

1. ¿Qué es un protocolo de comunicaciones?

- A) Un lenguaje de programación para redes.
- B) Un conjunto de reglas que permite la comunicación entre dispositivos en una red.
- C) Un tipo de hardware de red.
- D) Un antivirus especializado en redes.

Respuesta: B) Un conjunto de reglas que permite la comunicación entre dispositivos en una red.

2. ¿Cuál de los siguientes es un ejemplo práctico del protocolo TCP?

- A) Enviar un correo electrónico.
- B) Ver un video en YouTube.
- C) Hacer una videollamada en WhatsApp.
- D) Escuchar música en Spotify.

Respuesta: A) Enviar un correo electrónico.

3. ¿Qué función cumple la dirección IP en un protocolo de comunicación?

- A) Identifica de manera única a cada dispositivo en una red.
- B) Aumenta la velocidad de transmisión de datos.
- C) Sirve para cifrar la información enviada.
- D) Establece conexiones entre servidores de streaming.

Respuesta: A) Identifica de manera única a cada dispositivo en una red.

4. ¿Cuál de los siguientes códigos en Java representa el envío de datos usando TCP?

A)

```
java                                                                    Copy Edit
DatagramSocket socket = new DatagramSocket();
byte[] buffer = "¡Hola!".getBytes();
DatagramPacket paquete = new DatagramPacket(buffer, buffer.length, InetAddress.getByName("127.0.0.1"));
socket.send(paquete);
socket.close();
```

B)

java

```
Socket socket = new Socket("127.0.0.1", 8080);  
OutputStream output = socket.getOutputStream();  
PrintWriter writer = new PrintWriter(output, true);  
writer.println("¡Hola desde el cliente TCP!");  
socket.close();
```

C)

java

```
System.out.println("Mensaje enviado por Bluetooth");
```

D)

java

```
ServidorSocket servidor = new ServidorSocket();  
servidor.enviarDatos("Hola");
```

Respuesta: B) Código que usa Socket y PrintWriter.

5. ¿Qué protocolo se usa en una videollamada para minimizar retrasos?

- A) TCP
- B) UDP
- C) IPX
- D) FTP

Respuesta: B) UDP

6. ¿Cuál es la diferencia principal entre TCP y UDP?

- A) TCP es más rápido y UDP es más confiable.
- B) TCP garantiza la entrega de datos y UDP no.
- C) TCP no establece conexión antes de enviar datos, mientras que UDP sí.
- D) TCP solo funciona en redes privadas, mientras que UDP es para Internet.

Respuesta: B) TCP garantiza la entrega de datos y UDP no.

7. ¿Qué característica tiene el protocolo UDP en comparación con TCP?

- A) Es más lento pero más seguro.
- B) No garantiza la entrega de datos.
- C) Usa direcciones IP dinámicas.
- D) Es obligatorio para el envío de correos electrónicos.

Respuesta: B) No garantiza la entrega de datos.

8. ¿En qué tipo de aplicaciones es más conveniente usar TCP en lugar de UDP?

- A) Streaming de video en tiempo real.
- B) Juegos en línea.
- C) Transferencia de archivos grandes.
- D) Emisión de radio en vivo.

Respuesta: C) Transferencia de archivos grandes.

9. ¿Cuál es el proceso de establecimiento de conexión en TCP?

- A) Handshake de dos pasos.
- B) Three-Way Handshake.
- C) Transmisión sincrónica.
- D) Modo Half-Duplex.

Respuesta: B) Three-Way Handshake.

10. ¿Qué protocolo usarías si necesitas asegurarte de que cada byte de información llegue correctamente al destinatario?

- A) TCP
- B) UDP
- C) HTTP
- D) ARP

Respuesta: A) TCP

11. ¿Por qué UDP es más rápido que TCP?

- A) Porque usa direcciones IPv6 en lugar de IPv4.
- B) Porque no requiere establecer conexión antes de enviar datos.
- C) Porque cifra los datos antes de enviarlos.
- D) Porque siempre usa paquetes más pequeños.

Respuesta: B) Porque no requiere establecer conexión antes de enviar datos.

12. ¿Cuál de los siguientes servicios usa TCP en lugar de UDP?

