<u>Cursillo de Redes:</u> Iniciación al funcionamiento de una red y los Protocolos que la conforman.

Cuadernillo de Teoría y Ejercicios simples para "jugar" con lo aprendido.

Índice:

1. Introducción a las REDES:

1.1. ¿Qué es una red?

Una **red** es un **conjunto de dispositivos interconectados** (ordenadores, móviles, servidores, impresoras, sensores, etc.) que se comunican entre sí para **compartir información y recursos**.

Para entendernos dentro de las redes, existen términos y características de estas que es importante entender:

1. **Nodos:** Los **dispositivos finales** que se conectan a la red.

   Ejemplos: ordenadores, teléfonos, routers, impresoras, servidores.

2. **Medios de transmisión:** Son los "caminos" por los que viaja la información. Existen dos tipos: físicos e inalámbricos.

   - **Ejemplos Físicos**: cables de cobre (Ethernet), fibra óptica.
   - **Ejemplos Inalámbricos**: WiFi, Bluetooth, ondas de radio, 4G/5G.

3. **Protocolos de comunicación**: Conjunto de reglas que permiten a los dispositivos entenderse.

   Ejemplos: TCP/IP, HTTP, DNS, DHCP.

4. **Direccionamiento**: Cada dispositivo necesita una "dirección" única para identificarse en la red.

   Existen dos tipos de direccionamiento en redes: **dirección IP (lógica)** y **dirección MAC (física).**

   - **Dirección IP (lógica):** número asignado a un dispositivo para identificarlo dentro de una red; **puede cambiar** (dinámica con DHCP) **o mantenerse fija** (estática), y existen dos versiones: **IPv4 y IPv6.**
     - **Ejemplo: IPv4:** 192.168.1.25
     - **Ejemplo IPv6:** 2001:0db8:85a3:0000:0000:8a2e:0370:7334
   - **Dirección MAC (física):** código único grabado en la tarjeta de red por el fabricante; identifica de forma permanente al dispositivo a nivel de hardware y no cambia, aunque se conecte a distintas redes. No hay dos dispositivos diferentes con la misma MAC.

5. **Seguridad:** La seguridad en redes es el conjunto de medidas, técnicas y protocolos que buscan proteger la información y los dispositivos conectados a una red frente a accesos no autorizados, ataques o pérdidas de datos. Existen **tres objetivos principales dentro de la seguridad:**
   - **Confidencialidad**: que la información solo pueda ser leída por quien está autorizado.
   - **Integridad**: que los datos no sean modificados en el camino a su destino.
   - **Disponibilidad**: que los servicios estén accesibles siempre que se necesiten.

1.2. Direccionamiento IP

El **direccionamiento IP** es el mecanismo que permite que cada dispositivo conectado a una red tenga una **identidad única y reconocible**. Gracias a esta dirección, los datos saben exactamente **a qué equipo deben llegar** y cómo moverse entre diferentes redes. Sin direcciones IP, la comunicación en Internet sería imposible, porque los paquetes no tendrían un destino definido. Existen **dos versiones** de direcciones: **IPv4** y **IPv6**.

1.2.1. Direccionamiento IPv4

El **IPv4 (Internet Protocol version 4)** es la versión más utilizada del protocolo de direccionamiento en redes.

Cada dirección IPv4 está formada por **32 bits**, que se suelen representar en forma de **cuatro números decimales** separados por puntos, cada uno entre 0 y 255. Por ejemplo: **192.168.1.25**.

Una dirección IPv4 se compone de dos partes:

- La **parte de red**, que identifica a la subred a la que pertenece el dispositivo (usando la máscara de subred, esta parte se identifica con la cantidad de "1s" en binario).
- La **parte de host**, que identifica al dispositivo concreto dentro de esa red (usando la máscara de subred, esta parte se identifica con la cantidad de "0s" en binario).

