
Creando una solución simple para la ejecución de flujos de trabajo - nanoFT

Profesores

Julio Ariel Hurtado Alegría

Requerimiento

Un flujo de trabajo es la operacionalización de una actividad: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas.

Se desea construir nanosoFT una versión simplificada de un flujo de trabajo. La idea es proveer una solución simple sobre dos bloques de construcción esenciales (Tareas y Actividades) para facilitar la creación y ejecución de flujos de trabajo compuestos. La solución debe considerar lo siguiente:

1. Una Tarea es la unidad más simple de trabajo
2. Una Actividad es la abstracción primaria de flujo de trabajo
3. Una actividad se puede crear y ejecutar de la siguiente forma:
 - a. Secuencial, una actividad que ejecuta actividades o tareas en el orden que se encuentren.
 - b. Paralelo, una actividad con varias tareas o actividades que se ejecutan al mismo tiempo.
 - c. Condicional, una actividad que dependiendo de alguna variable realiza una u otra tarea o actividad.
 - d. Repetitivo, una actividad que repite una tarea o actividad varias veces.

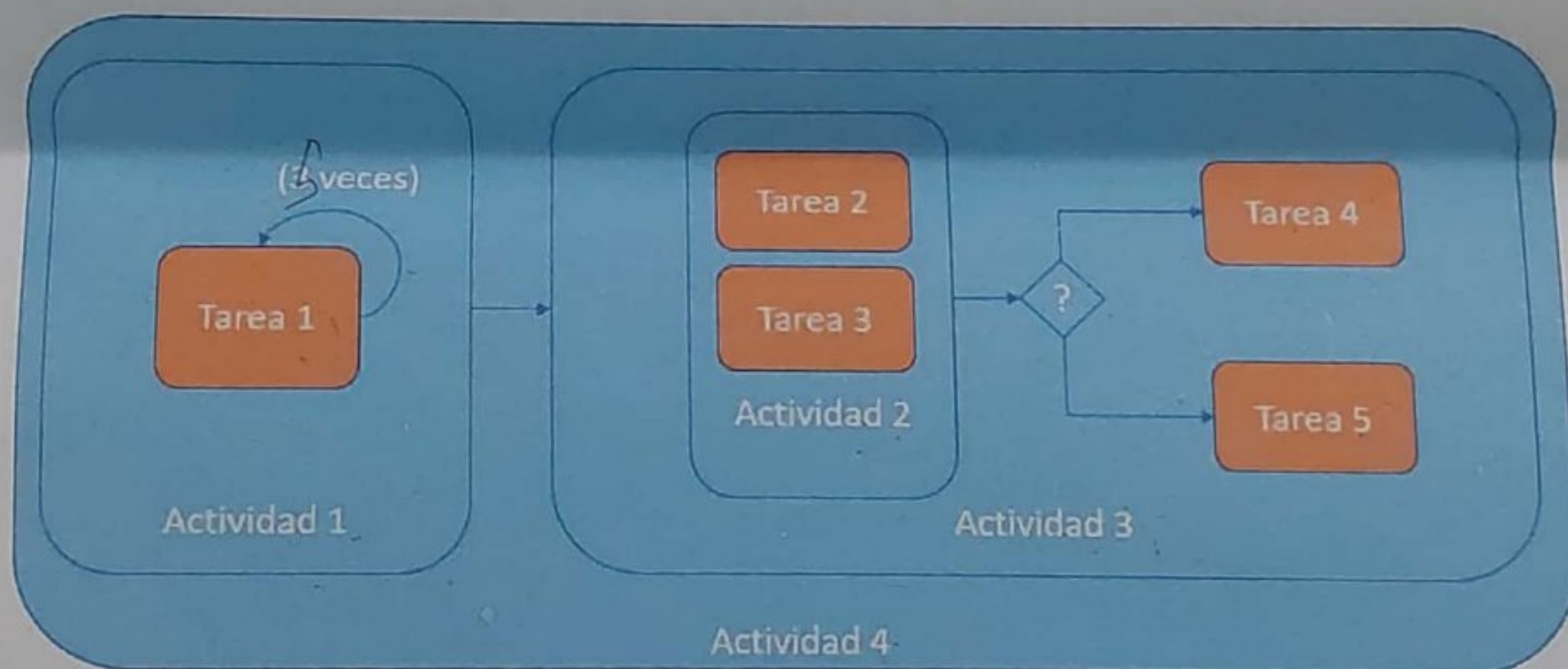
4. Ejecutar una Actividad a la cual se le debe siempre suministrar una variable que representa el Contexto sobre la cual debe ejecutar o sobre el cual se actualiza algún aspecto del contexto durante la ejecución de la actividad (en este puede fluir la información entre tareas además de su información retornada).