- A) Streaming de video.
- B) Juegos en línea.
- C) Transferencia de archivos con FTP.
- D) Radio por Internet.

Respuesta: C) Transferencia de archivos con FTP.

13. ¿Cuál de estas afirmaciones sobre direcciones IP es correcta?

- A) IPv6 es más rápido que IPv4.
- B) IPv6 permite conectar más dispositivos que IPv4.
- C) IPv4 es más seguro que IPv6.
- D) IPv4 y IPv6 no pueden coexistir en una red.

Respuesta: B) IPv6 permite conectar más dispositivos que IPv4.

14. En el siguiente código, ¿qué tipo de protocolo de comunicación se está usando?

```
java                                                                    Copy Edit
DatagramSocket socket = new DatagramSocket();
String mensaje = "¡Hola!";
byte[] buffer = mensaje.getBytes();
InetAddress direccion = InetAddress.getByName("127.0.0.1");
DatagramPacket paquete = new DatagramPacket(buffer, buffer.length, direccion, 8080);
socket.send(paquete);
socket.close();
```

- A) TCP
- B) UDP
- C) FTP
- D) HTTP

Respuesta: B) UDP

15. Si estás programando una aplicación que transmite datos en tiempo real, como una llamada de voz, ¿qué protocolo deberías usar y por qué?

- A) TCP, porque asegura que los datos lleguen en orden.
- B) TCP, porque tiene menor latencia que UDP.
- C) UDP, porque prioriza la velocidad sobre la confiabilidad.
- D) UDP, porque cifra los datos de forma automática.

Respuesta: C) UDP, porque prioriza la velocidad sobre la confiabilidad.

UD3 - 02.- Comunicación entre aplicaciones

Preguntas Tipo Test

1. Concepto de Comunicación entre Aplicaciones

Pregunta: ¿Qué opción describe mejor la comunicación entre aplicaciones?

- a) Solo se da entre dos aplicaciones en un mismo ordenador.
- b) Permite el intercambio de información entre aplicaciones a través de una red.
- c) Solo funciona en redes locales.
- d) Solo permite que los servidores envíen datos a los clientes.

✓ **Respuesta correcta: b)** Permite el intercambio de información entre aplicaciones a través de una red.

2. Modelo Cliente/Servidor - Definición

Pregunta: ¿Cuál es la función principal de un cliente en el modelo cliente/servidor?

- a) Proveer servicios a los servidores.
- b) Administrar la conexión entre otros clientes.
- c) Solicitar recursos o servicios al servidor.
- d) Funcionar como un nodo central en la red.

✓ **Respuesta correcta: c)** Solicitar recursos o servicios al servidor.

3. Modelo Cliente/Servidor - Características

Pregunta: ¿Cuál de las siguientes es una característica del modelo cliente/servidor?

- a) No requiere servidores, solo clientes.
- b) Depende de un servidor centralizado para gestionar recursos.
- c) Todos los nodos tienen los mismos privilegios y responsabilidades.
- d) No necesita conexión a una red para funcionar.

✓ **Respuesta correcta: b)** Depende de un servidor centralizado para gestionar recursos.

4. Modelo Cliente/Servidor - Ejemplo Práctico

Pregunta: En el código del servidor TCP en Java, ¿qué instrucción permite al servidor esperar una conexión de un cliente?

- a) new Socket()
- b) server.accept()
- c) server.send()
- d) server.readLine()

✓ **Respuesta correcta: b)** server.accept()

5. Modelo Cliente/Servidor - Ejemplo Práctico

Pregunta: ¿Qué hace la línea Socket socket = new Socket("127.0.0.1", 8080); en el código del cliente TCP?

- a) Cierra la conexión con el servidor.
- b) Crea un nuevo servidor en el puerto 8080.
- c) Establece una conexión con el servidor en la dirección 127.0.0.1 y el puerto 8080.
- d) Escucha conexiones entrantes.

✓ **Respuesta correcta: c)** Establece una conexión con el servidor en la dirección 127.0.0.1 y el puerto 8080.

6. Modelo P2P - Definición

Pregunta: ¿Qué caracteriza al modelo P2P (peer-to-peer)?

- a) Todos los nodos tienen los mismos derechos y pueden actuar como clientes y servidores.
- b) Depende de un servidor central para gestionar la comunicación.
- c) No permite la transferencia de archivos.
- d) Solo funciona en redes privadas.

