

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

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

1	2	3	4	5	6	7	8	9	Suma
(4)	(6)	(4)	(4)	(6)	(4)	(4)	(8)	10	(50)

**Pregunta 1:** [4 Puntos] Describe brevemente los patrones de diseño *actor* y *proactor* y su posible uso en un programa concurrente destacando también sus diferencias.

**Pregunta 2:** [6 Puntos] Describe brevemente el patrón de diseño *aviso de hecho* (*double scoped locking*). Razona críticamente sobre su implementación en Java, C++03 y C++11.

**Pregunta 3:** [4 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 4:** [4 Puntos] ¿Qué se entiende bajo el *principio de la bandera*? ¿Cómo se comprueba? ¿Para qué se usa?

**Pregunta 5:** [6 Puntos] Implementa en pseudo-código el sistema productor–consumidor con cola de comunicación finita donde varios productores y consumidores usan la misma cola.

**Pregunta 6:** [4 Puntos] Explica la semántica del modificador `volatile` de Java y su uso en programas concurrentes.

**Pregunta 7:** [4 Puntos] ¿Cuáles son las condiciones que se tienen que cumplir para que se produzca un bloqueo entre procesos?

**Pregunta 8:** [8 Puntos] Reflexiona brevemente sobre las posibilidades y limitaciones para la programación concurrente en el uso de `synchronized` en Java.

¿Cuáles son algunas de las principales aportaciones introducidas con los paquetes `java.util.concurrent` y `java.util.concurrent.atomic`?

#### Pregunta para las prácticas:

**Pregunta 9:** [10 Puntos] Durante el curso hemos realizado 4 tareas principales en las prácticas. Resume brevemente cuales eran los principales puntos críticos para realizar una solución/implementación correcta de cada una de ellas y cuales eran las “lecciones” aprendidas una vez haber visto las soluciones/implementaciones funcionando (refiriéndose a, por ejemplo: observaciones, dificultades y casos límites por tratar, comprobación de corrección, tiempo de ejecución en diferentes ordenadores y/o sistemas operativos, alternativas de implementación, estrategias y técnicas de depuración etc.).