Para separar estas dos partes se utiliza la **máscara de subred**. Un ejemplo muy común es **255.255.255.0**, que en binario sería 11111111 11111111 11111111 00000000, lo que indica que los tres primeros bloques (en nuestro ejemplo **192.168.1**) corresponden a la red y el último bloque (en nuestro ejemplo **.25**) al dispositivo.

La máscara de red también puede representarse en notación **CIDR** (Classless Inter-Domain Routing), como **192.168.1.25/24**, donde el "/24" significa que los 24 primeros bits son de red.

Las direcciones IPv4 pueden clasificarse en dos grandes grupos:

- **Direcciones públicas**: son las que se utilizan para identificar dispositivos en **Internet**. Cada una es **única en todo el mundo**, lo que significa que no puede haber dos equipos distintos en Internet con la misma dirección pública. Por ejemplo, un servidor web o el router de tu casa, cuando se conecta a Internet, utiliza una IP pública asignada por el proveedor de Internet (ISP).
- **Direcciones privadas**: están pensadas para usarse **solo dentro de redes internas**, como en casas, oficinas o universidades. Estas direcciones no pueden usarse directamente en Internet, es decir, si un dispositivo tiene una IP privada, necesita que el router traduzca esa dirección a una pública mediante un mecanismo llamado **NAT (Network Address Translation)**. Los rangos reservados para direcciones privadas son:
  - **10.0.0.0 – 10.255.255.255** (10.0.0.0/8)
    - **Rango:** ~16,7 millones de direcciones.
    - **Uso típico:** grandes empresas, operadores de red, universidades.
  - **172.16.0.0 – 172.31.255.255** (172.16.0.0/12)

        - **Rango:** ~1 millón de direcciones.
        - **Uso típico:** redes empresariales de tamaño medio.
    - **192.168.0.0 – 192.168.255.255** (192.168.0.0/16)
        - **Rango:** ~65.000 direcciones.
        - **Uso típico:** hogares y pequeñas oficinas (el clásico "192.168.1.x").

En la práctica, casi todos los routers domésticos asignan direcciones privadas del rango **192.168.x.x** a los dispositivos conectados (ordenadores, móviles, consolas…). **Desde fuera, todos estos equipos parecen compartir la misma IP pública del router, aunque internamente cada uno tenga su propia dirección privada.**

1.2.2. Direccionamiento IPv6

IPv6 (Internet Protocol version 6) es la evolución de IPv4 y se creó para resolver su gran limitación: el número reducido de direcciones disponibles. Mientras que IPv4 utiliza **32 bits** (unos 4.300 millones de direcciones), IPv6 usa **128 bits**, lo que significa que el espacio de direcciones es prácticamente infinito para nuestras necesidades.

Las direcciones IPv6 se escriben en **ocho bloques de números hexadecimales** separados por dos puntos, por ejemplo:

```
2001:0db8:85a3:0000:0000:8a2e:0370:7334
```

Para simplificarlas se pueden eliminar ceros innecesarios, por ejemplo:
`2001:db8:85a3::8a2e:370:7334`

En IPv6 existen varios tipos de direcciones, pero a nivel básico basta con saber:

- **Globales**: equivalentes a las direcciones públicas de IPv4, sirven para acceder a Internet.
- **Link-local**: se usan dentro de la red local y empiezan por `fe80::`.
- **Multicast**: permiten enviar un mensaje a varios equipos a la vez.

Una diferencia clave respecto a IPv4 es que en IPv6 ya no hace falta **NAT**, porque hay direcciones de sobra para que cada dispositivo tenga la suya.

Aunque IPv6 todavía no está tan extendido como IPv4, cada vez más operadores y servicios de Internet lo utilizan, ya que es la base del crecimiento futuro de la red.

EJERCICIO: Por ejemplo, en la UPV se utilizan tanto Ipv4 como Ipv6, puedes comprobarlo con el comando **IPCONFIG** en la terminal (cmd) dentro de la red de la UPV.