✓ **Respuesta correcta: a)** Todos los nodos tienen los mismos derechos y pueden actuar como clientes y servidores.

7. Modelo P2P - Características

Pregunta: ¿Cuál es una ventaja del modelo P2P?

- a) Los clientes dependen completamente de un servidor.
- b) Es más vulnerable a fallos en el servidor.
- c) Distribuye la carga entre todos los nodos, evitando la dependencia de un solo servidor.
- d) No puede escalar más allá de dos usuarios.

✓ **Respuesta correcta: c)** Distribuye la carga entre todos los nodos, evitando la dependencia de un solo servidor.

8. Modelo P2P - Ejemplo Práctico

Pregunta: En el código del nodo P2P en Java, ¿qué función permite a un nodo recibir mensajes de otro nodo?

- a) socket.getOutputStream()
- b) new ServerSocket(9090)
- c) socket.close()
- d) socket.sendMessage()

✓ **Respuesta correcta: b)** new ServerSocket(9090)

9. Modelo P2P - Ejemplo Práctico

Pregunta: En el código del nodo P2P en Java, ¿qué hace la siguiente línea de código?

```
java

PrintWriter escritor = new PrintWriter(output, true);
escritor.println(mensaje);
```

- a) Recibe un mensaje de otro nodo.
- b) Envía un mensaje a otro nodo a través del socket.
- c) Cierra la conexión con el servidor.
- d) Inicia un nuevo servidor P2P.

✓ **Respuesta correcta: b)** Envía un mensaje a otro nodo a través del socket.

10. Modelos Híbridos - Definición

Pregunta: ¿Cuál de las siguientes opciones describe un modelo híbrido de comunicación entre aplicaciones?

- a) Utiliza solo la arquitectura cliente/servidor sin variaciones.
- b) Utiliza solo la arquitectura P2P sin servidores centrales.
- c) Combinación de cliente/servidor y P2P, donde un servidor gestiona la red, pero los nodos pueden intercambiar datos directamente.
- d) No permite la descentralización.

✓ **Respuesta correcta: c)** Combinación de cliente/servidor y P2P, donde un servidor gestiona la red, pero los nodos pueden intercambiar datos directamente.

11. Modelos Híbridos - Características

Pregunta: ¿Qué ventaja tiene un modelo híbrido respecto a un modelo P2P puro?

- a) Mayor descentralización sin necesidad de servidores.
- b) Permite una estructura organizada con la ayuda de un servidor central para la gestión inicial.

- c) No requiere conexión a internet.
- d) Solo permite la comunicación entre dos nodos.

✓ **Respuesta correcta: b)** Permite una estructura organizada con la ayuda de un servidor central para la gestión inicial.

12. Modelos Híbridos - Ejemplo Práctico

Pregunta: En el código del Servidor Central en Java, ¿qué hace la instrucción `servidor.accept();`?

- a) Cierra el servidor.
- b) Acepta una conexión entrante de un nodo.
- c) Envía un mensaje a los clientes conectados.
- d) Crea un nuevo socket para el cliente.

✓ **Respuesta correcta: b)** Acepta una conexión entrante de un nodo.

13. Modelos Híbridos - Ejemplo Práctico

Pregunta: En el modelo híbrido, ¿qué rol juega el servidor central?

- a) Gestiona toda la comunicación entre nodos, sin permitir conexiones directas.
- b) Solo almacena los datos sin permitir conexión entre nodos.
- c) Ayuda a conectar nodos inicialmente, pero luego permite la comunicación directa entre ellos.
- d) No interviene en la comunicación en absoluto.

✓ **Respuesta correcta: c)** Ayuda a conectar nodos inicialmente, pero luego permite la comunicación directa entre ellos.

14. Comparación de Modelos

Pregunta: ¿Cuál de los siguientes modelos es más adecuado para descargar archivos en una red descentralizada?

- a) Cliente/servidor.
- b) P2P.
- c) Híbrido.
- d) Ninguno de los anteriores.

✓ **Respuesta correcta: b)** P2P.

15. Comparación de Modelos

Pregunta: ¿En qué modelo la caída del servidor central puede afectar a toda la red?

