**CD Examen Teoría** **Dr. Arno Formella, Mayo/Junio de 2012**

**Apellidos:** _____ **Nombre:** _____

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

**Prácticas realizadas:** _____ **expuestas:** _____

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Suma |
|---|---|---|---|---|---|---|---|---|
| (4) | (6) | (6) | (4) | (6) | (4) | (6) | (4) | (40) |

**Pregunta 1:** [4 Puntos] ¿Por qué puede ser ventajoso usar un programa concurrente incluso en un entorno mono-procesador?

**Pregunta 2:** [6 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 distribuidos.

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

**Pregunta 5:** [6 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 6:** [4 Puntos] Explica la semántica del modificador `volatile` de Java y su uso en programas concurrentes.

**Pregunta 7:** [6 Puntos] Reflexiona brevemente sobre el concepto de atomicidad implementado en Java y su mejora con la introducción del paquete `java.util.concurrent`.

**Pregunta 8:** [4 Puntos] ¿Cómo implementarías un ping-pong múltiple perfecto? ¿Por qué llamamos la versión *perfecto*?