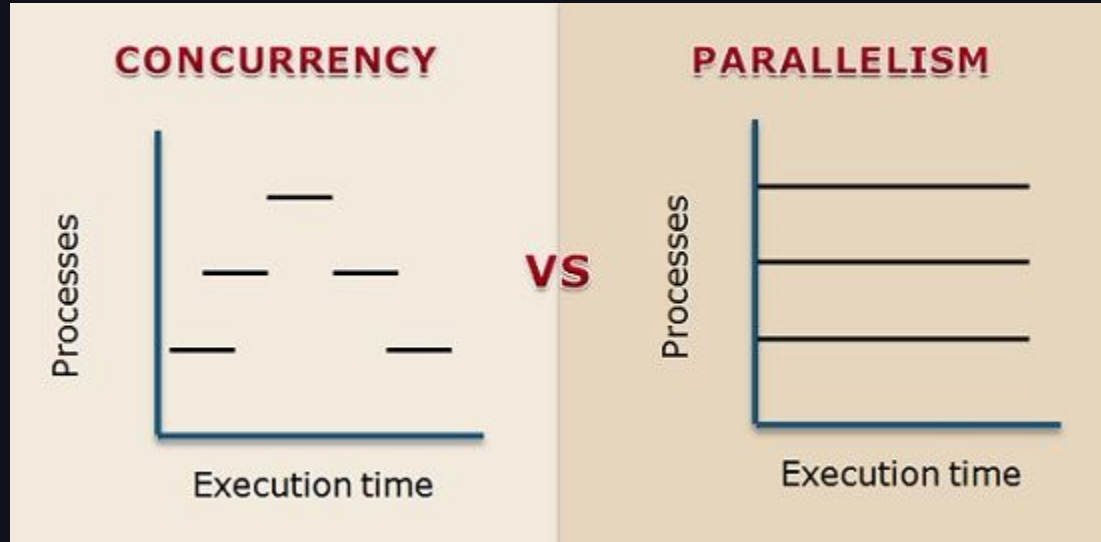
Clase 7- Agenda

- Foro No. 2
- Preguntas práctica 1
- Concurrencia y paralelismo

} ..



Concurrencia y paralelismo



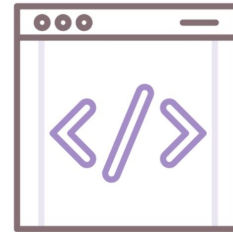


<https://www.youtube.com/watch?v=16v1-c1237U>

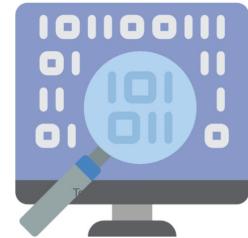


Procesos

Es una instancia en ejecución de un programa de computadora. Puede incluir el código ejecutable, datos, variables de entorno, asignaciones de memoria, recursos del sistema operativo y otros atributos que describen el estado de ejecución del programa.



