TEST REPASO Unidades Didácticas 7 a 9 - | MODELO A | Concurrencia y Sistemas Distribuidos

NOMBRE:		DNI:	

Este examen tiene una duración total de 70 minutos y tiene una puntuación máxima de **10 puntos**.

Indique, para cada una de las siguientes **40 afirmaciones**, si éstas son verdaderas (**V**) o falsas (**F**).
Cada respuesta vale: correcta= 0.25, errónea= -0.25, vacía=0.

Sobre los sistemas distribuidos:

1. La transparencia de distribución consiste en informar al usuario en qué ordenadores los procesos y los recursos están distribuidos, para así proporcionar una imagen de sistema único.	
2. En aplicaciones como Instagram se debe mantener una consistencia fuerte para así ofrecer una mayor transparencia de distribución de los datos.	
3. Si se dan las premisas para poder replicar un servicio mediante replicación activa y disponemos de un protocolo de difusión de mensajes que garantice la entrega de los mensajes a las diferentes réplicas en el mismo orden, podremos lograr un sistema con consistencia fuerte.	
4. Para aumentar la escalabilidad en sistemas distribuidos podemos delegar parte del procesamiento a los clientes, y repartir tanto las tareas como los datos entre múltiples nodos servidores	
5. En un sistema distribuido se deben ofrecer simultáneamente consistencia fuerte, alta disponibilidad y permitir que, en caso de particiones, todas ellas sigan funcionando.	
6. Utilizar replicación activa implica emplear algún algoritmo de difusión a grupo que garantice la entrega de los mensajes en cierto orden.	
7. Las particiones son un ejemplo de fallo bizantino.	
8. Decimos que un fallo es detectable si un nodo A puede determinar que otro nodo B ha fallado, por ejemplo, porque la respuesta proporcionada por el nodo B está fuera de rango.	
9. Si se emplea un servicio de pertenencia a grupo, cuando un nodo A sospecha del fallo de otro nodo B, debe comunicarlo a dicho servicio para que se acuerde el posible fallo de B.	
10. Por disponibilidad entendemos que un sistema no merma sus prestaciones al aumentar número de usuarios, recursos, nodos o peticiones simultáneas.	

Sobre la comunicación en sistemas distribuidos:

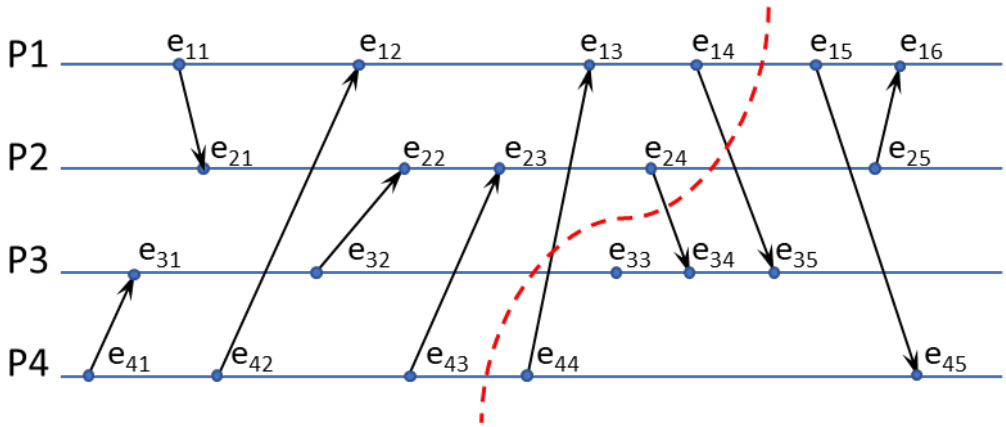
11. Los distintos tipos de direccionamiento pueden ser asincrónicos, sincrónicos en la entrega o sincrónicos en la respuesta.	
12. La persistencia es una característica que poseen aquellos middlewares de comunicación que pueden guardar mensajes pendientes de entrega.	
13. En Java RMI, los objetos que implementan la interfaz java.rmi.Remote que se pasan como argumentos en las invocaciones a métodos remotos, siempre se pasan por referencia.	
14. Para utilizar RMI, todos los objetos invocables de forma remota deben estar registrados en el servidor de nombres (rmiregistry)	
15. Para realizar cualquier llamada a un servicio web RESTful se requiere emplear el método HTTP GET, seguido de un objeto codificado en XML o JSON.	
16. En la respuesta a cualquier llamada a un servicio RESTful se requiere devolver un código de terminación basado en códigos de estado del protocolo HTTP.	
17. Java JMS proporciona un modelo de comunicación fundamentalmente sincrónico en la respuesta.	