Ejemplo

Vamos a considerar una pequeña actividad que hace los siguiente:

1. Imprime "hola" cinco veces
2. Luego imprime "Principios" and "SOLID" en paralelo
3. Si "Principios" and "SOLID" han sido imprimidos exitosamente en la consola entonces imprimir "Bien", sino imprimir "Algo ha fallado"

Ver la siguiente gráfica



En este caso:

4. Actividad 1 es una Actividad Repetitiva de la Tarea 1 la cual está imprimiendo "hola" cinco veces.
5. Actividad 2 es una Actividad Paralela de las Tarea 2 y Tarea 3 las cuales imprimen respectivamente y en paralelo las palabras "Principios" and "SOLID".
6. Actividad 3 es una Actividad condicional que primero ejecuta la Actividad 2 (una Actividad también es una Tarea), luego cuando se completa exitosamente la Actividad 2, ejecuta la Tarea 4 (imprimir bien), en caso contrario ejecuta la Tarea 5(imprimir Algo ha fallado).
7. Actividad 4 es una Actividad Secuencial, esta ejecuta la Actividad 1 y luego la Actividad 3 en secuencia.

Productos de Trabajo Esperados

- Diseño de nanoFT: diagrama de clases y diagrama de secuencia de una actividad secuencial.
- Código Java de la solución, ejecutando con el funcionamiento del flujo de trabajo ejemplo.
- Mostrar cómo se están cumpliendo o violando los principios SOLID en su solución.

Equipos

Equipos de cuatro integrantes

Tiempo de la Actividad

1 hora 30 minutos

main() {

Tarea tarea1 = new TareaImpr("Hola");

Actividad act1 = new ActividadRepetitiva(tarea1, 5);

Tarea tarea2 = new TareaImpr("Principios");

Tarea tarea3 = new TareaImpr("Solid");

Actividad act2 = new ActividadParalela(tarea2, tarea3);

Tarea tarea4 = new TareaImpr("Bien");

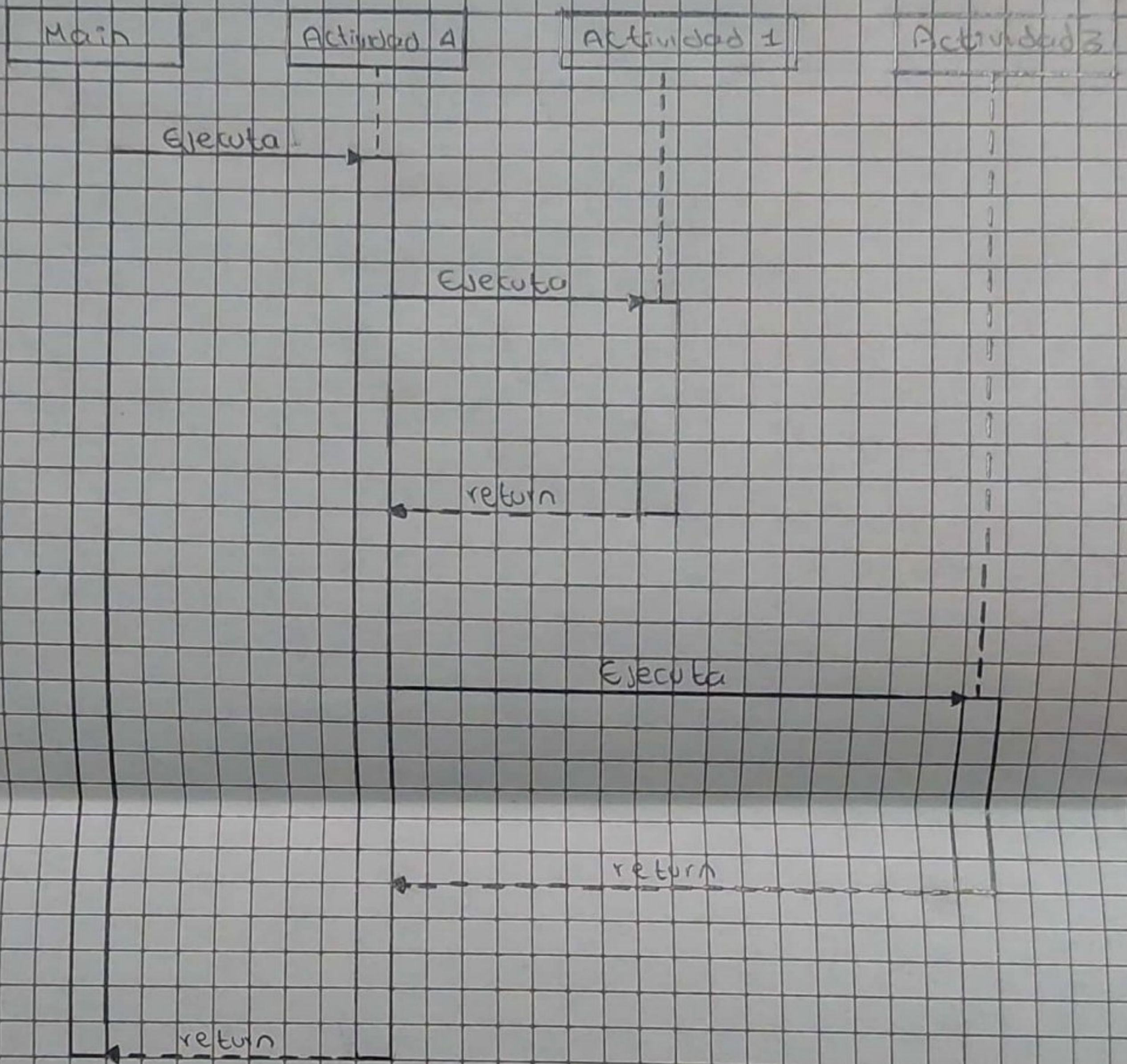
Tarea tarea5 = new TareaImpr("algo ha fallado");