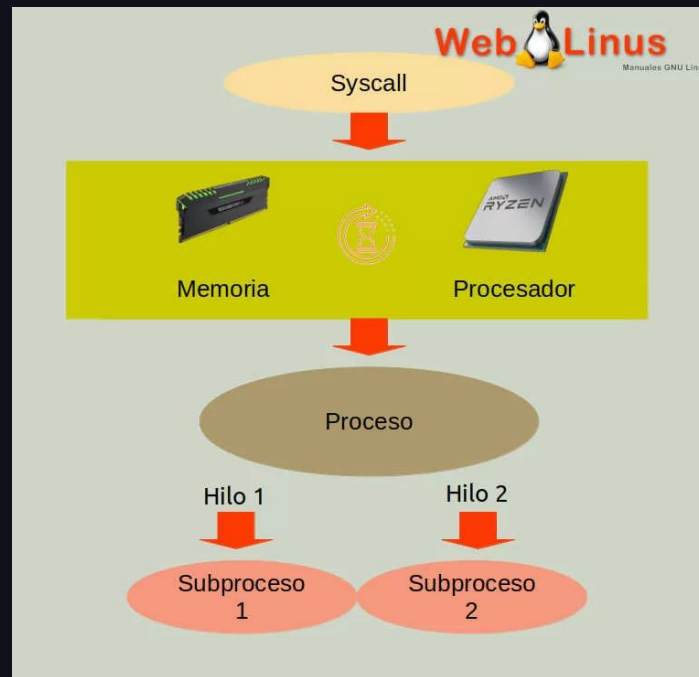
PROGRAMA



PROCESO

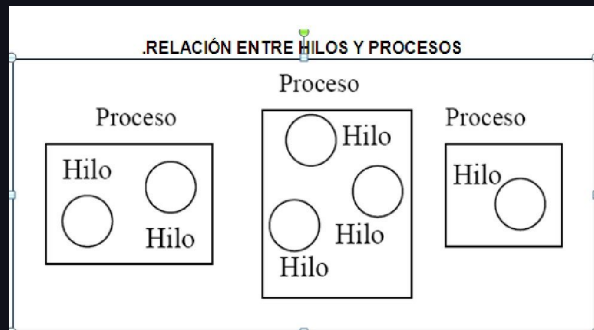
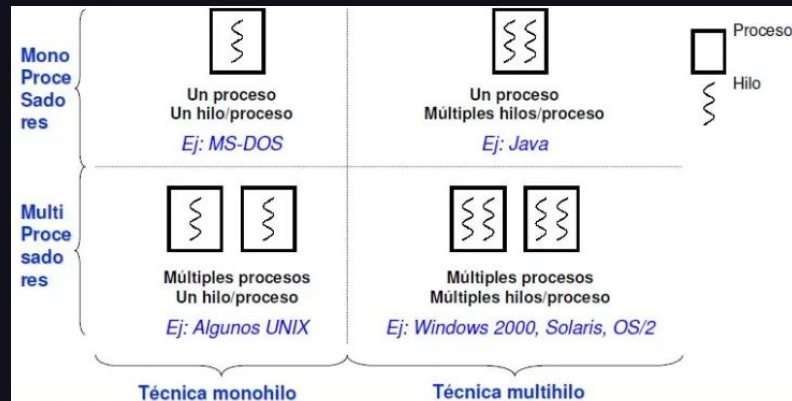
Subprocesos

Es una secuencia de instrucciones que puede ser ejecutada por un procesador de manera independiente de otros subprocesos. Los subprocesos comparten recursos con otros subprocesos del mismo proceso, lo que permite la ejecución de múltiples tareas de manera concurrente.



Hilos

Un hilo es como un trabajador en una fábrica de computadoras. Cada trabajador (hilo) tiene su propio conjunto de instrucciones que debe seguir (program counter), sus propias herramientas y espacio para guardar cosas temporalmente (registros y pila). El sistema operativo organiza y gestiona a estos trabajadores usando una estructura llamada TCB (thread control block).



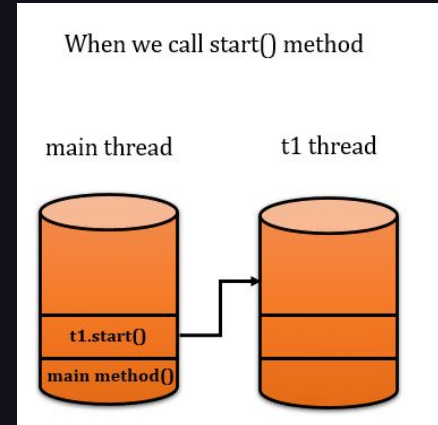
Cuando un proceso tiene varios hilos, es como si estuvieras ensamblando múltiples computadoras al mismo tiempo, lo que significa que puedes realizar más de una tarea al mismo tiempo, especialmente si tienes más de un "CPU" (unidad de procesamiento central). Esto hace que el trabajo sea más eficiente y rápido.

Método Start

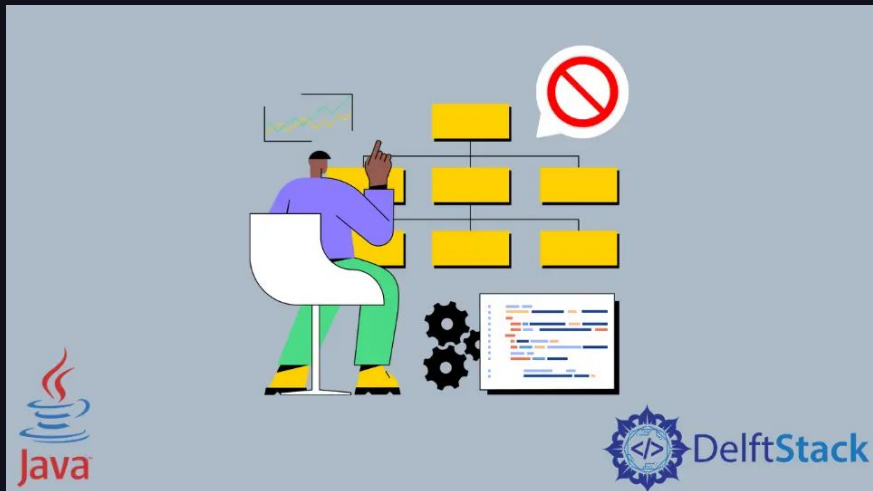
El método "start" de la clase Thread se utiliza para iniciar la ejecución de un hilo en Java. Al invocar este método, se inicia un nuevo hilo de ejecución que corre concurrentemente con otros hilos en el programa. Es importante señalar que el método "start" no ejecuta directamente el código definido para el hilo; en cambio, prepara y programa el sistema operativo para que ejecute el hilo de manera independiente.