- a) Cliente/servidor.
- b) P2P.
- c) Híbrido.
- d) Ninguno de los anteriores.

✓ **Respuesta correcta:** a) Cliente/servidor.

UD3 - 03.- Roles Cliente y Servidor

1. ¿Cuál de las siguientes opciones describe mejor el rol de un cliente en una comunicación en red?

- A) Espera solicitudes de los usuarios y responde con servicios.
- B) Inicia la comunicación solicitando servicios o recursos.
- C) Proporciona acceso a múltiples clientes de manera simultánea.
- D) Mantiene una conexión permanente con el servidor sin interrupción.

✓ **Respuesta:** B) Inicia la comunicación solicitando servicios o recursos.

2. ¿Cuál de estas aplicaciones es un ejemplo de cliente en un entorno cliente-servidor?

- A) Apache
- B) MySQL
- C) Google Chrome
- D) PostgreSQL

✓ **Respuesta:** C) Google Chrome

3. ¿Qué función cumple un servidor en una arquitectura cliente-servidor?

- A) Enviar constantemente datos sin recibir solicitudes.
- B) Esperar solicitudes y responder con servicios o recursos.
- C) Cerrar las conexiones de los clientes una vez recibida la solicitud.
- D) Controlar la interfaz de usuario del cliente.

✓ **Respuesta:** B) Esperar solicitudes y responder con servicios o recursos.

4. En el siguiente código, ¿qué sucede cuando el cliente envía el mensaje?+

```
java
PrintWriter escritor = new PrintWriter(output, true);
escritor.println("Hola, servidor!");
```

- A) El servidor ignora el mensaje.
- B) El servidor responde con "Hola, cliente. Mensaje recibido: Hola, servidor!"

- C) La conexión se cierra automáticamente antes de que el servidor pueda responder.
- D) El servidor espera que el cliente envíe otro mensaje antes de responder.

✓ **Respuesta:** B) El servidor responde con "Hola, cliente. Mensaje recibido: Hola, servidor!"

5. ¿Cuál de estas características describe mejor a un servidor?

- A) Es proactivo e inicia las conexiones con los clientes.
- B) Espera conexiones y responde a solicitudes.
- C) Funciona solo cuando un cliente se conecta.
- D) No puede manejar múltiples conexiones simultáneamente.

✓ **Respuesta:** B) Espera conexiones y responde a solicitudes.

6. ¿Cuál de las siguientes afirmaciones sobre la comunicación cliente-servidor es correcta?

- A) La conexión debe permanecer abierta indefinidamente.
- B) Un cliente puede iniciar la comunicación sin que el servidor esté activo.
- C) Un servidor puede manejar múltiples clientes al mismo tiempo.
- D) Un servidor solo puede manejar un cliente a la vez.

✓ **Respuesta:** C) Un servidor puede manejar múltiples clientes al mismo tiempo.

7. ¿Qué ocurrirá si ejecutamos primero el cliente antes que el servidor en el código del ejemplo?

- A) El cliente se conectará exitosamente.
- B) El cliente generará un error porque no encontrará el servidor.
- C) El cliente se conectará, pero no recibirá respuesta.
- D) El cliente se ejecutará sin mostrar ninguna salida.

✓ **Respuesta:** B) El cliente generará un error porque no encontrará el servidor.

8. ¿Cuál es la dirección IP utilizada en el código de ejemplo para conectarse al servidor?

- A) 192.168.1.1
- B) 127.0.0.1
- C) 8.8.8.8
- D) 255.255.255.255

✓ **Respuesta:** B) 127.0.0.1

9. En una arquitectura cliente-servidor, ¿qué protocolo permite una comunicación confiable entre ambos?

- A) UDP
- B) ICMP
- C) TCP
- D) HTTP

✓ **Respuesta:** C) TCP

10. ¿Cuál es el propósito del siguiente fragmento de código en el servidor?

```
java  
  
Socket socket = servidor.accept();
```

- A) Cerrar la conexión con el cliente.
- B) Iniciar la comunicación con otro servidor.
- C) Aceptar la conexión de un cliente.
- D) Enviar un mensaje al cliente automáticamente.

✓ **Respuesta:** C) Aceptar la conexión de un cliente.

11. ¿Qué sucedería si no cerramos el socket en el cliente después de recibir la respuesta del servidor?

