

# Sistemas Concurrentes y Distribuidos Examen II



*Escuela Técnica Superior de Ingenierías  
Informática y de Telecomunicación*

Los Del DGIIM, [losdeldgiim.github.io](https://losdeldgiim.github.io)

Doble Grado en Ingeniería Informática y Matemáticas  
Universidad de Granada



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0).

Eres libre de compartir y redistribuir el contenido de esta obra en cualquier medio o formato, siempre y cuando des el crédito adecuado a los autores originales y no persigas fines comerciales.

# Sistemas Concurrentes y Distribuidos Examen II

Los Del DGIIM, [losdeldgiim.github.io](https://losdeldgiim.github.io)

Granada, 2025

**Asignatura** Sistemas Concurrentes y Distribuidos.

**Curso Académico** 2024-25.

**Grupo** Único.

**Descripción** Examen parcial de los Temas 1, 2 de SCD.

**Fecha** 03 de Diciembre 2025.

**Profesor** Manuel Isidoro Capel Tuñón

**Ejercicio 1** (3.5 puntos). Preguntas de respuestas alternativas, visitar el enlace:

<https://losdeldgiim.github.io/subjects/SCD/Exámenes/Ex2.html>

Cada opción correcta suma (0,1). Dos preguntas erróneas anulan 1 correctamente respondida.

**Resolución de problemas: 65 %<sup>1</sup>**

**Ejercicio 1.** Considérese el siguiente programa concurrente, donde cada instrucción entre `{}y {}se ejecuta de forma atómica:`

$$\{x == 0 \wedge y == 0 \wedge n > 0\} \quad (1)$$

```
cobegin
< x = x + 1 > ||
while (y < n) do
begin
    < y = y + 1 >;
end
coend
x == 1 ∧ y == n
```

Se pide:

1. Determinar si el triple de Hoare  $\{x == 0 \wedge y == 0 \wedge n > 0\} P \{x == 1 \wedge y == n\}$  concurrente es demostrable usando la Lógica de Programas (incluyendo las reglas del while y la regla de composición concurrente).
2. En caso afirmativo, construir una demostración completa, indicando:
  - El invariante del bucle while.
  - Las triples atómicas utilizadas.
  - Las triples resultantes de combinar cada proceso mediante: la regla de composición secuencial, la
  - regla de composición iterativa (para el bucle), y la regla de composición concurrente.
3. Discutir si existe interferencia entre los dos procesos. En particular, justificar si el proceso del bucle puede invalidar las aserciones sobre x (o viceversa) según las reglas de interferencia de Hoare para concurrencia.

**Ejercicio 2.** Se diseña una solución de exclusión mutua en memoria compartida donde pueden entrar como máximo n procesos simultáneamente en sección crítica, con  $n \geq 2$ .

En esta versión:

- Se debe garantizar una cola FIFO para procesos que esperan cuando el afor está completo.
- El cerrojo permite hasta n procesos dentro, pero la cola usa un único contador protegido.
- Al salir un proceso, solo se desbloquea a un proceso de la cola, no a todos.

Tareas:

---

<sup>1</sup>Solo había que resolver 2 ejercicios y el otro podía mejorar la calificación.

1. Dar pseudocódigo de entrada y salida.
2. Probar exclusión mutua extendida, ausencia de interbloqueo y equidad.

**Ejercicio 3.** Se desea un monitor CentroRecursos para gestionar acceso de procesos de tipo 1 y tipo 3 con estas reglas:

1. Solo procesos del mismo tipo pueden estar dentro simultáneamente.
2. Nunca pueden coincidir procesos de tipos distintos.
3. Si hay procesos esperando de ambos tipos, tiene prioridad el tipo que haya acumulado más esperas totales.
4. Al terminar, un proceso despierta a uno del tipo con mayor prioridad acumulada.

El monitor exporta: entrar(tipo), salir(tipo). Tareas:

1. Diseñar el monitor con variables internas, colas de condición y contradores de espera.
2. Mostrar cómo se aplica la prioridad dinámica.
3. Justificar exclusión mutua, ausencia de interbloqueo y no inanición.