Universidad ICESI

Algoritmos y Programación II

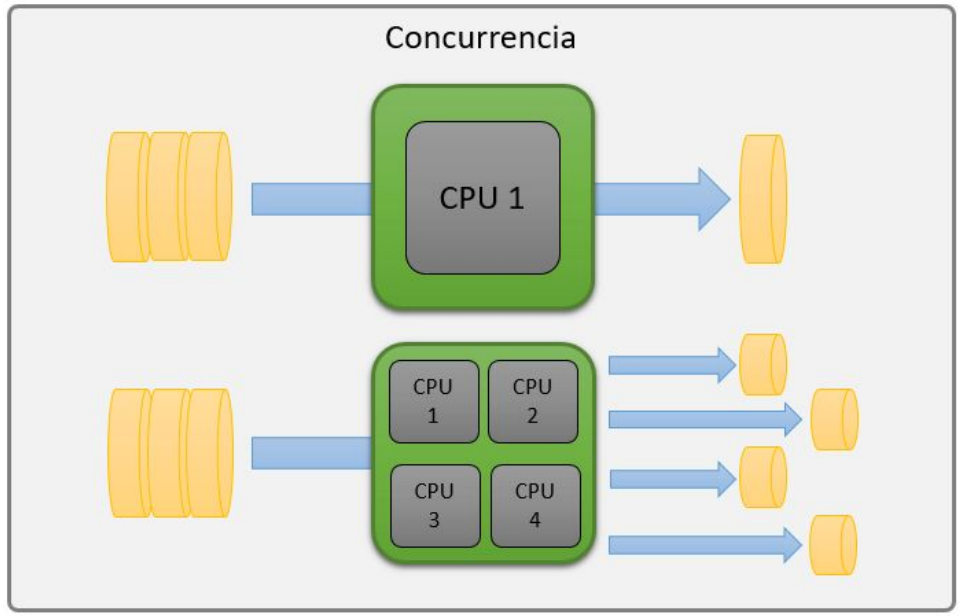
# Concurrencia, hilos y animación

Esta presentación proporciona una exploración en el concepto de hilos, su implementación con java. Cubre ejemplos.