- A) La comunicación continuará indefinidamente.
- B) El cliente se cerrará automáticamente.
- C) Se generará un error de conexión en el servidor.
- D) El servidor rechazará futuras conexiones de otros clientes.

✓ **Respuesta:** A) La comunicación continuará indefinidamente.

12. ¿Qué ventaja tiene utilizar multihilos en un servidor?

- A) Permite procesar múltiples conexiones de clientes al mismo tiempo.
- B) Asegura que el servidor solo maneje una conexión a la vez.
- C) Reduce la carga de procesamiento en el servidor.
- D) Obliga al cliente a esperar antes de enviar datos.

✓ **Respuesta:** A) Permite procesar múltiples conexiones de clientes al mismo tiempo.

13. ¿Qué ocurre si el puerto 8080 ya está en uso cuando el servidor intenta iniciarse?

- A) El servidor se ejecutará sin problemas en otro puerto automáticamente.
- B) El servidor generará un error de "puerto en uso".
- C) El servidor se ejecutará, pero ignorará las conexiones entrantes.
- D) El servidor utilizará el puerto 80 de forma predeterminada.

☒ **Respuesta:** B) El servidor generará un error de "puerto en uso".

14. ¿Por qué es importante cerrar la conexión después del intercambio de datos?

- A) Para evitar que el servidor se sobrecargue con conexiones innecesarias.
- B) Para asegurar que el servidor se apague correctamente.
- C) Para obligar al cliente a reiniciar la aplicación.
- D) Para evitar que el cliente envíe más datos.

☒ **Respuesta:** A) Para evitar que el servidor se sobrecargue con conexiones innecesarias.

15. Si quisiéramos que el servidor maneje múltiples clientes al mismo tiempo, ¿qué cambio deberíamos hacer en el código?

- A) Agregar una nueva clase ClienteTCP.
- B) Usar un bucle infinito para aceptar múltiples conexiones.
- C) Usar programación multihilo en el servidor.
- D) Cambiar el puerto del servidor en cada conexión.

☒ **Respuesta:** C) Usar programación multihilo en el servidor.

UD3 - 04.- Elementos para la programación de aplicaciones en red

1. ¿Cuál es la función de la clase ServerSocket en Java?

- A) Gestionar la comunicación entre múltiples clientes.
- B) Crear y gestionar conexiones de servidor en red.
- C) Enviar datos a un servidor desde un cliente.
- D) Analizar paquetes de red recibidos.

Respuesta: B

2. ¿Cuál de las siguientes opciones representa un socket orientado a conexión en Java?

- A) `ServerSocket servidor = new ServerSocket(8080);`
- B) `DatagramSocket socket = new DatagramSocket(8080);`
- C) `Socket cliente = new Socket(8080);`
- D) `URLConnection conexion = (URLConnection) url.openConnection();`

Respuesta: A

3. ¿Qué librería en Java se usa para manejar flujos de entrada y salida en aplicaciones de red?

- A) `java.net`
- B) `java.io`
- C) `java.nio`
- D) `java.security`

Respuesta: B

4. ¿Qué salida se espera al ejecutar el siguiente código de cliente en Java?

```
java

Socket cliente = new Socket("127.0.0.1", 8080);
OutputStream salida = cliente.getOutputStream();
salida.write("Hola, servidor".getBytes());
cliente.close();
```

- A) "Hola, servidor" se envía correctamente al servidor.
- B) Error, falta importar `java.net.Socket`.
- C) El cliente no podrá conectarse sin una dirección IPv6.
- D) El mensaje se pierde debido a que el socket se cierra antes de enviar datos.

Respuesta: A

5. ¿Cuál es la principal diferencia entre TCP y UDP?

- A) TCP es más rápido pero menos confiable que UDP.
- B) UDP no garantiza la entrega de datos, mientras que TCP sí.
- C) TCP y UDP siempre requieren una conexión establecida.
- D) UDP usa más recursos del sistema que TCP.

Respuesta: B

6. ¿Qué hace el siguiente fragmento de código en un servidor Java?

java

```
ServerSocket servidor = new ServerSocket(8080);  
Socket cliente = servidor.accept();
```

- A) Crea un socket cliente en el puerto 8080.
- B) Espera una conexión de un cliente en el puerto 8080.
- C) Envía datos a un cliente en el puerto 8080.
- D) Cierra todas las conexiones activas en el servidor.