1.3. Modelo OSI y modelo TCP/IP

1.3.1. Modelo OSI

Para entender cómo viajan los datos en una red, se utilizan modelos de referencia que dividen la comunicación en capas.

El más conocido es el modelo **OSI (Open Systems Interconnection),** creado por la ISO (International Organization for Standardization) en los años 80.
Su objetivo fue estandarizar las comunicaciones entre equipos de distintos fabricantes, de forma que todos "hablaran el mismo idioma".

El modelo OSI tiene **7 capas**, cada una con funciones específicas:

1. **Física**
2. **Enlace de datos**
3. **Red**
4. **Transporte**
5. **Sesión**
6. **Presentación**
7. **Aplicación**

1.3.2. Modelo TCP/IP

Sin embargo, **en la práctica Internet funciona con el modelo TCP/IP, más sencillo y eficiente, que se organiza en 5 capas.** Cada capa solo se comunica directamente con la capa superior e inferior, nunca se saltan capas. El funcionamiento es el siguiente:

- **Encapsulación (en el emisor):**
  Cada capa recibe los datos de la capa superior, les añade información de control (cabeceras, direcciones, puertos…) y los entrega a la capa inferior.
- **Desencapsulación (en el receptor):**
  Cada capa elimina la información que no le corresponde y entrega los datos a la capa superior.

Las 5 capas mencionadas son las siguientes:

1. **Capa física**
   La capa física es como la **carretera FÍSICA por la que viajan los datos**.
   Su función principal es **transmitir bits (0 y 1)** a través de un medio de comunicación, convirtiendo la información digital en **señales eléctricas, ópticas o electromagnéticas** que viajan de un dispositivo a otro.

2. Capa de Enlace de Datos

La **capa de enlace de datos** se encarga de que los datos viajen de forma correcta **entre dos dispositivos conectados en la misma red física** (ejemplo: un PC y el router de casa).
Su función es **organizar los bits de la capa física en tramas**, detectar errores básicos y usar direcciones físicas (**MAC**) para identificar a cada dispositivo.
Dentro de esta capa, los protocolos más comunes con los que se trabaja son: Ethernet (IEEE 802.3), WiFi (IEEE 802.11), PPP (Point-to-Point Protocol) y ARP (Address Resolution Protocol).

3. Capa de Red

La **capa de red** es la responsable de llevar los datos desde el dispositivo de origen hasta el de destino, incluso cuando están en **redes diferentes**.
Aquí aparece el concepto clave: **direccionamiento lógico (IP)** y **enrutamiento**.
Dentro de esta capa, los protocolos principales son: IP (Internet Protocol), ICMP (Internet Control Message Protocol) y IGMP (Internet Group Management Protocol).

4. Capa de Transporte

La **capa de transporte** asegura que la información llegue **completa y en orden** desde la aplicación del emisor hasta la aplicación del receptor.
Es la que se preocupa de la **fiabilidad de la comunicación**.
Sus funciones principales son la segmentación de los datos, la multiplexación (que varias apps usen la red a la vez), la fiabilidad (en el caso del protocolo TCP), el control de flujo, y la detección de errores.
Los protocolos principales dentro de esta capa son TCP (Transmission Control Protocol) y UDP (User Datagram Protocol).

5. Capa de Aplicación

La capa de aplicación es la más cercana al usuario. En ella viven los **protocolos que hacen posible los servicios que usamos día a día**: web, correo, chats, etc.
Los protocolos más comunes de esta capa son: HTTP/HTTPS, DNS, SMTP, IMAP, POP3, FTP, SFTP, DHCP…

2. Protocolos principales

2.1. Protocolos de Enlace de Datos

Los **protocolos de enlace de datos** son los encargados de que la comunicación dentro de una red local funcione de manera ordenada y sin errores. Su papel comienza una vez que los bits ya están viajando por el medio físico (cables, fibra o WiFi) y consiste en organizarlos en unidades manejables llamadas **tramas**. Además, añaden información de control para que el receptor pueda detectar si se ha producido algún error en la transmisión y, en algunos casos, corregirlo.