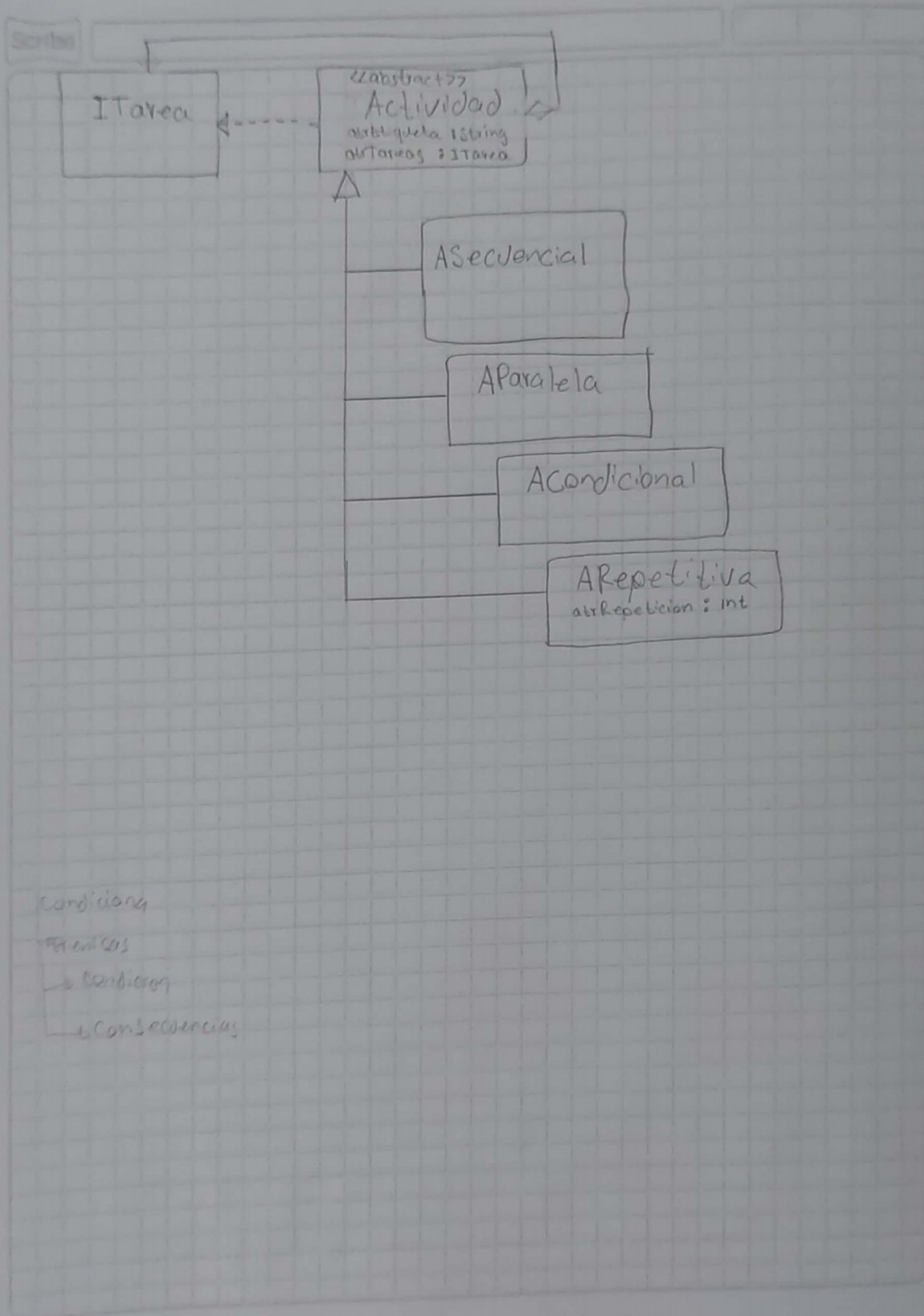
Actividad act3 = new ActividadCondiciona1(act2, Cond, tarea4, tarea5);

Actividad act4 = new ActividadSecuencial(Actividad1, Actividad3)

act4.ejecutar();

}





condiciona
repetitiva
→ Condiciona
→ Consecuencia