Respuesta: B

7. ¿Cuál de las siguientes opciones es una API para enviar correos electrónicos en Java?

- A) JavaMail
- B) java.io MailSender
- C) SMTPHandler
- D) NetworkEmailSender

Respuesta: A

8. ¿Qué tipo de socket usarías para transmitir video en tiempo real?

- A) TCP
- B) UDP
- C) HTTP
- D) FTP

Respuesta: B

9. ¿Cuál de las siguientes afirmaciones sobre OutputStream es correcta?

- A) Se usa para leer datos de un socket.
- B) Permite enviar datos a través de un socket.
- C) Solo funciona con protocolos UDP.
- D) Especifica la dirección IP de un servidor.

Respuesta: B

10. ¿Qué ocurre si ejecutas el siguiente código sin un servidor activo en el puerto 8080?

java

```
Socket cliente = new Socket("127.0.0.1", 8080);
```

- A) La conexión se establece exitosamente.
- B) Se lanza una excepción de conexión rechazada.
- C) Se reintenta la conexión automáticamente.
- D) Se cierra la aplicación sin error.

Respuesta: B

11. ¿Qué salida imprimirá el siguiente código si un cliente se conecta y envía "Hola"?

java

```
InputStream entrada = cliente.getInputStream();  
byte[] buffer = new byte[1024];  
int bytesLeidos = entrada.read(buffer);  
System.out.println("Mensaje recibido: " + new String(buffer, 0, bytesLeidos));
```

- A) Mensaje recibido: Hola
- B) Mensaje recibido: null
- C) Mensaje recibido: (vacío)
- D) Error de ejecución

Respuesta: A

12. ¿Cómo se establece una comunicación bidireccional entre cliente y servidor?

- A) Usando solo `OutputStream` en el servidor.
- B) Usando `InputStream` en el cliente y `OutputStream` en el servidor.
- C) Configurando solo el servidor sin necesidad de código en el cliente.
- D) Definiendo un puerto dinámico en cada mensaje.

Respuesta: B

13. ¿Qué función tiene la clase `java.net.DatagramSocket`?

- A) Gestionar conexiones TCP.
- B) Enviar y recibir paquetes de datos sin conexión (UDP).
- C) Crear servidores HTTP.
- D) Comprimir datos antes de enviarlos por la red.

Respuesta: B

14. ¿Cuál es la principal ventaja de usar API en la programación de redes?

- A) Permite a las aplicaciones interactuar con servicios de red sin conocer sus detalles internos.
- B) Obliga a los desarrolladores a implementar todo desde cero.
- C) Solo funciona en entornos Windows.
- D) No permite el uso de sockets en Java.

Respuesta: A

15. ¿Cuál es un posible caso de uso de sockets en red?

- A) Un juego en línea que envía y recibe datos en tiempo real.
- B) Un documento de texto almacenado localmente.
- C) Una función matemática que solo usa memoria interna.
- D) Una aplicación de diseño gráfico sin conexión a internet.

Respuesta: A

UD3 - 05.- Zócalos (Sockets)

1. ¿Qué es un socket en el contexto de la programación en red?

- a) Un tipo de cable de conexión física
- b) Un punto final para la comunicación entre dos nodos en una red
- c) Un dispositivo de hardware que mejora la velocidad de internet
- d) Una librería exclusiva de Java para conexiones seguras

Respuesta: b) Un punto final para la comunicación entre dos nodos en una red

2. ¿Cuál de los siguientes protocolos utiliza sockets orientados a conexión?

- a) UDP
- b) HTTP
- c) TCP
- d) FTP

Respuesta: c) TCP

3. ¿Qué clase de Java se usa para crear un socket en el lado del servidor?

- a) ClientSocket
- b) NetworkSocket

- c) ServerSocket
- d) SocketServer

Respuesta: c) ServerSocket

4. ¿Cuál de los siguientes fragmentos de código crea correctamente un socket en el lado del cliente?

