

**EXAMEN SEGUNDO PARCIAL**

**Unidades Didácticas 7 a 10 - Prácticas 4, 5 y 6**

**Concurrencia y Sistemas Distribuidos**

**Fecha: 5 de Junio de 2017**

Este examen tiene una duración total de 90 minutos.

Este examen tiene una puntuación máxima de **10 puntos**, que equivalen a **3.5 puntos** de la nota final de la asignatura. Consta tanto de preguntas de las unidades didácticas como de las prácticas. Indique, para cada una de las siguientes **58 afirmaciones**, si éstas son verdaderas (V) o falsas (F).

**Cada respuesta vale: correcta= 10/58, errónea= -10/58, vacía=0.**

Importante: Los **primeros 3 errores no penalizarán**, de modo que tendrán una valoración equivalente a la de una respuesta vacía. A partir del 4º error (inclusive), sí se aplicará el decremento por respuesta errónea.

**PREGUNTAS DE TEORÍA**

Sobre las características de los sistemas distribuidos:

1. Cuando un sistema distribuido ofrece transparencia de fallos, los usuarios perciben claramente si un componente del sistema falla.	F
2. Las características de los sistemas distribuidos son acceso homogéneo, disponibilidad, ocultación y escalabilidad.	V
3. Cuando un sistema distribuido oculta en la identificación de los recursos el lugar donde se ubican, decimos que proporciona transparencia de identificación.	F
4. El uso de estándares permite la construcción de sistemas distribuidos formados por módulos, incluso de distintos fabricantes, fácilmente reemplazables o ampliables.	V
5. La escalabilidad administrativa se consigue utilizando técnicas de replicación.	F
6. Para conseguir escalabilidad de distancia es necesario tener en cuenta que al aumentar la distancia entre nodos, los tiempos de respuesta y la fiabilidad empeoran.	V
7. En general, la escalabilidad se ve amenazada cuando se adoptan estrategias descentralizadas.	F

Sobre los mecanismos de comunicación en los sistemas distribuidos:

8. La estructura de los mensajes es la misma en todos los mecanismos de comunicación.	F
9. Existen mecanismos de comunicación que se utilizan haciendo llamadas a procedimientos o métodos remotos.	V
10. En la comunicación asincrónica el middleware responde al emisor del mensaje cuando el receptor ha confirmado la entrega correcta del mensaje.	F

Sobre los mecanismos de comunicación ROI y Java RMI:

11. En ROI, el proxy es el encargado de realizar las verdaderas llamadas al objeto remoto.	F
12. Cuando en ROI se pasa un objeto por referencia se crea un nuevo objeto en el ordenador destino copia del objeto original.	F
13. Cuando se crea un objeto a iniciativa del cliente, éste tiene que registrar primero dicho objeto en el servidor de nombres.	F
14. En Java RMI, si el objeto que se pasa como argumento implementa la interfaz Remote, se serializa y se pasa por valor.	F
15. Java RMI puede considerarse un mecanismo de comunicación en el que se usa direccionamiento directo.	V

Respecto a los servicios web RESTful:

16. Los servicios web RESTful suelen ser servicios sin estado, lo que implica que el cliente debe suministrar toda la información necesaria en cada petición.	V
17. Cuando se siguen las directrices de los servicios web RESTful no importa qué método HTTP utilizar en una petición, puesto que el servidor nunca utiliza dicha información.	F
18. Cuando se siguen las directrices de los servicios web RESTful, las URIs empleadas pueden contener argumentos para, por ejemplo, realizar consultas.	V
19. RESTful se considera normalmente como un mecanismo de comunicación no persistente.	V

Respecto a Java Message Service:

20. JMS es una API Java que permite a las aplicaciones invocar métodos de objetos remotos.	F
21. Los principales componentes de JMS son los proveedores JMS, los clientes JMS, los mensajes y los objetos administrados: factorías de conexiones y destinos.	V
22. En JMS los mensajes carecen completamente de estructura.	F
23. La comunicación en JMS se considera normalmente como asincrónica.	V

Sobre la sincronización de relojes:

24. En el algoritmo de Cristian cada cliente puede sincronizar su reloj con independencia del resto de clientes.	V
25. En el algoritmo de Cristian, para fijar la hora del cliente se calcula el promedio entre el reloj del cliente y el del servidor.	F
26. Con relojes lógicos establecemos un orden causal (causa-efecto), de forma que decimos que el evento "a" precede al evento "b" ( $a \rightarrow b$ ) si todos los nodos observan exactamente dicho orden.	V
27. En los relojes vectoriales, para cualquier par de vectores distintos $V(a)$ $V(b)$ , se cumple que $V(a) < V(b)$ o $V(b) < V(a)$ .	F
28. Si en el algoritmo de Berkeley el reloj del servidor es igual a 10.005 y el de un cliente 10.000, el servidor comunicará a dicho cliente que debe adelantar su reloj en 5 unidades de tiempo, independientemente de los valores de los relojes del resto de clientes.	F
29. Si el reloj lógico de Lamport asociado al evento "a" es igual a 3 y el reloj lógico de Lamport asociado al evento "b" es igual a 5, podemos afirmar que el evento "a" ha ocurrido antes que el evento "b".	F
30. Suponga que el reloj lógico de Lamport de un nodo es igual a 5. Si dicho nodo recibe un mensaje etiquetado con un reloj lógico de Lamport igual a 3, el nuevo valor de su reloj lógico de Lamport será igual a 4, es decir, uno más que el reloj lógico de Lamport del mensaje.	F
31. Siendo $V(a) = [5, 9, 7, 3, 8]$ y $V(b) = [4, 3, 9, 1, 10]$ dos relojes vectoriales, podemos afirmar que los eventos "a" y "b" son concurrentes entre sí.	V
32. Cuando llega un mensaje a un nodo, el reloj vectorial de dicho nodo se actualiza sumando al valor actual el valor del reloj vectorial que se transmite en el mensaje.	F

