

(2 puntos) 3. Dada la siguiente clase *thread*:

<pre>static class T extends Thread { volatile private int y; public T (int y) { this.y = y; } }</pre>	<pre>public void run() { y++; } }</pre>
---	---

Se pide marcar la afirmación correcta.

- (a) `y++` nunca es una sección crítica.
- (b) `y++` puede ser una sección crítica.

(1 punto) 4. Supongamos un programa concurrente con E procesos ($E > 0$) que ejecutan repetidamente operaciones $r.esperar()$ y D procesos ($D > 0$) que ejecutan repetidamente operaciones $r.despertar(n)$ con $n \leq E$, siendo r un recurso compartido del tipo especificado a continuación:

C-TAD Señal

OPERACIONES

ACCIÓN esperar:

ACCIÓN despertar: $\mathbb{N}[e]$

SEMÁNTICA

DOMINIO:

TIPO: Señal = \mathbb{N}

INICIAL: self = 0

CPRE: self > 0

esperar()

POST: self^{pre} = self + 1

PRE: $n \leq E$

CPRE: Cierto

despertar(n)

POST: self = n

Se pide señalar la respuesta correcta.

- (a) El recurso compartido r podría pasar a lo sumo por E estados distintos.
- (b) El recurso compartido r podría pasar a lo sumo por $E + 1$ estados distintos.

(1 punto) 5. Dado el programa concurrente descrito en la pregunta 4. Se pide señalar la respuesta correcta.

- (a) El programa concurrente cumple la propiedad de ausencia de interbloqueo.
- (b) El programa concurrente no cumple la propiedad de ausencia de interbloqueo.

(2 puntos) 6. Obsérvese la siguiente implementación del recurso compartido Señal especificado en la pregunta 4:

<pre>class Sennal { private Semaphore sennal = new Semaphore(0); private Semaphore mutex = new Semaphore(1); }</pre>	
<pre>public void esperar() { mutex.await(); sennal.await(); mutex.signal(); }</pre>	<pre>public void despertar(int n) { mutex.await(); for (int i = 0; i < n; i++) sennal.signal(); mutex.signal(); }</pre>

La idea principal consiste en que el semáforo `sennal` represente el valor interno (de tipo \mathbb{N}) del recurso.

Se pide señalar la respuesta correcta.

- (a) Es una implementación correcta del recurso compartido.
- (b) Es una implementación incorrecta del recurso compartido.

Apellidos:

Nombre:

Matrícula:

- (2 puntos) 7. **Se pide** completar la siguiente especificación de un recurso compartido para que su comportamiento sea idéntico al de semáforos contadores.

C-TAD Semaforo

OPERACIONES

ACCIÓN await:

ACCIÓN signal:

SEMÁNTICA

DOMINIO:

TIPO: *Semaforo* =

INVARIANTE:

INICIAL:

CPRE:

await()

POST:

CPRE:

signal()

POST:

(Página intencionadamente en blanco, puede usarse como hoja en sucio).