

## **Comunicación en un grupo confiable**

1. Comunicación punto a punto donde una computadora envía mensajes a otra computadora.  
**Unicast.**
2. Multitransmisión en la cual una computadora envía mensajes a todas las computadoras en una red.  
**Broadcast.**
3. Multitransmisión en la cual una computadora envía mensajes a una o más computadoras en una red.  
**Multicast.**
4. Un sistema que continúa operando con normalidad ante fallas.  
**Transparente ante fallas / Tolerante a fallas.**
5. Si un sistema es tolerante a fallas entonces es un sistema:  
**Fiable.**
6. Capacidad de un sistema de funcionar correctamente siempre:  
**Disponibilidad.**
7. Capacidad de un sistema de funcionar continuamente sin fallar:  
**Confiabilidad.**
8. Propiedad que tiene el sistema de no causar un evento catastrófico cuando falla:  
**Seguridad.**
9. Capacidad que tiene un sistema de ser reparado cuando falla:  
**Mantenimiento.**
10. Falla que se presenta cuando el sistema estaba funcionando normalmente y de pronto se detiene:  
**Falla de congelación.**
11. Falla que se presenta cuando el sistema no recibe los mensajes o no envía los mensajes:  
**Falla de omisión.**
12. Falla que se produce cuando el tiempo de respuesta del sistema es mayor al especificado en los requisitos no funcionales:  
**Falla de tiempo.**

13. Falla que produce una respuesta con un valor incorrecto:  
**Falla de respuesta.**
14. Falla que produce cualquier respuesta, en cualquier momento y con cualquier tiempo de respuesta:  
**Falla arbitraria.**
15. Rango del primer byte en una dirección IP V4 de clase A:  
**1 - 126.**
16. Rango del primer byte en una dirección IP V4 utilizada para multicast:  
**224 - 239.**
17. Número máximo de redes y hosts por red en una dirección IP V4 clase C:  
**2 097 150 redes y 254 hosts.**
18. Establece una conexión virtual uno-a-uno utilizando *handshaking*:  
**Socket stream.**
19. No hay acknowledgement ni re-envío:  
**Socket datagram.**
20. Protocolo utilizado generalmente para implementar comunicación unicast confiable:  
**TCP.**
21. ¿Cómo se ocultan las fallas por omisión en la comunicación unicast confiable implementada mediante TCP?  
**Mediante retransmisión de mensajes.**
22. Fallas que oculta el protocolo TCP:  
**Fallas por omisión.**
23. Garantiza que todos los miembros de un grupo reciben los mensajes transmitidos, sin importar el orden en que se reciben:  
**Multicast confiable.**
24. Implementación de multicast confiable que envía acuse solo cuando no se recibe el mensaje:  
**Acuse negativo.**
25. Garantiza que un mensaje llegue a todos a destinatarios o a ninguno:  
**Multicast atómico.**

26. Crea un datagrama para envío:  
**new DatagramPacket(buffer, buffer.length, grupo, puerto).**
27. Método utilizado por el cliente para recibir los mensajes enviados a un grupo:  
**socket.joinGroup(grupo).**
28. En una aplicación multicast ¿Qué identifica el grupo al cual se envía los mensajes?  
**Una dirección IP de clase D.**

## **Kahoot 2 (RMI)**

1. **Es el modelo natural para el desarrollo de sistemas distribuidos, ya que reproduce la comunicación entre las personas**  
Paso de mensajes
2. **El programador debe serializar los datos antes de enviarlos, y des-serializar los datos después de recibirlos**  
Paso de mensajes
3. **El paso de mensajes es un paradigma:**  
orientado a datos
4. **Encapsula variables y funciones**  
Objeto
5. **El paradigma de objetos distribuidos es:**  
orientado a la acción
6. **Objetos cuyos métodos son invocados por un proceso que se ejecuta en la misma computadora donde reside el objeto.**  
Objeto cuyos métodos son invocados por un proceso que ejecuta en la misma computadora donde reside el objeto  
Objeto local.
7. **Comparten el espacio de direcciones**  
Objetos locales
8. **Objeto cuyos métodos son invocados por procesos que son ejecutados en otra computadora.**  
Objeto remoto
9. **Servidor de nombres en java RMI**

rmiregistry

**10. ¿Qué registra el servidor RMI en el servidor de nombres?**

Una url y una instancia del objeto remoto

**11. Método utilizado por un servidor RMI para registrar un objeto remoto en rmiregistry.**

java.rmi.Naming.bind

**12. Método utilizado por un cliente RMI para obtener una referencia a un objeto remoto previamente registrado en rmiregistry**

