

Apellidos: _____ Nombre: _____

D.N.I: _____ Firma: _____

1	2	3	4	5	6	7	8	9	10	Suma
(4)	(4)	(3)	(4)	(4)	(5)	(4)	(4)	(3)	(5)	(40)

Pregunta 1: [4 Puntos] ¿Por qué se puede esperar en cierto tipo de aplicaciones una mejora en tiempo de ejecución de un programa concurrente comparándolo con un programa secuencial? ¿Por qué puede ser ventajoso usar un programa concurrente incluso en un entorno mono-procesador?

Pregunta 2: [4 Puntos] Hemos visto un semáforo general implementado con unos semáforos binarios. Detalla el código y razona por qué funciona asumiendo el funcionamiento correcto de un semáforo binario.

Pregunta 3: [3 Puntos] ¿Por qué es interesante garantizar una “espera finita” para todos los procesos en todas las situaciones cuando estén compitiendo por los recursos en un programa concurrente?

Pregunta 4: [4 Puntos] Describe brevemente la posibilidad usando el “compare-and-swap” como ayuda en hardware para implementar acceso con exclusión mutua a la memoria y destaca su diferencia con una implementación usando la operación “test-and-set”.

Pregunta 5: [4 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 6: [5 Puntos] ¿Cuáles son las condiciones que se tienen que cumplir para que se produzca un bloqueo entre procesos? Describe los tres métodos presentados en clase para solventar el problema del bloqueo. Razona brevemente sobre sus propiedades en entornos distribuidos.

Pregunta 7: [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 8: [4 Puntos] ¿Qué se entiende bajo el *principio de la bandera*? ¿Cómo se comprueba? ¿Para qué se usa?

Pregunta 9: [3 Puntos] Explica la semántica del modificador `volatile` de Java y su uso en programas concurrentes.

Pregunta 10: [5 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`?