Sobre el estado global (Algoritmo Chandy-Lamport) y la elección de líder:

33. Chandy-Lamport sería válido para una topología en anillo, pero sólo si se garantiza que los canales son fiables y FIFO.	F
34. En el algoritmo de elección de líder con topología en anillo, el líder es aquel nodo que posee el token.	F
35. Cualquier algoritmo de elección de líder requiere que cada nodo disponga de un identificador único, que además debe ser conocido por los otros nodos.	V
36. Una instantánea consistente se caracteriza porque todos los nodos registran su estado en el mismo instante de tiempo real.	F
37. En el algoritmo de Chandy-Lamport, cuando un nodo "p" recibe el mensaje MARCA por el canal "c", si ya había registrado su estado deja de registrar la actividad en dicho canal "c".	V
38. En el algoritmo Bully, se escoge como líder al primer nodo que envía el mensaje ELECCIÓN al nodo coordinador.	F

Sobre los algoritmos de exclusión mutua en sistemas distribuidos:

39. En el algoritmo de exclusión mutua distribuido, un nodo entra en la sección crítica cuando el resto de nodos le responden con un mensaje OK.	V
40. En el modelo centralizado, el coordinador debe mantener una lista de respuestas pendientes (nodos que han solicitado acceso al recurso, pero a los que todavía no se les ha contestado).	V
41. En el algoritmo de exclusión mutua para anillos, si un nodo que desea entrar a la sección crítica recibe el token, utiliza relojes lógicos (Lamport) para determinar si tiene prioridad sobre otros que también desean entrar.	F

Respecto a las arquitecturas de los sistemas distribuidos:

42. Una arquitectura basada en eventos es un estilo arquitectónico en el que los componentes del software se modelan en base a objetos y llamadas a métodos.	F
43. La organización por niveles se basa en que cada nivel ofrece servicios al nivel inmediatamente inferior y solita servicios al nivel inmediatamente superior.	F
44. Las arquitecturas software indican cómo los componentes lógicos están organizados y cómo deberían interactuar entre sí.	V
45. En el modelo cliente/servidor, los clientes envían peticiones y esperan respuestas mientras que los servidores reciben peticiones, las procesan y contestan.	V

Respecto a los sistemas peer-to-peer, los sistemas Grid y los sistemas Cloud:

46. Los sistemas Grid consisten en un conjunto de recursos que se asocian para ejecutar aplicaciones que requieren gestionar un elevado volumen de información y gran cantidad de cálculos computacionales.	V
47. En los sistemas Cloud, el modelo de Software como Servicio permite la contratación de máquinas virtuales que se ejecutan en las instalaciones del proveedor.	F
48. Cuando clasificamos los sistemas peer-to-peer en base a su grado de centralización, obtenemos tres tipos de arquitecturas: puramente descentralizadas, parcialmente centralizadas y descentralizadas híbridas.	V
49. Los sistemas cloud que permiten desarrollar, desplegar y ejecutar aplicaciones web se conocen como Plataformas como servicio (PaaS)	V

## REGUNTAS DE PRÁCTICAS

Sobre la práctica 4 (Servicios de dominio de Active Directory AD DS):

1. En las actividades de la práctica dos de los ordenadores debían configurarse como controladores del mismo dominio.	V
2. En las actividades de la práctica, se ha creado un bosque con dos árboles: uno cuyo primer dominio raíz es eovic.csd y otro cuyo primer dominio raíz es amsterdam.csd	F
3. En el directorio ADDS, una unidad organizativa es un tipo de contenedor, que puede contener otros elementos del directorio, como usuarios y equipos.	V

Sobre la práctica 5 (Chat distribuido orientado a objetos basado en RMI):

4. Cuando un usuario envía un mensaje al canal, otro usuario del canal obtiene el contenido del mensaje mediante invocación de un método sobre el objeto remoto ChatMessage correspondiente al mensaje.	V
5. En esta práctica, rmiregistry es un componente de java RMI que permite encontrar el objeto servidor de la aplicación chat.	V
6. La interfaz gráfica (ChatUI) es utilizada por ChatRobot para obtener las indicaciones sobre a qué servidor y canal debe conectarse.	F

Sobre la práctica 6 (Java Message Service):

7. El cliente CsdMessengerClient requiere inicializar un contexto JNDI para establecer la conexión con el bróker.	V
8. En esta práctica, la cola "csd" se crea a través del proveedor JMS Apache ActiveMQ Artemis.	V
9. CsdMessengerClient y CsdMessengerServer utilizan la misma cola "csd" para enviarse y recibir mensajes entre sí.	F