

# Programación Concurrente en Java

## Programación Concurrente – Tema 5

Miguel Ángel Rodríguez García

Carlos Grima

Lucía Serrano Luján

# Introducción

Programación Concurrente en Java -  
Tema 5.1

## • Modelo de concurrencia

- Java ofrece un modelo de concurrencia de **memoria compartida** integrado en el lenguaje Java y en la librería estándar
- El soporte de concurrencia ha **evolucionado mucho en Java**
- En las últimas versiones existen **herramientas de alto nivel** en la librería estándar que permiten al desarrollador **abstraerse de detalles de bajo nivel**

- La concurrencia también está integrada en el propio **lenguaje de programación**
  - Está definido cómo se comparte la memoria entre diferentes hilos en el Modelo de Memoria de Java [1] (**Java Memory Model**)
  - Existen palabras reservadas en el propio lenguaje relacionadas con la concurrencia, por ejemplo para delimitar zonas de exclusión mutua se usa **synchronized**

[1] <http://www.cs.umd.edu/~pugh/java/memoryModel/jsr-133-faq.html>

- También tiene soporte en la **librería estándar**
  - Como Java es un **lenguaje orientado a objetos**, los hilos se representan como objetos de la clase **java.lang.Thread**
  - Esta clase tiene **métodos para controlar el ciclo de vida del hilo** (configurar su nombre, prioridad, iniciar, esperar a que finalice, etc...)
  - En Java los hilos se pueden **crear y destruir en cualquier momento** de la ejecución del programa, **no** tienen que seguir la **rígida estructura** de los programas de **SimpleConcurrent**

- En las sucesivas versiones de la plataforma Java el soporte de **programación concurrente** ha avanzado considerablemente
- Es muy importante tener en cuenta la **evolución de las herramientas** de concurrencia en Java cuando se revise **documentación** (para evitar que esté obsoleta)

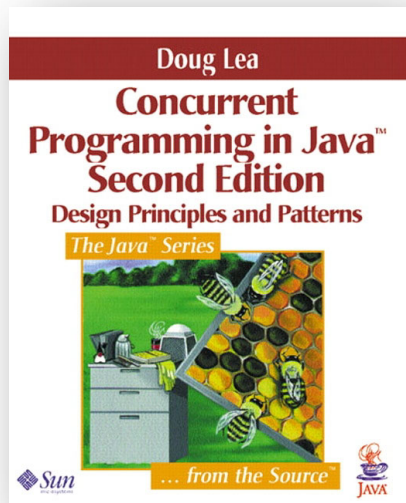
# INTRODUCCIÓN

Año	Versión Java	Concurrencia
1995	Java 1.02	Hilos y monitores (exclusión mutua y sinc condicional)
1996	Java 1.1	
1998	Java 2 1.2	Estructuras de datos concurrentes. Se desaconsejó el uso de ciertas funciones de gestión de hilos por ser inseguras (stop, pause, resume)
2000	Java 2 1.3	
2002	Java 2 SE 1.4	
2004	Java 2 SE 1.5 o 5	Muchas mejoras en concurrencia: Java Memory Model, Ejecutores, Colas de procesamiento, Temporizadores, Mejores estructuras de datos concurrentes
2006	Java SE 6	Colas de procesamiento mejoradas
2011	Java SE 7	Framework Fork/Join
2014	Java SE 8	Mejoras en estructuras de datos concurrentes, lambdas, estructuras de datos con procesamiento funcional (declarativo) y ejecución paralela, Futuros mejorados
2017	..Java SE 8	... se continúa actualizando. Por ejemplo MAX_LOCKS → Máximo de bloqueos simultáneos que FileHandler puede manejar

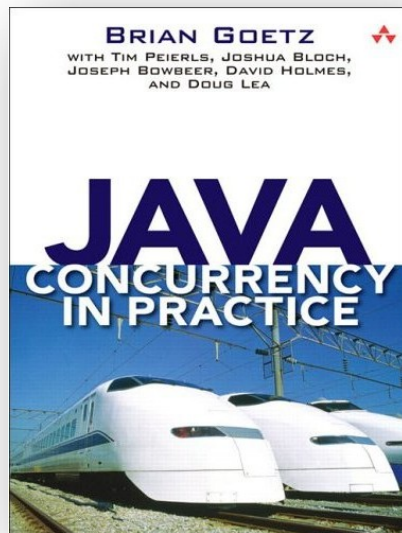
- En java se pueden usar **otros modelos de concurrencia** con **librerías**
  - **Actores:**
    - **Akka:** <http://akka.io/>
    - **Quasar:** <http://www.paralleluniverse.co/quasar/>
  - **Memoria software transaccional:**
    - **Multiverse:** <http://multiverse.codehaus.org>
  - **Comunicando procesos secuenciales (CSP):**
    - **JCSP:** <http://www.cs.kent.ac.uk/projects/ofa/jcsp/>
  - **Programación funcional:**
    - **Framework Collections en Java 8:** <http://java.dzone.com/articles/title-devovx-2012-java-8>
    - **RxJava:** <https://github.com/Netflix/RxJava>

- **Colección de libros** recomendables para dominar la **programación concurrente**, especialmente para sistemas de **memoria compartida**

1999



2006



2004



(y muchos más...)

<http://software.intel.com/en-us/articles/technical-books-for-multi-core-software-developers/>