En esta capa también se utilizan las direcciones físicas **MAC**, que identifican de manera única a cada tarjeta de red. Gracias a ellas, los dispositivos saben exactamente a quién va dirigido un mensaje dentro de la red local.

2.1.1. Protocolo Ethernet

El protocolo **Ethernet se encuentra en la capa de enlace de datos (capa 2)**, aunque también define aspectos de la capa física, como el tipo de cableado y los conectores. Es el protocolo más utilizado en las redes cableadas y el que permite que los dispositivos de una red local (LAN) se comuniquen de forma rápida y ordenada.

Su función principal es **administrar cómo viajan los datos por el cable**. Define la **capacidad de transmisión** *(velocidades como 100 Mbps, 1 Gbps o más de 100 Gbps en redes modernas)*, el **tipo de *cable*** *(cobre o fibra óptica)* y la manera en que la información se organiza en **tramas**.

Cada trama de Ethernet incluye las **direcciones MAC** de origen y destino. Estas direcciones, únicas para cada tarjeta de red, aseguran que los datos lleguen exactamente al dispositivo correcto. Además, las tramas llevan un control de errores para verificar que la información se ha transmitido sin fallos.

En el día a día, cada vez que conectamos un ordenador al router con un cable de red, estamos utilizando Ethernet.

2.1.1. Protocolo ARP

El protocolo **ARP (Address Resolution Protocol)** se sitúa en la **capa de enlace de datos (capa 2),** ya que trabaja directamente con direcciones físicas **MAC**, aunque actúa como un puente con la capa de red (IP).

Su función es muy concreta: **traducir direcciones IP en direcciones MAC**. En una red local, un ordenador sabe la dirección IP a la que quiere enviar datos, pero para que la comunicación funcione necesita conocer también la dirección MAC del dispositivo de destino.

Para resolverlo, el equipo envía un mensaje ARP a toda la red preguntando: *"¿Quién tiene esta IP?"*. El dispositivo que posee esa dirección responde con su dirección MAC, y a partir de ese momento los dos equipos pueden comunicarse de manera directa a nivel físico.

Este proceso ocurre de manera automática y es invisible para el usuario, pero es esencial para que las redes funcionen. Sin ARP, los ordenadores podrían saber qué dirección lógica (IP) quieren alcanzar, pero no tendrían forma de encontrar físicamente al destinatario dentro de la red.

Un ejemplo cotidiano de ARP lo encontramos en casa: cuando un portátil se conecta al WiFi e intenta imprimir un documento en la impresora de red, lo primero que hace es enviar una petición ARP para averiguar qué dirección MAC corresponde a la IP de la impresora. Una vez resuelto, ya puede enviarle el archivo y la impresión se realiza sin problemas.

2.2. Protocolos de Red

Los **protocolos de red** permiten que los datos viajen de un dispositivo a otro incluso cuando están en redes distintas. Su misión principal es gestionar el **direccionamiento lógico** mediante direcciones IP y decidir la **ruta** que seguirá la información hasta llegar a su destino.

2.2.1 Protocolo ICMP

El protocolo **ICMP (Internet Control Message Protocol)** se encuentra en la **capa de red (capa 3)**, ya que funciona junto con el protocolo IP. Su misión no es transportar datos de usuario, sino **enviar mensajes de control y diagnóstico** que informan sobre el estado de la red.

Gracias a ICMP, los dispositivos pueden avisar cuando un paquete no llega a su destino, cuando una ruta no está disponible o cuando la red está congestionada. Estos mensajes permiten detectar problemas y ayudan a que la comunicación en Internet sea más fiable.