java.rmi.Naming.lookup

**13. ¿Cual es el puerto por default utilizado por rmiregistry?**

1099

**14. ¿Qué url es válida para rmiregistry?**

rmi://localhost:1000/nombre

**15. La interface correspondiente al objeto remoto hereda de :**

java.rmi.Remote

**16. La clase correspondiente al objeto remoto hereda de:**

java.rmi.server.UnicastRemoteObject

**17. El constructor default de la clase correspondiente al objeto remoto puede producir la excepción:**

java.rmi.RemoteException

**18. Los métodos exportados del objeto remoto pueden producir:**

java.rmi.RemoteException

**19. ¿En qué computadora se ejecuta rmiregistry?**

debe ejecutar en la misma computadora donde se ejecuta el servidor RMI

**20. ¿En qué orden se debe ejecutar rmiregistry, el cliente RMI y el servidor RMI?**

rmiregistry, servidor, cliente

## JSON

1. Formato texto para el intercambio de datos como objetos serializados.

**JSON**

2. En JSON es posible crear dos estructuras

**Objetos y arreglos**

3. En JSON {“nombre”:”Hugo”,”edad”:20,”sueldo”:1000.0} es

**Un objeto**

4. En JSON [{“nombre”:”Hugo”,”edad”:20,”sueldo”:1000.0}] es

**Un arreglo**

5. En JSON, una colección no ordenada de parejas nombre:valor separadas por comas

**Objeto**

6. En JSON, una colección ordenada de valores separados por comas

**Arreglo**

7. En JSON, cadena de caracteres entre comillas, número, objeto, arreglo, true, false o null

**Valor**

8. En JSON, cero o más caracteres unidos cerrados entre comillas (puede tener secuencias de escape)

**Cadena de caracteres**

9. Si un objeto tiene un campo que es una auto-referencia al mismo objeto ¿Cómo serializa GSON este campo?

**No despliega el campo**

10. Si un objeto tiene un campo que es una referencia a otro objeto ¿Cómo serializa GSON este campo?

**Despliega el nombre del campo, dos puntos y el objeto referenciado completo.**

11. ¿Qué parámetro se debe pasar a la función SimpleDateFormat para utilizar el formato ISO8601 para las fechas?

**“yyyy-MM-dd'T'HH:mm:ss.SSS”**

12. ¿Qué función de GSON se utiliza para serializar un objeto o un arreglo?

**toJson**

13. ¿Qué función de GSON se utiliza para deserializar un string?

**fromJson**

14. En GSON si serializa y deserializa un objeto con auto-referencia ¿Qué valor tendrá el campo que hace la auto-referencia?

**null**

15. Implementación de JSON desarrollada por Google:

**GSON**

## **Servicios web**

1. Un sistema de software diseñado para soportar la interacción interoperable de máquina-a-máquina sobre una red.

**Servicio web**

2. Software o hardware que envía y recibe mensajes que implementa un servicio web.

**Agente**

3. Aquella persona u organización que provee un agente que implementa un determinado servicio.

**Entidad proveedora**

4. Recurso caracterizado por un conjunto abstracto de la funcionalidad que se provee.

**Servicio**

5. Persona u organización que desea hacer uso del servicio.

**Entidad solicitante**

6. Contrato que gobierna el significado y propósito de la interacción con el servicio.

**Semántica del servicio**

7. Contrato que gobierna la mecánica de interacción con un servicio en particular

**Descripción del servicio**

8. Define un protocolo de RPC, para la interacción cliente-servidor a través de la red utilizando HTTP y XML.

**SOAP**

9. Documento XML que contiene la descripción de un servicio web SOAP

**WSDL**

10. Define un conjunto de principios arquitectónicos para la creación de servicios web.

**REST**

11. Método HTTP utilizado por un servicio web estilo REST para crear recursos

**POST**

12. Método HTTP utilizando por un servicio web estilo REST para leer recursos

### **GET**

13. Método HTTP utilizado por un servicio web estilo REST para cambiar el estado o actualizar un recurso

### **PUT**

14. Método HTTP utilizando por un servicio web estilo REST para borrar un recurso

### **DELETE**

15. ¿Qué principio de diseño indica que los servicios web estilo REST deben enviar particiones completas e independientes?

**Los servicios son “sin estado”**