

## Especificación de una Tarea de Investigación: Sobre Concurrencia en Java

Acrónimo de referencia rápido para efectos del curso: `JavaConcurr`.

### Introducción

Este documento especifica los elementos básicos de una tarea de investigación sobre el tema de concurrencia en la plataforma de Java.

Se aspira a que el producto de esta investigación pueda servirle a un estudiante de ingeniería de sistemas a obtener una introducción al tema, entender su relevancia y un punto de partida para profundizar en una o varias de las herramientas sobre las que, a manera de estudios de casos, se propone cubrir en el trabajo.

Se entiende todo esto dentro de los márgenes de tiempo establecidos por el semestre y los alcances que se indican adelante y a los que se espera poder llegar.

Si es necesario se podrán hacer ajustes a esta especificación las que permitan a los grupos de trabajo producir resultados razonables en calidad y tiempo.

Se busca que el aprendizaje sea de beneficio en el futuro profesional de los estudiantes, más allá de cumplir con un trabajo de un curso, más allá que por puntos en una nota únicamente.

### Objetivos generales de la Investigación

- Establecer, para los efectos de la investigación, un marco de referencia que caracterice el tema pedido y justifique su importancia.
- Introducir cada una de los temas y herramientas escogidos como representativas del marco de estudio.
- Contrastar características e inventariar referencias a algunas ventajas y desventajas de cada modelo o herramienta de la manera más objetiva posible.
- Enlazar los temas con los proyectos programados de forma que estos sirvan de demostraciones del aprendizaje práctico de los conceptos. En especial lograr conexiones con el tema de programación funcional.
- Organizar el reporte de forma que el lector pueda dar unos primeros pasos en la comprensión y uso de los conceptos y herramientas (servir como un “getting started”).
- Documentar y referenciar el material usado de una forma que se aproxime lo mejor posible a los objetivos planteados dadas las condiciones del curso y en respeto a las fuentes utilizadas, pero con elaboración propia, citando textualmente en los casos que sea obligatorio.
- Tratar de usar material más allá de “blogs sesgados” y sitios de intercambio de opiniones o de influencia comercial sesgada. Si es posible tratar de incorporar al menos una fuente obtenida y consultada del SIDUNA relacionada con algún aspecto de la investigación.

**Este material es de uso exclusivo para los estudiantes del curso del autor. No distribuir ni compartir sin permiso explícito del mismo.**

- No abusar del uso de una IA para la generación del trabajo. Hacer explícito en cada caso que se use.
- Destacar enlaces con los temas propios directos del curso, en especial el dualismo declarativo versus operacional que es central en el curso.

### Criterios de Evaluación

El profesor revisará a su criterio personal y profesional el producto tanto en forma como en fondo. Los criterios pueden ser subjetivos dada la naturaleza de la investigación. Los criterios incluyen al menos los siguientes:

- a) Cobertura de los objetivos planteados
- b) Calidad del marco de referencia
- c) Cobertura de los casos de herramientas, temas y demos solicitados
- d) Originalidad y calidad en la cobertura de cada caso de estudio, donde se aprecie elaboración propia.
- e) Respeto a una estructura formal de reporte en presentación externa (forma), contenido y uso de fuentes y respeto a la propiedad intelectual de esas fuentes.
- f) Legibilidad y calidad de la presentación.

### Objetivos y Temas Para Cubrir

Se debe generar un reporte técnico sobre manejo de concurrencia en Java que tiene por objetivo permitirles a los estudiantes autores y a los lectores poder obtener y expresar una comprensión sobre el tema de concurrencia y la forma en que la plataforma JAVA lo implementa. Para lograr ese objetivo, el reporte debe abarcar una introducción al tema de concurrencia, en general, los problemas de su manejo y opciones que JAVA ofrece para atacarlos bajo el concepto de hilo. Debe estar actualizado para cubrir las propuestas de hilos virtuales y variables de ámbito (“scoped variables”). Debe permitir entender las ventajas y desventajas de estos nuevos enfoques en relación con hilos nativos. Debe comparar y establecer las principales diferencias, ventajas y desventajas, entre hilos nativos y virtuales. De dar ejemplos demostrativos usando problemas clásicos de manejo de concurrencia.

Se adjunta como anexo una propuesta de estructura de contenido generada por ChatGPT.

### Herramientas esperadas a Desarrollar y Cubrir

La investigación debe cubrir hilos, hilos virtuales, futuros, ExecutorFramework. Las primeras son en este momento (java 20) propuestas preliminares en etapa de maduración.

### Estructura, Presentación y Normas

Debe ser un reporte técnico formal, con portada, abstract, índice de contenido (incluyendo de figuras, tablas, diagramas, etc), bibliografía (formato APA) y anexos si los hay. Toda frase o afirmación o imagen o fuente que no sea de propia elaboración debe ser debidamente citada (lo

**Este material es de uso exclusivo para los estudiantes del curso del autor. No distribuir ni compartir sin permiso explícito del mismo.**

contrario anulará en reporte). Si se usa una IA como ChatGPT eso debe consignarse como si fuera una fuente formal. No debe abusarse de esa técnica, de forma que resulte que el reporte fue generado en su mayoría por la IA. Eso anulará el reporte. Debe haber elaboración propia. En particular debe el grupo poder demostrar la compilación y ejecución de los ejemplos propios de código.

El reporte no debe exceder 30 páginas en total, espacio sencillo, Arial 11 puntos. Debe tener encabezados y subencabezados debidamente numerados para brindar una estructura ordenada y coherente. Debe tener numeración de página. Toda figura, tabla o similar debe tener su “caption” referenciado en el índice de contenido.

