

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/as estudiantes **con** evaluación continua deben contestar solamente las primeras **seis** (6) preguntas (Parte I). Tiempo: 1 hora y media.

Los/as estudiantes **sin** evaluación continua deben contestar todas las diez (10) preguntas (Parte I+II), más las preguntas relacionadas con las prácticas (Parte III). Tiempo: 3 horas y media.

Una vez empezado el examen, el estudiante asume que cualquier uso de un dispositivo móvil sin previo aviso, resulta en un suspenso inmediato del examen con puntuación 0, informe a los órganos competentes de la Universidad, y posibilidad a un futuro examen en este curso solamente mediante orden escrito del órgano superior.

Parte I

Pregunta 1: [5 Puntos] La *región crítica* es un concepto abstracto útil para la programación concurrente. ¿Cómo está implementado en Java este concepto? Razona críticamente sobre sus ventajas y desventajas. ¿Cuál es el desafío principal cuando se traslada el concepto a un entorno distribuido?

Pregunta 2: [5 Puntos] Durante la fase de análisis para una nueva aplicación ¿cuáles serían posibles requisitos que sugieren el diseño de un programa concurrente?

Pregunta 3: [5 Puntos] ¿Cuáles son las condiciones que se tienen que cumplir para que se produzca un bloqueo entre procesos? ¿Cuál es el peligro principal de un bloqueo entre procesos? ¿Cómo se puede prevenir la apariencia de un bloqueo entre procesos en muchas ocasiones ya durante el diseño de una aplicación? Razona críticamente sobre sus ventajas y limitaciones.

Pregunta 4: [5 Puntos] Compara brevemente el funcionamiento de la instrucción hardware TAS con las instrucciones CAS ambas disponibles para muchos procesadores. ¿Para qué se pueden utilizar? ¿Cuál ventaja destaca de CAS sobre TAS?

Pregunta 5: [5 Puntos] Alguien te ofrece unos protocolos de entrada y salida para garantizar la exclusión mutua de una sección crítica involucrando dos procesos. ¿Cómo verificas si los protocolos son correctos (en el sentido que garantizan la exclusión mutua) antes de comprarlos?

Pregunta 6: [5 Puntos] Durante la fase de depuración de un programa concurrente, al/a responsable de realizar las pruebas ocurre la idea de producir diferentes versiones del código introduciendo simples comandos de `sleep` al azar entre líneas sin cambiar nada más. ¿Crees que es buena idea? En caso que sí, ¿qué tipo de fallos se pueden detectar? ¿Te ocurre algún inconveniente de tal prueba (a parte que las pruebas siempre consumen tiempo en la producción de una aplicación software)?

Parte II

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] Explica la semántica del modificador `volatile` de Java y su uso en programas concurrentes, entre otras, ¿qué tiene que ver con una relación *ha-pasado-antes*, es decir, qué garantías tiene el/a programador/a con el uso de `volatile` escribiendo y leyendo variables en su código?

Pregunta 9: [5 Puntos] Describe brevemente cinco (5) características disponibles en los paquetes `java.util.concurrent` y `java.util.concurrent.lock`. Destaca en cada caso su especial relevancia para su uso en programas concurrentes.

Pregunta 10: [5 Puntos] Detalla el teorema llamado el *principio de la bandera* y compruébalo.