- a) Socket socket = new Socket("127.0.0.1", 8080);
- b) ServerSocket socket = new ServerSocket(8080);
- c) Socket socket = new ServerSocket(8080);
- d) ServerSocket socket = new Socket(8080);

Respuesta: a) Socket socket = new Socket("127.0.0.1", 8080);

5. ¿Qué método se usa en ServerSocket para aceptar una conexión entrante?

- a) connect()
- b) listen()
- c) accept()
- d) bind()

Respuesta: c) accept()

6. ¿Qué sucede si el puerto que intenta usar un servidor ya está en uso?

- a) La conexión se establece, pero el servidor no responde
- b) Se lanza una BindException
- c) Se crea otro puerto automáticamente
- d) No afecta al funcionamiento del servidor

Respuesta: b) Se lanza una BindException

7. ¿Cuál es la diferencia principal entre TCP y UDP?

- a) TCP es más rápido que UDP
- b) UDP garantiza la entrega de datos, mientras que TCP no
- c) TCP es confiable y orientado a conexión, mientras que UDP es más rápido pero no confiable
- d) UDP requiere autenticación, mientras que TCP no

Respuesta: c) TCP es confiable y orientado a conexión, mientras que UDP es más rápido pero no confiable

8. ¿Cómo se leen datos enviados desde un socket en Java?

- a) Con `BufferedReader.readLine()`
- b) Con `socket.getInputStream().read()`
- c) Con `socket.readData()`
- d) Con `serverSocket.listenData()`

Respuesta: b) Con `socket.getInputStream().read()`

9. ¿Cuál de los siguientes fragmentos de código envía datos a través de un socket en Java?

- a) `OutputStream salida = socket.getOutputStream(); salida.write("Mensaje".getBytes());`
- b) `socket.send("Mensaje");`
- c) `socket.stream().write("Mensaje");`
- d) `socket.output().writeBytes("Mensaje");`

Respuesta: a) `OutputStream salida = socket.getOutputStream(); salida.write("Mensaje".getBytes());`

10. ¿Qué ventaja tiene el uso de sockets asíncronos sobre los sincrónicos?

- a) Permiten procesar múltiples conexiones sin bloquear la ejecución
- b) Son más seguros que los sincrónicos
- c) No necesitan un puerto específico
- d) No requieren conexión a internet

Respuesta: a) Permiten procesar múltiples conexiones sin bloquear la ejecución

11. ¿Cuál de las siguientes aplicaciones usaría más probablemente sockets UDP en lugar de TCP?

- a) Un sistema de mensajería instantánea
- b) Un servicio de streaming de video en vivo
- c) Un sistema de banca en línea
- d) Un servidor de correo electrónico

Respuesta: b) Un servicio de streaming de video en vivo

12. ¿Cuál es el propósito del método close() en sockets?

- a) Establecer una nueva conexión
- b) Finalizar la conexión y liberar los recursos del socket
- c) Volver a escuchar en el mismo puerto
- d) Cambiar la dirección IP del socket

Respuesta: b) Finalizar la conexión y liberar los recursos del socket

13. ¿Qué sucede si un socket cliente intenta conectarse a un servidor que no está en ejecución?

- a) Se establece la conexión, pero no se envían datos
- b) Se lanza una IOException
- c) Se conecta a otro servidor disponible en la red
- d) El cliente entra en un bucle infinito

Respuesta: b) Se lanza una IOException

14. ¿Qué hace el siguiente código en Java?

```
java  
  
ServerSocket serverSocket = new ServerSocket(9090);  
Socket socket = serverSocket.accept();
```

- a) Crea un servidor que escucha en el puerto 9090 y espera una conexión
- b) Crea un cliente que se conecta al puerto 9090
- c) Enlaza el socket a cualquier puerto disponible automáticamente
- d) Cierra la conexión inmediatamente después de abrirla

Respuesta: a) Crea un servidor que escucha en el puerto 9090 y espera una conexión

15. ¿Cuál de las siguientes opciones es una ventaja del uso de sockets para la comunicación en red?

- a) Requieren menos código que otras soluciones como HTTP
- b) Permiten comunicación directa entre aplicaciones sin necesidad de intermediarios
- c) No necesitan manejo de errores, ya que los datos siempre llegan correctamente
- d) Solo funcionan con dispositivos conectados a la misma red local

Respuesta: b) Permiten comunicación directa entre aplicaciones sin necesidad de intermediarios