

Apellidos: _____ Nombre: _____

D.N.I.: _____ Firma: _____

Prácticas presentadas: _____

1	2	3	4	5	6	7	8	9	10	Suma
(6)	(8)	(4)	(6)	(8)	(4)	(6)	(6)	(6)	(6)	(60)

Pregunta 1: [6 Puntos]

Reflexiona brevemente sobre las ventajas y desventajas de sustituir en una empresa los ordenadores mono-procesador por nuevos equipos con procesadores multi-core.

Pregunta 2: [8 Puntos]

Queremos simular un semáforo general con unos semáforos binarios. Detalla el código y razona por qué tu solución funciona.

Pregunta 3: [4 Puntos]

¿Por qué es la instrucción hardware CAS más potente que la instrucción hardware TAS? (Ayuda: reflexiona sobre su uso para pasar el control sobre una región crítica de hilo a hilo.)

Pregunta 4: [6 Puntos]

Describe brevemente el patrón de diseño *active object* y su posible uso en un programa concurrente.

Pregunta 5: [8 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 eficiencias en entornos distribuidos.

Pregunta 6: [4 Puntos]

Razona por qué el algoritmo asimétrico visto en clase garantiza la exclusión mutua para el acceso a la región crítica (es decir, el gato y el perro nunca llegan juntos al patio).

Pregunta 7: [6 Puntos]

Reflexiona brevemente sobre las limitaciones para la programación concurrente en el uso de `synchronized` en Java. ¿Cuáles son algunas de las principales mejoras introducidas con los paquetes `java.util.concurrent` y `java.util.concurrent.atomic`?

Pregunta 8: [6 Puntos]

¿Cómo se puede conseguir que solo un hilo construye un objeto en Java (es decir, un segundo hilo que lo intente ya encuentra el objeto construido y lo nota)? ¿Cuándo hace falta una construcción de tal tipo?

Preguntas para las prácticas:

Pregunta 9: [6 Puntos]

En el ejercicio del PingPong vimos que el uso de `notify()` y `notifyAll()` resultaba en un número de jugadas entre los jugadores en cierto intervalo de tiempo que dependía del número de jugadores participando en el juego.

- ¿Por qué eso era el caso?
- ¿Cómo conseguimos finalmente un número constante de jugadas, en cierto intervalo de tiempo, independientemente del número de jugadores participantes?
- ¿Cómo implementarías un método para medir bien el número de jugadas en cierto tiempo (sin medir, dentro de lo posible, la influencia de otras partes del programa)?

Pregunta 10: [6 Puntos]

En las prácticas hemos utilizado la implementación de una lista concurrente que disponía de las operaciones *insertar*, *borrar*, e *iterar*.

- Describe brevemente cuáles fueran las técnicas usadas para que varios hilos pueden manipular la lista concurrentemente (por lo menos en aquellas situaciones en las cuales las modificaciones tienen lugar en posiciones suficientemente separadas en la lista), es decir, hacer uso de las tres operaciones, sin que exista la posibilidad de un bloqueo mutuo.
- Usamos dicha lista para implementar una tabla de hash simple. Describe brevemente cómo implementaste las operaciones principales de tal tablas que son *insertar*, *buscar*, y *borrar*.
- ¿Cuáles consideras las principales desventajas de dicha solución comparándolo con la tabla de hash disponible en el paquete `java.util.concurrent`?