

*Este examen consta de 14 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.*

Apellidos: \_\_\_\_\_ **SOLUCIÓN** \_\_\_\_\_ Nombre: \_\_\_\_\_ Grupo: \_\_\_\_\_

- 1** [1p] ¿Qué tres características tienen la mayoría de los sistemas distribuidos en la actualidad?
- ☒ a) Procesos concurrentes, fallos independientes y ausencia de reloj global.
  - ☐ b) Procesos concurrentes, fiabilidad en las comunicaciones y ausencia de reloj global.
  - ☐ c) Procesos concurrentes, sin latencia en las comunicaciones y ausencia de reloj global.
  - ☐ d) Procesos concurrentes, con seguridad a nivel de red y ausencia de reloj global.
- 2** [1p] ¿Qué aspecto de la definición de un protocolo proporciona la especificación de una interfaz utilizando un lenguaje tipo IDL de un sistema distribuido?
- ☒ a) Sintaxis
  - ☐ b) Sintaxis, semántica y temporización
  - ☐ c) Temporización
  - ☐ d) Semántica
- 3** [1p] Un compañero de trabajo le informa de que hay que introducir seguridad en un sistema distribuido que están implementando ¿Qué aspectos debe tener en cuenta para empezar a diseñar/implementar la seguridad?
- ☒ a) Garantizar la confidencialidad, integridad y disponibilidad.
  - ☐ b) A nivel de sistema distribuido, garantizar la disponibilidad del sistema.
  - ☐ c) El uso de comunicaciones cifradas a nivel de red.
  - ☐ d) Garantizar la confidencialidad en el acceso al sistema distribuido.
- 4** [1p] En su proceso de selección de un middleware de comunicaciones ¿qué debe tener en cuenta?
- ☐ a) El número de sistemas operativos y lenguajes de programación a los que ha sido portado.
  - ☒ b) El grado de soporte a las características no funcionales del sistema distribuido a construir.
  - ☐ c) El número de servicios horizontales (persistencia, eventos, etc.)
  - ☐ d) El modelo de seguridad que implemente.
- 5** [1p] Su jefe le ha pedido que convierta una aplicación centralizada implementada en Java, en una aplicación distribuida. Sin darle mas información ¿Qué modelo le parecería mas adecuado y mas natural para implementar el sistema distribuido?
- ☒ a) RMI
  - ☐ b) RPC
  - ☐ c) Comunicación Indirecta
  - ☐ d) Sockets

E. [1p] Se pretende diseñar un protocolo basado en cliente/servidor para controlar varios LEDs de un edificio de manera remota. Las operaciones posibles sobre un LED son:

- ON: encender LED
- OFF: apagar LED

Seleccione la sintaxis más adecuada:

> **6**

a)

```
Request = | ID PROTO (1B) | ID LED (1B) | ID OPERACION (1B) |
Reply   = | COD. ERROR (1B) | ID LED (1B) |
```

b)

```
Request = | ID PROTO (1B) | ID LED (1B) | PAYLOAD LENGTH (1B) | CRC (2B) |
Reply   = | COD. ERROR (1B) | ID LED (1B) |
```

c)

```
Request = | SERVER IP (4B) | SERVER PORT (2B) | ID PROTO (1B) | ID LED (1B) |
Reply   = | CLIENT IP (4B) | CLIENT PORT (2B) | ERROR CODE (1B) | ID LED (1B) |
```

d)

```
Request = | SERVER IP (4B) | SERVER PORT (2B) | TCP/UDP (1B) | ID PROTO(1B) | ID LED(4B) |
Reply   = | CLIENT IP (4B) | CLIENT PORT (2B) | ERROR CODE (1B) | ID LED (4B) |
```

☒

a)

☐

b)

☐

c)

☐

d)

**7**

[1p] Se pretende diseñar un protocolo extensible que permita añadir nuevas operaciones en el futuro, por ejemplo, obtener/fijar la intensidad de luz en un determinado LED.

¿Qué campo debería añadir a la cabecera para lograrlo?

☐

a) Código de redundancia cíclica (CRC)

☐

c) Longitud del payload

☐

b) Número de secuencia

☒

d) Versión del protocolo

**8**

[1p] ¿En qué consiste la temporización de un protocolo?

