CD Examen Teoría y Práctica **Dr. Arno Formella, Diciembre de 2011**

Apellidos: _____ **Nombre:** _____

D.N.I: _____ **Firma:** _____

Prácticas realizadas: _____ **expuestas:** _____

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

Pregunta 1: [6 Puntos] ¿Cuáles son técnicas, metodologías, y herramientas para depurar programas concurrentes? Razona críticamente sobre sus aplicabilidades y como algunas has aplicado tu en las prácticas.

Pregunta 2: [9 Puntos] ¿Por qué es interesante garantizar una *espera finita* para todos los procesos en todas las situaciones cuando estén competiendo por los recursos en un programa concurrente? ¿Qué puede pasar si no se puede garantizar para todos los procesos? ¿Cuáles son los métodos disponibles para solucionar la *peor* situación? Razona sobre la aplicabilidad de los métodos en programas concurrentes.

Pregunta 3: [6 Puntos] ¿Qué se entiende bajo el *principio de la bandera*? ¿Cómo se comprueba? ¿Para qué se suele usar?

Pregunta 4: [4 Puntos] En un sistema distribuido donde los nodos de procesamiento están conectados por canales de comunicación se pueden provocar diferentes tipos de fallos enviando mensajes de un nodo al otro. Enumera dichos tipos de fallos y razona sobre técnicas disponibles para superar la deficiencia del canal.

Pregunta 5: [6 Puntos] Reflexiona brevemente sobre el concepto de atomicidad implementado en Java y su mejora con la introducción del paquete `java.util.concurrent`.

Pregunta 6: [9 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 el paquete `java.util.concurrent`?

Pregunta 7: [5 Puntos] En las prácticas vimos un planificar para el acceso a un recurso compartido. Describe brevemente las especificaciones del planificador que implementaste y argumenta como comprobaste que el planificador no lleva a ningún hilo a una espera infinita.

Pregunta 8: [5 Puntos] En las prácticas hemos visto dos tipos de estructuras de datos concurrentes: una implementación propia de una lista concurrente que dispone de las operaciones *insertar*, *borrar*, e *iterar*, y una tabla de dispersión que está disponible en el paquete `java.util.concurrent` llamada `ConcurrentHashMap`. Describe brevemente cuáles son las principales características de las dos implementaciones. Razona brevemente sobre posibles aplicaciones de tales estructuras de datos concurrentes.