18. En Java JMS, los objetos que implementan la interfaz Queue se crean llamando a métodos de la interfaz JMSConsumer.	

Sea un sistema con 7 nodos (N0, N1, .., N6) donde en un momento determinado están activos **cuatro de ellos**. Se asume que los nodos no activos permanecerán caídos durante todo el tiempo y que no se producirán más fallos. En el caso de la topología en anillo, asumimos que el sucesor de Ni es N(i+1 mod 7).:

19. Si el nodo N3 está activo e inicia el algoritmo Bully, podemos asegurar que al finalizar el algoritmo este nodo no será coordinador.	
20. Si el nodo N2 está activo e inicia el algoritmo de elección de líder en anillo, en algún momento de dicho algoritmo se enviará al nodo N0 un mensaje ELECCIÓN.	
21. Si el nodo N4 está activo e inicia el algoritmo Bully de elección de líder, enviará un mensaje ELECCIÓN a todos los demás nodos del sistema, independientemente de si están caídos o no.	
22. Si el nodo N2 está activo e inicia el algoritmo de elección de líder en anillo, podemos asegurar que al finalizar el algoritmo este nodo no será coordinador	

Observando la siguiente figura, podemos afirmar:



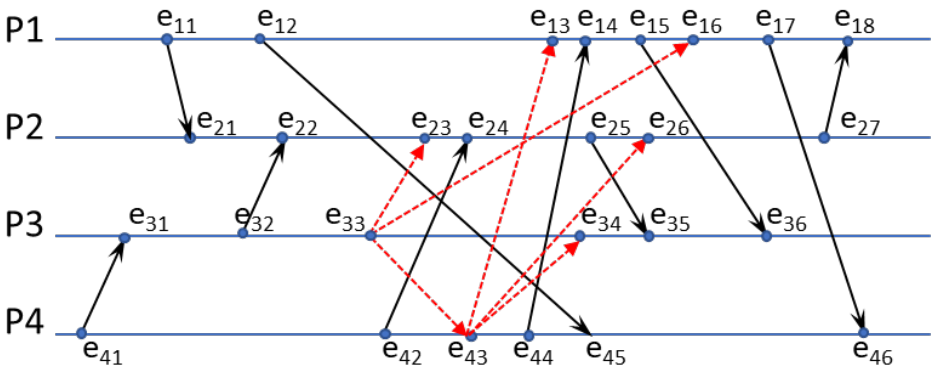
23. El reloj vectorial del evento e25 es [1, 5, 2, 3]	
24. El reloj lógico del evento e45 es 8 y el reloj lógico del evento e14 es 5	
25. Como se cumple que e41 → e16 podemos afirmar que el reloj lógico del evento e41 es menor que el reloj lógico del evento e41, es decir, C(e41) <C (e16)	
26. Los eventos e33 y e24 son concurrentes	
27. Si el algoritmo de Lamport etiqueta dos eventos a y b con el mismo valor lógico, no sabemos si "a->b" o "a||b"	
28. El corte de la ejecución tomado en la línea de puntos, es un corte consistente.	

Sobre los algoritmos distribuidos:

29. El algoritmo de Berkeley consta de 3 rondas: tras las dos primeras el coordinador ajusta su reloj y difunde un mensaje con ese valor de reloj a todos los clientes.	
30. En el algoritmo de Cristian, si un cliente C pregunta en el instante 6000 (según el reloj de C) al servidor su hora, recibe la respuesta en el instante 6030 (según el reloj de C), y esta respuesta es 6015, el cliente no necesita ni adelantar ni parar su reloj.	

31. En la variante distribuida (sin coordinador) del algoritmo de exclusión mútua, cada nodo ha de mantener una cola con la identidad de los nodos a los que está pendiente de responder OK.	
32. En el algoritmo de consenso distribuido tolerante a fallos visto en clase, se supone que los nodos pueden haber fallado previamente, pero que durante la ejecución del algoritmo los nodos no pueden fallar.	
33. Suponga que empleamos el algoritmo de exclusión mutua basado en un anillo visto en clase para coordinar el acceso a una sección crítica por parte de N nodos. Cuando uno de estos nodos pida entrar a su sección crítica, harán falta al menos N mensajes del algoritmo para que consiga entrar.	
34. Para obtener un orden total de los eventos de un sistema utilizando los valores de los relojes lógicos de Lamport, podemos añadirles el identificador de nodo como prefijo, y la ordenación resultante impone un orden en los eventos concurrentes, respetando la relación “ocurre antes” del resto de eventos.	
35. En el algoritmo de consenso distribuido tolerante a fallos visto en clase, si consideramos una red formada por 7 nodos que ejecuten este algoritmo, si el coordinador recibe 3 mensajes ACK y un NACK, entre los que se incluye el suyo propio, genera una decisión.	

En un sistema distribuido con cuatro nodos (P1 a P4), en un momento determinado el nodo P3 ha iniciado la ejecución del algoritmo Chandy-Lamport. En la figura se muestra una representación temporal de parte de dicha ejecución, donde las líneas discontinuas representan mensajes MARCA de la ejecución del algoritmo de Chandy-Lamport, las líneas continuas representan mensajes normales, mientras que "eX" representan los eventos de envío y/o recepción. Además, los mensajes MARCA que deban enviar los nodos P1 y P2 llegarán al resto de nodos después de los ya enviados por P3 y P4.



36. El algoritmo de Chandy-Lamport devolverá un corte que comprenderá el estado registrado en los eventos "e16", e23", "e33" y "e43"	
37. En el canal (P4, P1) se habrá registrado 1 mensaje	
38. En el canal (P1, P4) se habrán registrado 2 mensajes	
39. En el canal (P4, P2) se habrá registrado 1 mensaje	
40. El mensaje MARCA que envíe el nodo P1 al nodo P4 llegará después de e45 pero antes de e46.	