☐

a) Establecer límites de tiempo para la transmisión de los mensajes definidos en el protocolo.

☒

b) Definir la secuencia de mensajes intercambiados entre las entidades que se comunican.

☐

c) Incorporar un campo «timestamp» o marca de tiempo a los mensajes para saber cuándo son transmitidos.

☐

d) Crear aplicaciones cliente/servidor síncronas.

E. [2p] Se desea extender el protocolo para añadir dos nuevas operaciones:

- GET(): devuelve la intensidad de luz que emite un LED.
- PUT(value): fija la intensidad de luz que emite un LED con el valor *value*.

> **9** Se decide representar los mensajes de este protocolo utilizando Google Protocol Buffers versión 2. Seleccione la opción más adecuada para representar los mensajes de petición de operación:

a)

```
message LED {
  enum Operation {ON=0; OFF=1; GET=2; PUT=3;
    UNKNOWN=4;}
  required int version = 1;
  required Operation LedOperation = 2;
  required int IdLed = 3;
  optional float value = 4;
}
```

c)

```
message LED {
  enum Operation {ON=0; OFF=1; GET=2; PUT=3;
    UNKNOWN=4;}
  required Operation LedOperation = 1;
  required int IdLed = 2;
  required float value = 3;
}
```

b)

```
message LED {
  enum Operation {ON=0; OFF=1; GET=2; PUT=3;
    UNKNOWN=4;}
  required Operation LedOperation = 1;
  required int IdLed = 2;
  optional float value = 3;
  optional float response = 4;
}
```

d)

```
message LED {
  enum Operation {ON=0; OFF=1; GET=2; PUT=3;
    UNKNOWN=4;}
  required Operation LedOperation = 1;
  required int IdLed = 2;
  optional float value = 3;
}
```

☐ a)

☐ b)

☐ c)

☒ d)

> **10** ¿Qué método utilizará para deserializar un mensaje LED instanciado mediante `myLed = Led()`?

☐ a) `myLed.unpack()`

☐ c) `myLed.SerializeToString()`

☐ b) `myLed.ntohs()/myLed.ntohl()`

☒ d) `myLed.ParseFromString()`

**11** [1p] Se requiere operar remotamente el mecanismo de apertura de una puerta automática. ¿Qué semántica de invocación permitiría conseguir el comportamiento deseado al mínimo coste (memoria, procesamiento y uso de red)?

☐ a) maybe

☐ c) at most once

☒ b) at least once

☐ d) exactly once

**12** [1p] La «computación huérfana» sucede cuando...

☐ a) un cliente no encuentra el servidor adecuado que debe invocar.

☐ b) un servidor RPC no dispone del procedimiento solicitado por el cliente.

☒ c) un servidor continua ejecutando el código de un procedimiento a pesar de que el cliente haya desaparecido.

☐ d) ocurre un fallo en el servidor que le impide devolver el resultado de un procedimiento RPC.

**13** [1p] ¿Para que sirve el programa portmapper?

☒ a) Es un enlazador (*binder*) para localizar servidores RPC.

☐ b) Lista todos los servidores RPC disponibles en la red.

☐ c) Genera los *stubs* a partir de una especificación XDR.

☐ d) Permite averiguar la dirección de los clientes de un servidor RPC concreto.

**14** [1p] ¿Cuál es la relación entre el tiempo de respuesta de una llamada a un procedimiento en el mismo espacio de memoria de un proceso respecto al uso de un mecanismo RPC?

☒ a) La llamada RPC es siempre considerablemente más lenta.

☐ b) Si el destino está en el mismo computador, no hay ninguna diferencia apreciable.

☐ c) Si el destino está en la misma LAN, no hay ninguna diferencia apreciable.

☐ d) Dependiendo de la tecnología RPC, puede variar entre 1 a 1000 veces más rápida.

**15** [1p] ¿Qué tipo de herencia permiten las interfaces remotas que el programador define cuando utiliza un middle-ware RMI?

☒ a) De interfaz

☐ c) De implementación (múltiple)

☐ b) De implementación (única)

☐ d) No suelen soportar herencia