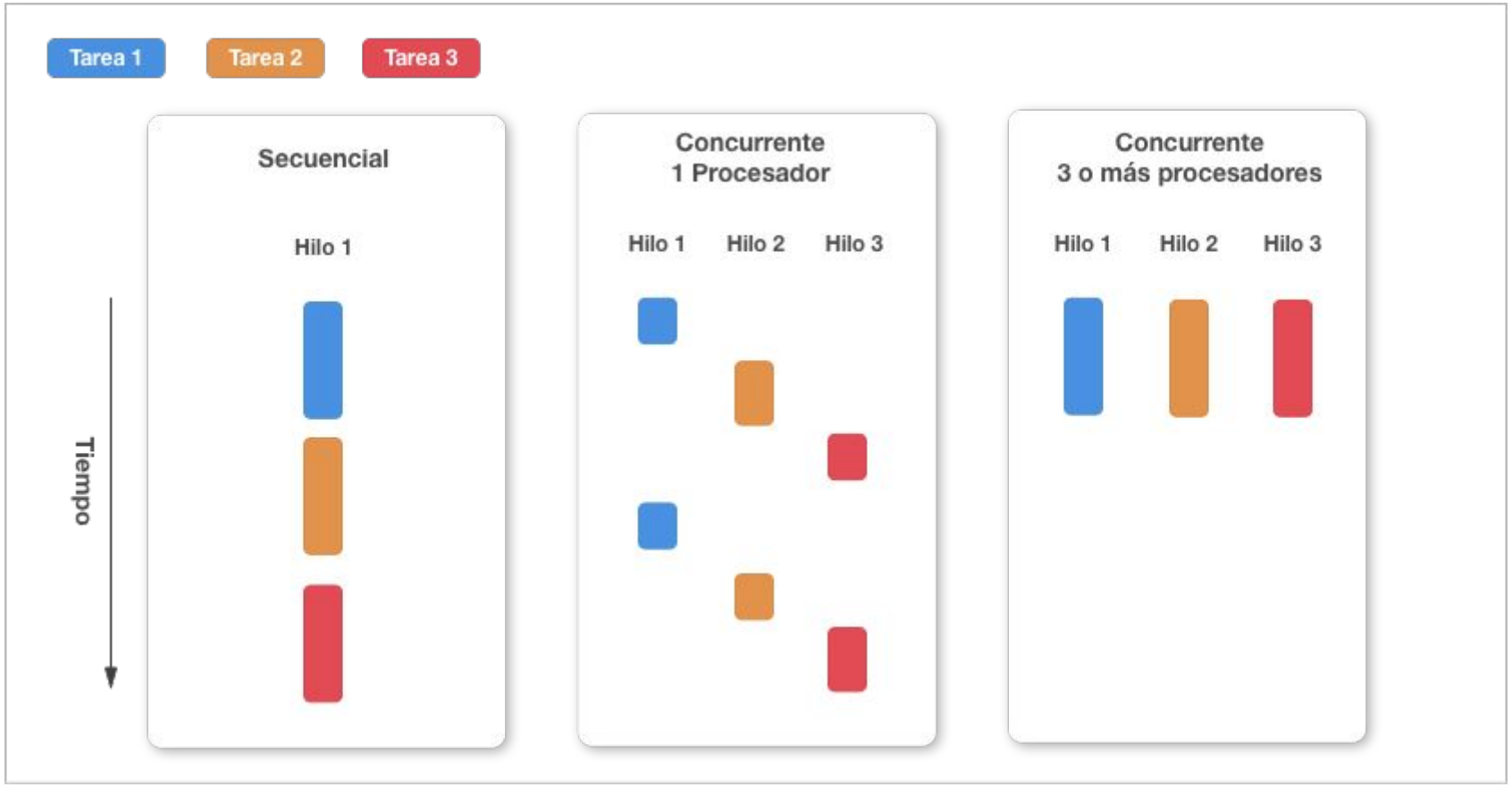
# ¿Qué es la concurrencia?

La concurrencia es la capacidad de un sistema para ejecutar varias tareas de forma simultánea o dar esa apariencia, permitiendo que diferentes partes de un programa progresen de manera independiente.

No implica necesariamente ejecución en paralelo (ejecución en múltiples procesadores), ya que puede ocurrir en sistemas de un solo núcleo mediante la intercalación de tareas que comparten el procesador.



