

# UD3 - 01.- Protocolos de comunicación

## Contenidos:

1. Concepto de protocolo de comunicaciones.
2. Protocolo IP (Internet Protocol):
  - Definición.
  - Funciones principales.
  - Direcciones IP.
3. Protocolo TCP (Transmission Control Protocol):
  - Definición.
  - Características principales.
  - Funcionamiento orientado a la conexión.
4. Protocolo UDP (User Datagram Protocol):
  - Definición.
  - Características principales.
  - Diferencias con TCP.
5. Comparativa entre TCP y UDP.

---

## Desarrollo de Contenidos

### 1. Concepto de Protocolo de Comunicaciones

Un protocolo de comunicaciones es un conjunto de reglas y procedimientos que permiten a dos o más dispositivos comunicarse entre sí en una red. Piensa en un protocolo como un idioma común que los dispositivos entienden y utilizan para enviar y recibir datos. Sin estos protocolos, los dispositivos no podrían comunicarse correctamente.

Ejemplo sencillo: Cuando envías una carta, sigues un protocolo básico:

1. Escribes el mensaje.
2. Pones la dirección del destinatario en el sobre.
3. Envías la carta.

De manera similar, los protocolos de red garantizan que los datos viajen de forma ordenada y lleguen correctamente a su destino.

---

### 2. Protocolo IP (Internet Protocol)

#### Definición

El protocolo IP es la base de las comunicaciones en Internet. Se encarga de enrutar (enviar) los datos desde un dispositivo de origen hasta el destino correcto, utilizando direcciones únicas llamadas direcciones IP.

### Funciones principales del IP

- **Direccionamiento:** Cada dispositivo conectado a una red tiene una dirección IP única (como la dirección de tu casa). Esto permite identificar a cada dispositivo en la red.
- **Enrutamiento:** IP decide por dónde viajarán los datos para llegar a su destino. Piensa en esto como un GPS que encuentra el mejor camino.

### Direcciones IP

Existen dos versiones principales de direcciones IP:

1. **IPv4:** Formato clásico (ejemplo: 192.168.1.1). Tiene un rango limitado de direcciones, por eso se introdujo IPv6.
2. **IPv6:** Formato más moderno y extenso (ejemplo: 2001:0db8:85a3:0000:0000:8a2e:0370:7334). Soporta más dispositivos conectados.

---

## 3. Protocolo TCP (Transmission Control Protocol)

### Definición

El protocolo TCP es uno de los protocolos más utilizados en Internet. Es orientado a la conexión, lo que significa que establece una conexión entre el dispositivo de origen y el de destino antes de enviar los datos. Esto garantiza que los datos lleguen completos y en el orden correcto.

### Características principales

- **Confiabilidad:** TCP asegura que los datos lleguen correctamente. Si un paquete de datos se pierde, se vuelve a enviar.
- **Orden:** Los datos se entregan en el orden en que fueron enviados.
- **Control de flujo:** TCP regula la cantidad de datos enviados para evitar sobrecargar al receptor.

### Funcionamiento

1. **Establecimiento de conexión:** Se realiza un proceso llamado Three-Way Handshake (apretón de manos en tres pasos).
  - El dispositivo A envía un mensaje al dispositivo B indicando que quiere conectarse.
  - El dispositivo B responde confirmando la solicitud.
  - El dispositivo A confirma que recibió la respuesta y la conexión se establece.
2. **Ejemplo práctico:** Cuando abres una página web, tu navegador establece una conexión TCP con el servidor antes de descargar la información.
3. **Transmisión de datos:** Una vez establecida la conexión, los datos se dividen en paquetes y se envían al destino.
4. **Finalización de la conexión:** Cuando termina la comunicación, se cierra la conexión para liberar recursos.

#### 4. Protocolo UDP (User Datagram Protocol)

##### Definición

El protocolo UDP es un protocolo no orientado a la conexión. Es más rápido que TCP, pero no garantiza que los datos lleguen ni que lo hagan en el orden correcto. Se utiliza en situaciones donde la velocidad es más importante que la confiabilidad.

##### Características principales

- Sin conexión: No hay un proceso de establecimiento de conexión.
- Bajo retraso: Ideal para aplicaciones donde el tiempo es crítico.
- Menor sobrecarga: Como no hay control de errores, UDP tiene menos reglas que TCP, lo que lo hace más rápido.

##### Funcionamiento

1. El dispositivo A envía los datos directamente al dispositivo B sin verificar si está listo para recibirlos.
2. Si un paquete de datos se pierde, no se reenvía.

##### Ejemplo práctico:

- Streaming de video y audio: Aplicaciones como YouTube o Spotify usan UDP para transmitir datos rápidamente.
- Videollamadas: En una videollamada, perder un pequeño paquete de datos es preferible a tener retrasos.

#### 5. Comparativa entre TCP y UDP

Característica	TCP	UDP
Orientación	Orientado a conexión	No orientado a conexión
Confiabilidad	Garantiza la entrega de datos	No garantiza la entrega
Velocidad	Más lento debido a la confiabilidad	Más rápido, pero menos confiable
Aplicaciones	Transferencia de archivos, correos electrónicos	Streaming, videollamadas

## Ejemplo Práctico 1: Uso de TCP para Enviar Datos

```
import java.io.OutputStream;
import java.io.PrintWriter;
import java.net.Socket;

public class ClienteTCP {
    public static void main(String[] args) {
        try {
            // Establecer conexión con el servidor
            Socket socket = new Socket("127.0.0.1", 8080);

            // Enviar datos al servidor
            OutputStream output = socket.getOutputStream();
            PrintWriter writer = new PrintWriter(output, true);
            writer.println("¡Hola desde el cliente TCP!");

            // Cerrar la conexión
            socket.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

---

## Ejemplo Práctico 2: Uso de UDP para Enviar Datos

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;

public class ClienteUDP {
    public static void main(String[] args) {
        try {
            DatagramSocket socket = new DatagramSocket();
            String mensaje = "¡Hola desde el cliente UDP!";
```

```
byte[] buffer = mensaje.getBytes();

// Enviar datos al servidor UDP
InetAddress direccion = InetAddress.getByName("127.0.0.1");
DatagramPacket paquete = new DatagramPacket(buffer, buffer.length, direccion, 8080);
socket.send(paquete);

// Cerrar el socket
socket.close();
} catch (Exception e) {
    e.printStackTrace();
}
}
```

---

## Resumiendo

Los protocolos de comunicaciones son esenciales para permitir que los dispositivos en una red intercambien datos de manera eficiente. TCP ofrece confiabilidad y orden, ideal para aplicaciones críticas, mientras que UDP prioriza la velocidad, perfecto para aplicaciones en tiempo real. Estos protocolos forman la base de muchas aplicaciones modernas y son fundamentales para entender cómo funcionan las redes de comunicación.