

Este examen consta de 15 preguntas con un total de 15 puntos. Tres preguntas incorrectas restan un punto. Sólo una opción es correcta a menos que se indique algo distinto. No está permitido el uso de calculadora. La duración máxima de este examen será de 30 minutos. Sobre la HOJA DE RESPUESTAS:

- Rellene sus datos personales en el formulario superior.
- Indique «Sistemas Distribuidos» en el campo EVALUACIÓN.
- Indique su DNI en la caja lateral (marcando también las celdillas correspondientes).
- Marque la casilla «1» en la caja TIPO DE EXAMEN.

Marque sus respuestas sólo cuando esté completamente seguro. El escáner no admite correcciones ni tachones de ningún tipo, las anulará automáticamente. Debe entregar únicamente la hoja de respuestas.

Apellidos: \_\_\_\_\_ Nombre: \_\_\_\_\_ Grupo: \_\_\_\_\_

- 1 [1p] En un sistema distribuido ¿qué práctica resuelve el problema de la heterogeneidad?
 

<input type="checkbox"/> a) El uso de sistemas abiertos y estándares	<input type="checkbox"/> c) La comunicación indirecta
<input type="checkbox"/> b) El uso de APIs binarias	<input type="checkbox"/> d) El uso de APIs RMI
  
- 2 [1p] ¿Qué mecanismo proporciona transparencia de localización?
 

<input type="checkbox"/> a) URL	<input type="checkbox"/> c) Dirección MAC
<input type="checkbox"/> b) Dirección IP privada/Puerto	<input type="checkbox"/> d) Dirección binaria
  
- 3 [1p] ¿Cuál de los siguientes paradigmas de comunicación se considera invocación remota?
 

<input type="checkbox"/> a) Publicador-Subscriber	<input type="checkbox"/> c) Colas de mensajes
<input type="checkbox"/> b) Petición-Respuesta	<input type="checkbox"/> d) Espacio de tuplas
  
- 4 [1p] ¿En qué se diferencian los patrones arquitecturales de diseño por capas (layering) del diseño por niveles (tiered architecture)?
 

<input type="checkbox"/> a) Ambos son lo mismo.
<input type="checkbox"/> b) El diseño por niveles incluye el software y el hardware con funciones específicas. El diseño por capas se refiere solo al software
<input type="checkbox"/> c) El diseño por capas se refiere sólo al software. El diseño por niveles sólo implica hardware.
<input type="checkbox"/> d) El diseño por niveles es solo para el middleware y el por capas solo para la base de datos
  
- 5 [1p] En un sistema distribuido de llamada por voz, tiene un serio problema de *jitter* que hace que la calidad de la comunicación sea muy mala, ¿Qué está pasando?
 

<input type="checkbox"/> a) Los mensajes de audio llegan con una variación en el tiempo de entrega muy pequeña.
<input type="checkbox"/> b) Los mensajes de audio llegan con un ancho de banda insuficiente.
<input type="checkbox"/> c) Los mensajes de audio llegan con un CRC erróneo.
<input type="checkbox"/> d) Los mensajes de audio llegan con una variación en el tiempo de entrega muy grande.
  
- 6 [1p] Se encuentra diseñando el protocolo binario para dar soporte a un middleware de tipo RMI sobre el protocolo UDP. ¿Qué campo añade para evitar duplicados?
 

<input type="checkbox"/> a) Marca de tiempo	<input type="checkbox"/> c) Identificador del mensaje
<input type="checkbox"/> b) Checksum de los datos del mensaje	<input type="checkbox"/> d) Código de operación
  
- 7 [1p] Se encuentra diseñando el protocolo binario para dar soporte a un middleware de tipo RMI sobre el protocolo UDP. ¿Qué serialización de las estructuras de datos ocupa un menor número de bytes en sus mensajes?
 

<input type="checkbox"/> a) XML	<input type="checkbox"/> b) JSON	<input type="checkbox"/> c) BSON	<input type="checkbox"/> d) YAML
---------------------------------	----------------------------------	----------------------------------	----------------------------------

- 8** [1p] Se encuentra diseñando el protocolo binario para dar soporte a un middleware de tipo RMI sobre el protocolo UDP. Para ayudarle, un compañero le pasa un diagrama de secuencia UML con las invocaciones y argumentos de llamada entre entidades del sistema al cual le debe dar soporte su protocolo, también adjunta un documento tipo RFC explicando dicho diagrama. Este documento le ayuda a especificar la:
- ☐ a) Temporización.
  - ☐ b) Sintaxis
  - ☐ c) Semántica
  - ☐ d) Sintaxis, semántica y temporización
- 9** [1p] ¿Cuál de los siguientes campos añade de forma automática el compilador de *Proccol buffers* a su estructura definida al serializarla?
- ☐ a) CRC
  - ☐ b) Marca de tiempo
  - ☐ c) Identificador del tipo de mensaje
  - ☐ d) No se añade ninguno de los indicados
- 10** [1p] Marque la afirmación FALSA en relación al proceso de «serialización»:
- ☐ a) Todos los componentes de un middleware usan el mismo sistema de serialización.
  - ☐ b) Produce siempre el mismo resultado para los mismos datos de entrada.
  - ☐ c) Depende del lenguaje de implementación de cliente y servidor.
  - ☐ d) Es un proceso reversible.
- 11** [1p] ¿Qué patrón de llamada NO es adecuado para implementar una invocación RMI confiable?
- ☐ a) Request(R)
  - ☐ b) Request-Reply (RR)
  - ☐ c) Request-Reply-Ack (RRA)
  - ☐ d) Ninguno de los anteriores
- 12** [1p] ¿Qué semántica de invocación parece a priori la más adecuada para una señal periódica que indica que un dispositivo sigue activo (*heartbeat*)?
- ☐ a) maybe
  - ☐ b) at-least-once
  - ☐ c) at-most-once
  - ☐ d) exactly-once
- 13** [1p] Un proceso A invoca, mediante un mecanismo RPC, una función en el proceso B en otra máquina con un argumento x. La invocación puede retornar un código de error. En esta comunicación:
- ☐ a) La serialización (marshalling) y la des-serialización (unmarshalling) del argumento x se realiza tanto en A como en B.
  - ☐ b) La serialización del argumento x se realiza en B y la des-serialización se realiza en A.
  - ☐ c) La serialización del valor de retorno se realiza en B y la des-serialización se realiza en A.
  - ☐ d) A es el servidor y B es el cliente.
- 14** [1p] ¿Qué mecanismo requiere el estilo RRA que no necesitan los demás?
- ☐ a) Un hilo por petición para evitar llamadas bloqueantes.
  - ☐ b) Números de secuencia.
  - ☐ c) Registro de invocaciones realizadas (historia).
  - ☐ d) No requiere ningún mecanismo adicional.
- 15** [1p] Cuando se dice que un mecanismo de resolución de servidores RPC es persistente ¿a qué nos estamos refiriendo?
- ☐ a) El *binder* está disponible siempre en la misma IP y puerto.
  - ☐ b) El *binder* almacena las referencias a los servidores RPC en disco.
  - ☐ c) El *binder* devuelve siempre la misma referencia para un servidor dado.
  - ☐ d) El cliente resuelve la referencia en la primera invocación y luego la reutiliza.