# Tareas Secuenciales vs Tareas concurrentes vs Tareas Paralelas

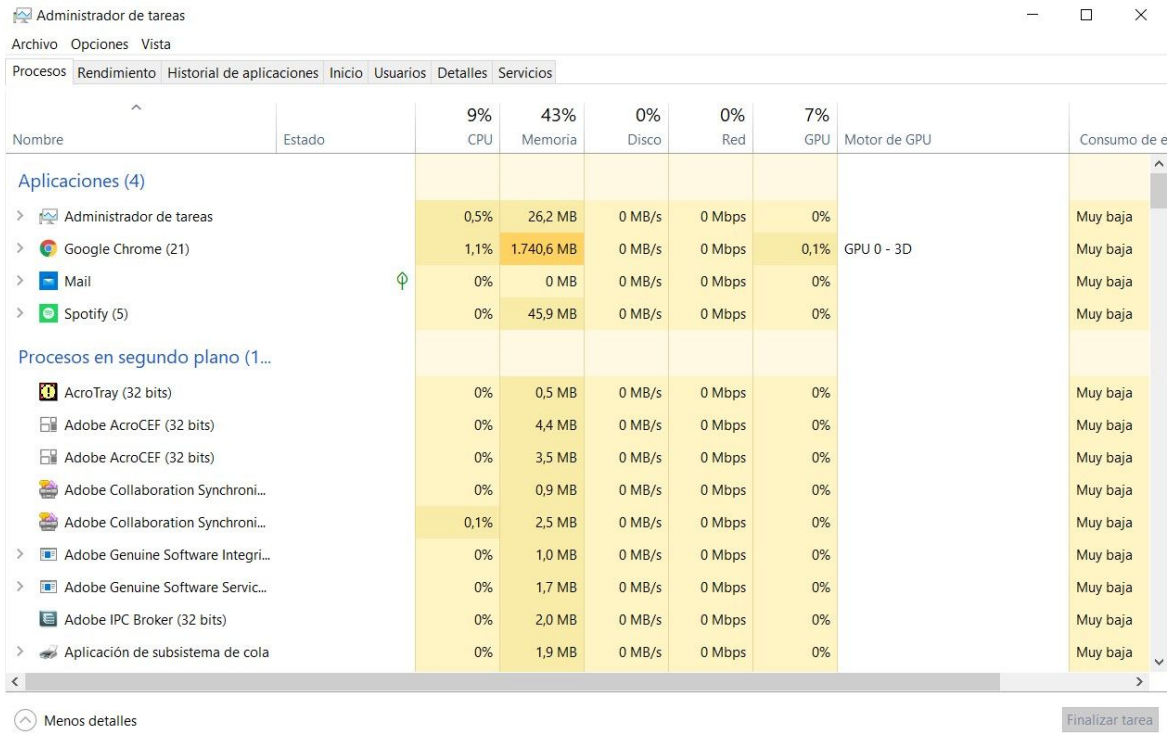


# Programa/Aplicación

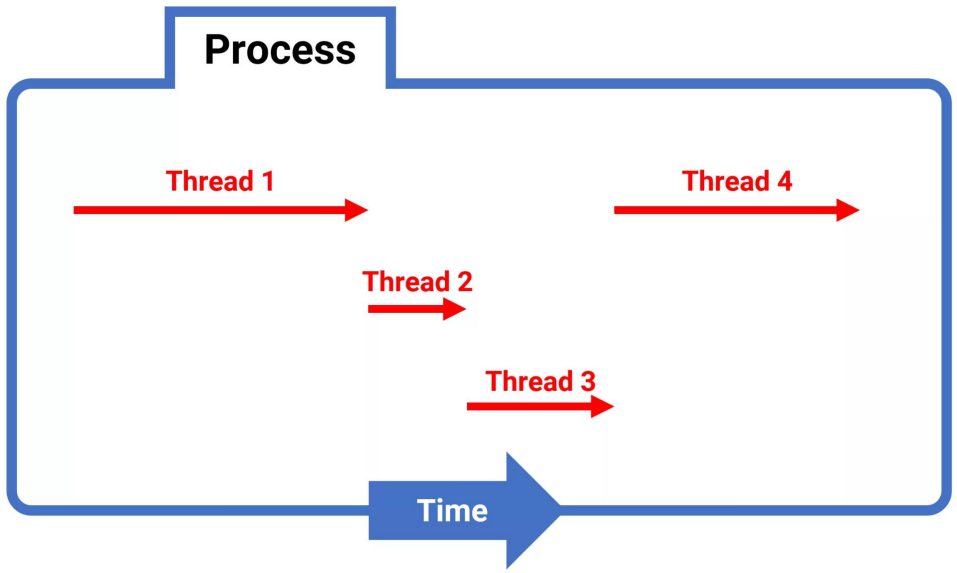
Conjunto de instrucciones escritas en un lenguaje de programación específico, que define la secuencia de acciones que debe realizar una computadora para llevar a cabo una tarea o resolver un problema. Ejemplos de programas incluyen navegadores web, procesadores de texto y juegos.

# Proceso

Es una instancia en ejecución de un programa. Cada proceso es una entidad activa que posee recursos asignados, como memoria, espacio en disco y acceso a dispositivos de entrada/salida. Los procesos son creados y gestionados por el sistema operativo, que se encarga de planificar su ejecución y distribuir los recursos disponibles entre ellos. en un SO, se pueden ejecutar múltiples procesos simultáneamente.



# Hilo




Un hilo (también conocido como subproceso) es la unidad más pequeña de ejecución en un proceso. Los hilos también comparten recursos y espacio de memoria con otros hilos dentro del mismo proceso. Los hilos permiten que un proceso realice múltiples tareas de manera concurrente, mejorando la eficiencia y capacidad de respuesta del programa.


