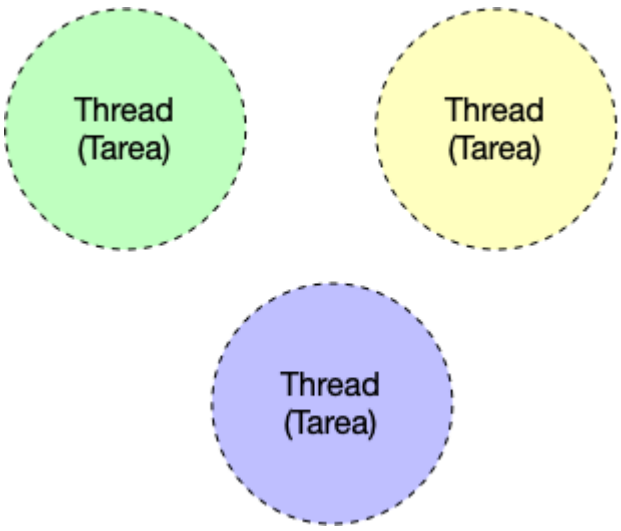
Tabla de Contenidos

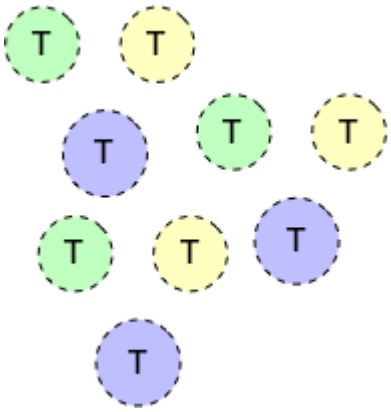
¿Que son los Java Virtual Threads? . Cuando programamos en Java y queremos hacer varias cosas al mismo tiempo, usamos hilos. (Threads) Un hilo es una abstracción sobre el concepto de tarea y se pueden ejecutar en paralelo. Cada Thread de Java es un hilo del sistema operativo.



El problema es que los hilos normales usan bastante memoria y el sistema operativo los tiene que controlar. Esto puede hacer que la aplicación sea lenta si usamos muchos de forma simultanea.

Con Virtual Threads (hilos virtuales), Java nos permite crear miles o millones de hilos de forma ligera y rápida. Así, podemos hacer muchas tareas a la vez sin que el sistema se

sobrecargue ya que como su nombre indica son virtuales no son hilos de sistema operativo.



¿Qué ventajas tienen los Virtual Threads?

Los Virtual Threads traen varias ventajas importantes:

1. Muchos hilos, sin problemas de memoria

Podemos tener millones de tareas activas sin preocuparnos porque la memoria se llene.

2. Código más fácil de entender

Seguimos escribiendo código de forma normal y sencilla, sin tener que usar técnicas complicadas como la programación reactiva que solo se necesita en algunos casos concretos que impliquen sincronizaciones y gestión avanzada

3. Mejor uso de recursos

Las aplicaciones que usan Virtual Threads aprovechan mejor la memoria y el procesador ya que cada hilo no tiene porque ser un hilo de sistema operativo

---

¿Cómo se usan los Virtual Threads en Java?

Utilizar Virtual Threads es muy fácil en Java 21 o superior. Aquí tienes un ejemplo simple:

```
Runnable tarea = () -> {
    System.out.println("Hola desde un Virtual Thread: " +
Thread.currentThread());
};

Thread hiloVirtual = Thread.ofVirtual().start(tarea);
```

También podemos usar un Executor que se encarga de lanzar las tareas automáticamente:

```
try (var ejecutor = Executors.newVirtualThreadPerTaskExecutor()) {
    ejecutor.submit(() -> {
        System.out.println("Tarea en Virtual Thread");
    });
}
```

Con este enfoque, puedes lanzar muchas tareas de forma muy ligera.

¿Dejo de usar los hilos normales?

No siempre.

- Si tu tarea espera respuestas (como leer una base de datos o llamar a otro servidor), los Virtual Threads son ideales.
- Si tu tarea hace muchos cálculos pesados, los hilos normales todavía son una buena opción. Lo mejor es usar ambos según el tipo de tarea que tengas.

Resumen: ¿Por qué usar Java Virtual Threads?

- Son hilos ligeros que usan menos memoria.
- Permiten manejar miles o millones de tareas al mismo tiempo.
- Hacen que el código siga siendo simple.
- Mejoran el rendimiento de las aplicaciones modernas.
  Si quieres aplicaciones más rápidas y fáciles de programar, ¡empieza a usar Virtual Threads con Java 21!

- Arquitectura WebSocket y Threads
- ¿Cómo elegir un buen Framework?
- Java wait notify y threads
- Servlet 3.0 (II) Servlets Asincronos
- ¿Cómo funciona JPA First Level Cache ?