

Apellidos: \_\_\_\_\_ Nombre: \_\_\_\_\_

D.N.I: \_\_\_\_\_ Firma: \_\_\_\_\_

Prácticas realizadas: \_\_\_\_\_ expuestas: \_\_\_\_\_

1	2	3	4	5	6	7	Suma
(8)	(8)	(6)	(8)	(6)	(6)	(8)	(50)

**Pregunta 1:** [8 Puntos]

Queremos simular un semáforo general con unos semáforos binarios. Detalla el código y razona por qué tu solución funciona.

**Pregunta 2:** [8 Puntos]

¿Qué se entiende con “espera finita” de un proceso cuando quiere acceder a un recurso?  
¿Cuáles son las propiedades de dicha espera finita que hay que tener en cuenta cuando se analiza el comportamiento en una aplicación concurrente concreta?

**Pregunta 3:** [6 Puntos]

¿Por qué es la instrucción hardware CAS más potente que la instrucción hardware TAS?  
(Ayuda: reflexiona sobre su uso para pasar el control sobre una sección crítica de hilo a hilo.)

**Pregunta 4:** [8 Puntos]

Describe brevemente el patrón de diseño *proactor* y su posible uso en un programa concurrente.

**Pregunta 5:** [6 Puntos]

Describe el concepto de reloj lógico en un sistema distribuido con el cual los procesos *sincronizan* sus relojes locales de cierto modo.

**Pregunta 6:** [6 Puntos]

Razona sobre la *justicia* presente en el algoritmo asimétrico visto en clase que garantiza la exclusión mutua para el acceso a la sección crítica.

**Pregunta 7:** [8 Puntos]

Reflexiona brevemente sobre las limitaciones para la programación concurrente en el uso de `synchronized` en Java. ¿Cuáles son algunas de las principales mejoras introducidas con los paquetes `java.util.concurrent` y `java.util.concurrent.atomic`?