# CONCURRENCIA EN JAVA

# ¿Cómo se aplica la Concurrencia en Java?

En Java, la concurrencia se modela por la clase Thread o la interfaz Runnable y permite:

- Ejecutar los programas en múltiples **hilos (threads)** simultáneamente.

- Mejorar el **rendimiento** y la **responsividad** de programas, especialmente en interfaces gráficas, redes, servidores, juegos, etc.

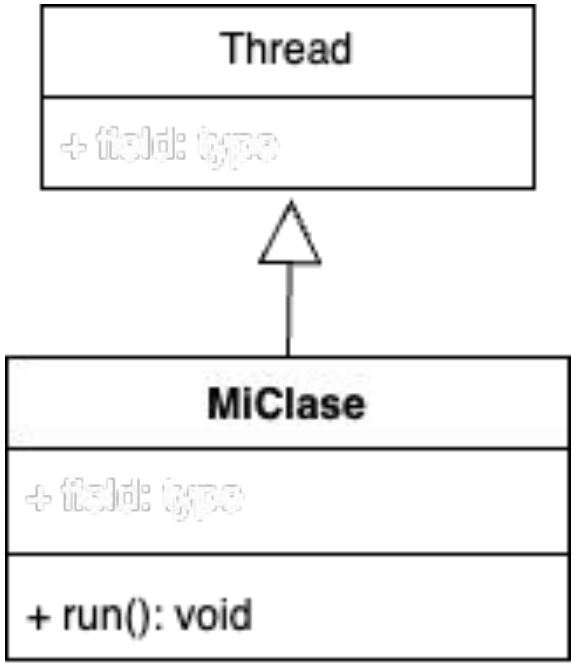
# Clase Thread – Interfaz Runnable

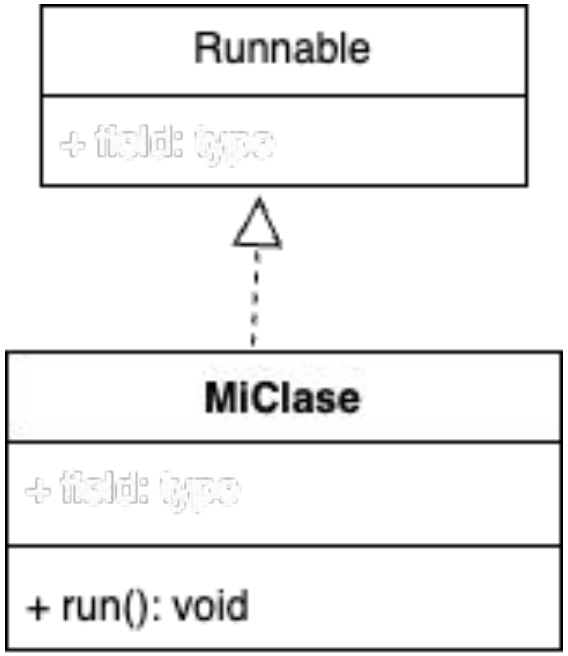
Las clases para manejar hilos en Java están en el **paquete java.lang**, que ya está **importado por defecto**.

Principales clases e interfaces:

- Thread (clase): se hereda, sobrescribir el método run() para definir las tareas que realizará el hilo.
- Runnable (interfaz): se implementa, en una clase y definir las tareas del hilo en el método run().