### Valor en la Nota

Este trabajo aporta al 10% de la nota dedicada a Investigación en la carta al estudiante con un valor de al menos un 85%. Los puntos extras (ver más adelante) pueden hacer la nota mayor que 100 hasta un máximo de un 20% adicional. **Este trabajo podrá ser usado por el docente, según su calidad, más allá del puntaje asignado como un elemento de valoración y aprecio del interés por aprender y ganar el curso y reflejarlo significativamente en el cálculo de la nota final del curso.**

### Forma de Trabajo, Entrega y Fechas

La versión final se espera sea entregada en la semana 17 en forma y hora a definir al menos una semana antes. Se podrán pedir avances progresivos obligatorios de entregas que determinen el cálculo de la nota final del trabajo. En particular un primer avance, en la semana 9 aproximadamente, que comprenda una cobertura de al menos un 35%. Se podrán pedir avances intermedios cortos antes de la semana 9 y la semana 16, a criterio del docente. La versión final puede tener mejoras sobre el primer avance, mejoras que podrán ser tomadas en cuenta.

### Rol del Coordinador

El coordinador de cada grupo deberá ser el canal principal de comunicación entre el grupo y el docente, se espera ejerza el liderazgo necesario para que el trabajo avance constante y positivamente y se detecten y corrijan problemas oportunamente en coordinación con el docente. **Se le dará el poder por parte del docente de tener una opinión a la hora de evaluar el proyecto y el trabajo de los miembros del grupo.** Se espera que el grupo trabaje como tal y se repartan las tareas de manera equitativa y que haya la interacción que permita compartir el aprendizaje mutuamente. Cada grupo debe poder demostrar formalmente la cantidad de trabajo invertido a un momento dado en el semestre.

### Puntos Extra

Se pueden, a criterio del profesor, conceder puntos extra de hasta un 20% adicional sobre la nota del trabajo. Debe ser propuestos por los estudiantes y aceptados de previo por el profesor.

Posibles iniciativas para extras (cada grupo puede sugerir otras que se relacionen con la temática):

- a) Otras herramientas y ejemplos distintos o complementarias a las pedidas.

**Este material es de uso exclusivo para los estudiantes del curso del autor. No distribuir ni compartir sin permiso explícito del mismo.**

- b) Hilos y concurrencia en Prolog versus Java
- c) Consulta de personas con criterio profesional que hayan revisado y emitido una “validación” de criterio sobre el trabajo (no puede ser el docente, por supuesto, ni otros estudiantes). Deben ser aprobadas por el docente con antelación.

## Anexo. Estructura de contenido propuesta por ChatGPT

### Prompt:

Darme una estructura de contenido para un reporte técnico sobre concurrencia en Java. Debe abarcar nociones básicas de concurrencia, en general y en particular en Java. Debe cubrir modelos alternativos a hilos como futuros, el executor framework. Debe cubrir "virtual threads" y "scoped variables". Debe incluir una sección de ejemplos integrales usando hilos virtuales en resolución de problemas clásicos de concurrencia.

### Respuesta:

Estructura de contenido para un reporte técnico sobre concurrencia en Java:

#### I. Introducción

- A. Definición de concurrencia
- B. Importancia de la concurrencia en el desarrollo de software
- C. Objetivos del reporte

#### II. Nociones básicas de concurrencia

- A. Concepto de proceso y hebra (hilo)
- B. Problemas comunes en concurrencia
- C. Ventajas y desventajas de la concurrencia

#### III. Concurrencia en Java

- A. Modelo de hilos en Java
  - 1. Creación y manejo de hilos
  - 2. Sincronización de hilos
- B. Problemas comunes en la concurrencia en Java
  - 1. Condición de carrera (race condition)
  - 2. Deadlock

- 3. Inanición (starvation)
- C. Modelos alternativos a hilos en Java
  - 1. Futuros (Futures)
  - 2. Executor Framework
  - 3. Ventajas y casos de uso de los modelos alternativos
- IV. Virtual Threads en Java
  - A. Introducción a los hilos virtuales
  - B. Diferencias entre hilos virtuales y hilos tradicionales
  - C. Beneficios de los hilos virtuales en términos de rendimiento y escalabilidad
- V. Scoped Variables en Java
  - A. Concepto de variables en contexto (scoped variables)
  - B. Uso de scoped variables para evitar condiciones de carrera
  - C. Implementación y mejores prácticas de uso
- VI. Ejemplos integrales utilizando hilos virtuales
  - A. Problema clásico de productor-consumidor
    - 1. Implementación con hilos virtuales
    - 2. Ventajas y comparación con la implementación con hilos tradicionales
  - B. Problema de lectores y escritores
    - 1. Implementación con hilos virtuales
    - 2. Comparación con la implementación con hilos tradicionales
  - C. Otros ejemplos relevantes de concurrencia resueltos con hilos virtuales
- VII. Conclusiones
  - A. Resumen de los puntos clave
  - B. Ventajas y desafíos de la concurrencia en Java
  - C. Recomendaciones para el uso de hilos virtuales y scoped variables

#### VIII. Referencias bibliográficas

Esta estructura de contenido proporciona una base sólida para un reporte técnico sobre concurrencia en Java, abordando desde nociones básicas de concurrencia hasta los modelos alternativos de hilos, como los futuros y el Executor Framework. También se incluyen secciones dedicadas a los hilos virtuales y las variables en contexto (scoped variables), así como ejemplos integrales que demuestran el uso de hilos virtuales en la resolución de problemas clásicos de concurrencia.