La utilización de "start" es crucial para aprovechar la concurrencia y la ejecución simultánea de tareas en aplicaciones Java. La implementación adecuada de este método es esencial para evitar posibles problemas de sincronización y asegurar un comportamiento predecible y eficiente en entornos con múltiples hilos.

La concurrencia en programación se refiere a la ejecución simultánea de múltiples tareas o procesos, permitiendo que estas operaciones se lleven a cabo al mismo tiempo.



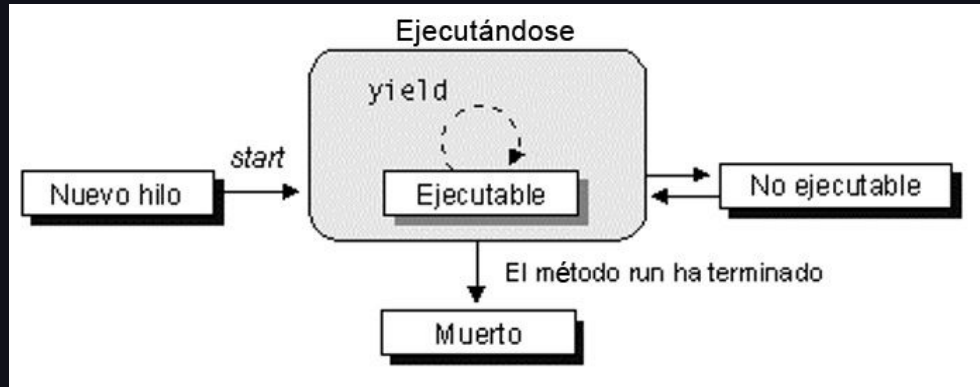
Detener hilo



En Java, el método `stop()` que estaba originalmente diseñado para detener hilos está marcado como obsoleto y no se recomienda su uso debido a problemas de seguridad y estabilidad. En su lugar, se aconseja utilizar una variable de control (bandera) que el hilo consulte periódicamente para determinar si debe finalizar su ejecución. Al cambiar el valor de esta variable, puedes indicar al hilo que debe detenerse de manera segura.

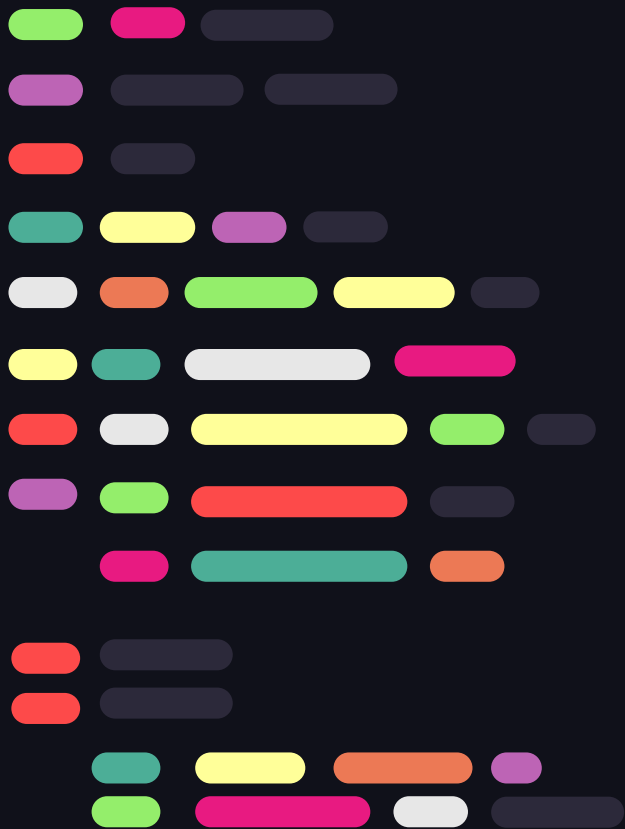
Espera de finalización de un hilo

Para esperar la finalización de un hilo en Java, se puede utilizar el método `join()`. Este método se llama en el hilo principal y bloquea su ejecución hasta que el hilo al que se llama haya finalizado. Así, `join()` asegura que el hilo principal no continúe su ejecución hasta que el hilo que se está esperando haya completado su tarea.



¿Dudas?





{ ..



Ejemplo

} ..