# Clase Thread – Interfaz Thread



# Ciclo de vida de un hilo

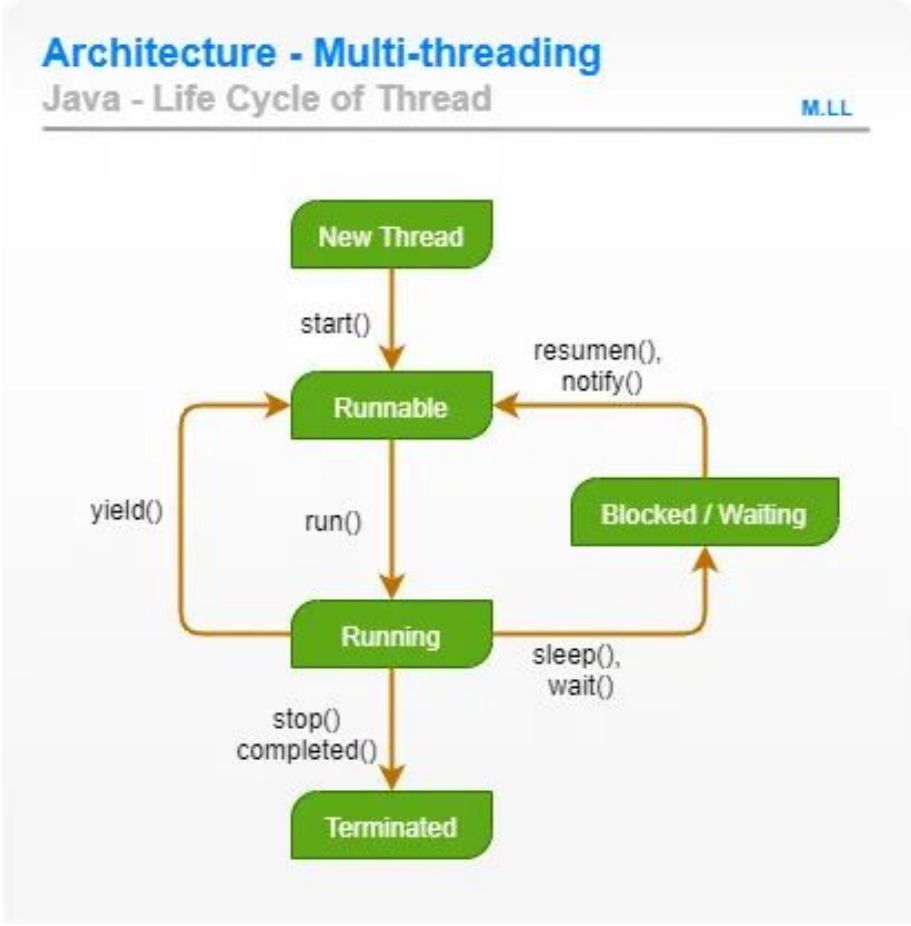
```
New Thread (crear un hilo)
    Thread t = new Thread();

Runnable (iniciar un hilo)
    t.start();

Running (Ejecución)
    implementar el método
    run()

Blocked / waiting
    t.sleep(1000);

Terminated / Dead
    Cuando el método run
    termina, el hilo entra en
    estado Terminated o dead
```



# Ejemplo

```
public class MiHilo extends Thread {
    @override
      public void run() {
           try{
                while(true){
                System.out.println("Hola desde el hilo: " +
Thread.currentThread().getName());
                Thread.sleep(1000);
                }
           }
           catch (InterruptedException e){
                System.out.println(e.getMessage());
           }
      }
}
```

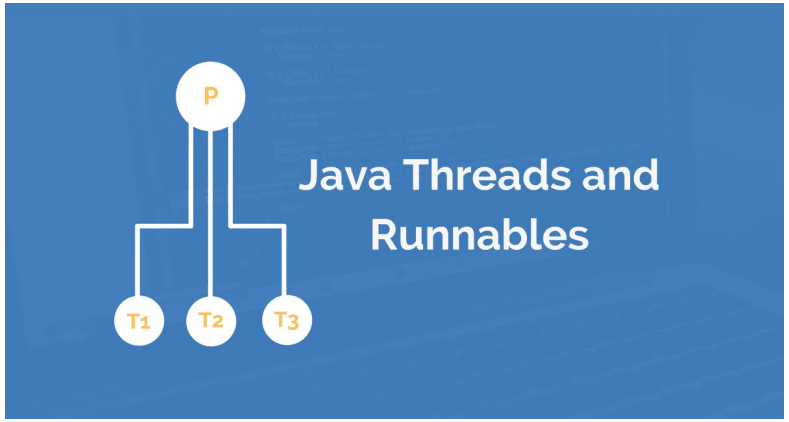
```
public class Ejemplo {
    public static void main(String[] args) {
        MiHilo hilo = new MiHilo();
        hilo.start(); // Llama internamente a run(), en un nuevo
hilo
    }
}
```

