

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

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

1	2	3	4	5	6	7	8	9	10	Suma
(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(30/50)

Los estudiantes **con** evaluación continua deben constestar solamente **seis** (6) de las 10 preguntas. ¡Ojo, en caso que se conteste más de seis preguntas se eliminará la/s respuesta/s con más puntos obtenidos!  
 Tiempo: 1 hora y media.

Los estudiantes **sin** evaluación continua deben constestar todas las preguntas, más las preguntas relacionadas con las prácticas. Tiempo: 3 horas y media.

**Pregunta 1:** [5 Puntos] ¿Cómo está implementado en Java el concepto de *región crítica*?

**Pregunta 2:** [5 Puntos] ¿Por qué puede ser ventajoso usar un programa concurrente incluso en un entorno mono-procesador?

**Pregunta 3:** [5 Puntos] ¿Cuáles son las condiciones que se tienen que cumplir para que se produzca un bloqueo entre procesos?

**Pregunta 4:** [5 Puntos] Describe brevemente el funcionamiento de la instrucción hardware CAS disponible para muchos procesadores.

**Pregunta 5:** [5 Puntos] ¿Qué se entiende bajo el *principio de la bandera*? ¿Cómo se comprueba? ¿Para qué se usa?

**Pregunta 6:** [5 Puntos] ¿Cómo se puede prevenir la apariencia de un bloque entre procesos en muchas ocasiones durante el diseño de una aplicación?

**Pregunta 7:** [5 Puntos] ¿Qué se entiende bajo *espera activa*, cuáles son los problemas principales, y qué se puede hacer para mejorar?

**Pregunta 8:** [5 Puntos] ¿Cuáles son las propiedades principales de códigos *Raptor*?

**Pregunta 9:** [5 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 10:** [5 Puntos] Explica la semántica del modificador `volatile` de Java y su uso en programas concurrentes.