# Nota Importante sobre los métodos

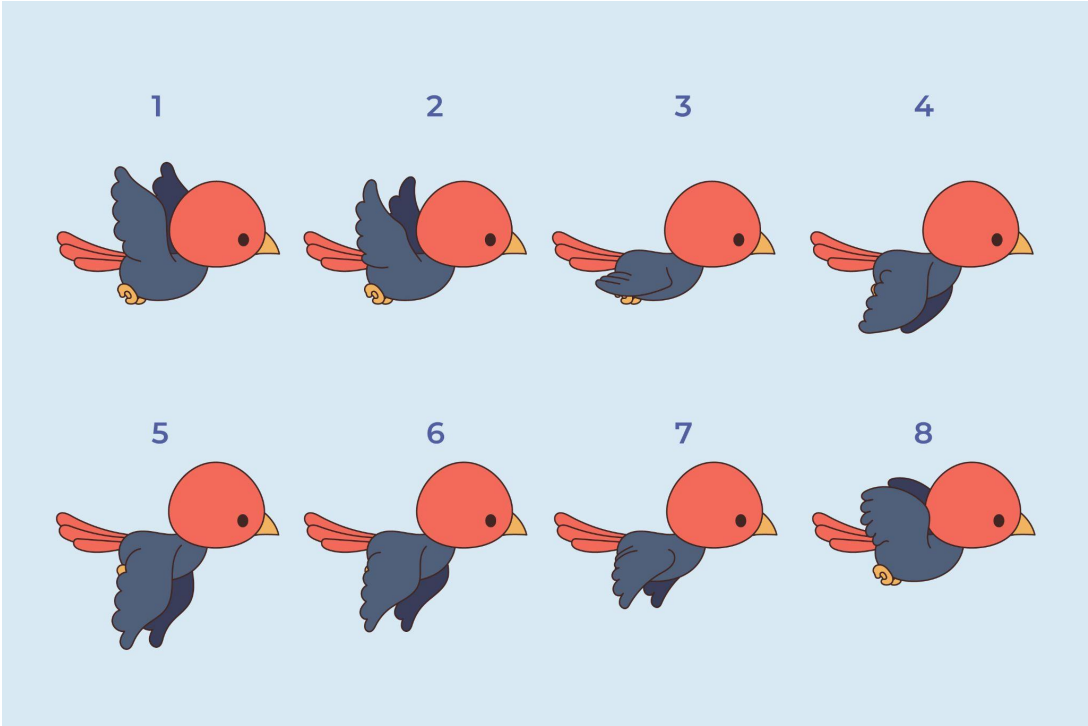
- **start()** se utiliza para **iniciar** un hilo, creando un nuevo objeto hilo y poniéndolo en ejecución dentro del planificador del sistema operativo.

- **run()** define el **código** que se va a ejecutar por parte del hilo, pero no debe usarse para iniciarlo.

Siempre se debe utilizar el método **start()** para iniciar un hilo en Java. Esto garantiza que el hilo se ejecute de forma correcta, asíncrona y controlada, aprovechando las ventajas de la concurrencia y evitando problemas de sincronización y rendimiento.



# Threads y Animación




**¿Cómo animar elementos gráficos en JavaFX?**

1. Gráficos 2D: Pintar los elementos gráficos sobre la gui (imágen de pájaro).
2. Lógica de movimiento: Definir una lógica de movimiento para la entidad que representa el elemento gráfico (moverse en una dirección).
3. Ejecutar el movimiento a través de Threads: Ejecutar el movimiento como un hilo del programa para que repetidamente realice el movimiento. Esto también incluye cambiar la imagen para formar la secuencia de movimiento (pájaro volando).

Universidad ICESI

# Gracias. 😄