El ejemplo más conocido de ICMP es el comando **ping**. Al escribir ping www.google.com, el ordenador envía un mensaje ICMP a esa dirección y espera respuesta. Si el servidor responde, sabemos que hay conexión y podemos medir cuánto tarda en contestar (latencia). Si no responde, significa que hay un problema de conectividad.

Aunque el usuario normal no lo vea, ICMP es una de las herramientas más utilizadas por técnicos y administradores de red, ya que permite comprobar rápidamente si un dispositivo está en línea o si existe un fallo en la comunicación.

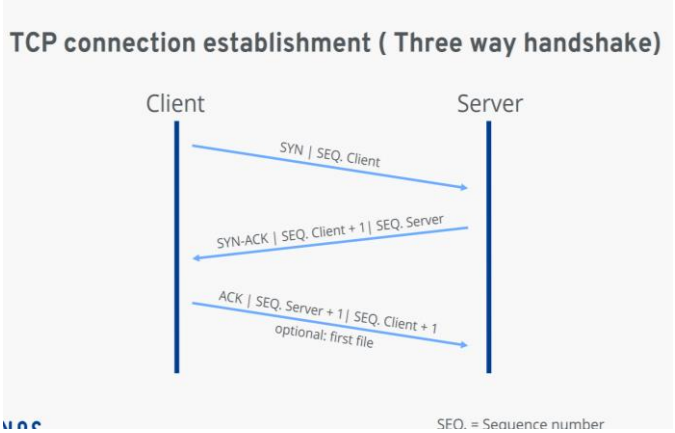
2.3. Protocolos de Transporte

Dentro de la capa de transporte existen dos protocolos fundamentales: **TCP (Transmission Control Protocol)** y **UDP (User Datagram Protocol)**. Ambos se encargan de que los datos de las aplicaciones lleguen a su destino, pero lo hacen de manera muy distinta.

2.3.1 Protocolo TCP

TCP es un protocolo **orientado a la conexión**. Esto significa que antes de enviar datos establece una comunicación entre el emisor y el receptor, asegurando que ambos están preparados para hablar.

Además, TCP **garantiza que los datos lleguen completos, sin errores y en el mismo orden en el que fueron enviados**. Para lograrlo, divide la información en segmentos, los numera y espera confirmaciones de recepción (los llamados ACK). Si detecta que algún segmento se ha perdido, lo reenvía automáticamente. Este proceso hace que TCP sea muy fiable, aunque un poco más lento.



Antes de empezar realmente a transmitir datos, TCP realiza un proceso especial conocido como *three-way handshake* (apretón de manos en tres pasos). En este intercambio, el cliente envía un primer mensaje **SYN** (*synchronize*, mensaje para sincronizar) para indicar que quiere iniciar la comunicación. El servidor responde con un **SYN-ACK** (*synchronize-acknowledgment*, mensaje para confirmar recibimiento), confirmando que está disponible y preparado. Finalmente, el cliente devuelve un **ACK** (*acknowledgment***)**, cerrando el ciclo y dejando establecida la conexión. **Solo a partir de ese momento comienza el envío de información real**. Este mecanismo asegura que tanto emisor como receptor están sincronizados y listos para comunicarse.

Otro detalle interesante es que TCP utiliza números de secuencia y números de confirmación para controlar el flujo de datos. Esto no solo garantiza el orden, sino que también permite detectar pérdidas y evitar que un receptor se vea saturado si recibe más información de la que puede procesar.

TCP se utiliza en servicios donde la exactitud de los datos es esencial, como en la navegación web (HTTP/HTTPS), el correo electrónico (SMTP, IMAP, POP3) o la transferencia de archivos (FTP).

2.3.2 Protocolo UDP

UDP, en cambio, es un protocolo mucho más simple y rápido. **No establece conexión previa ni comprueba si los datos llegan al destino**. Envía la información en bloques llamados datagramas, sin preocuparse de si se pierden o llegan desordenados.

Por eso se dice que no es fiable, pero al mismo tiempo es más eficiente cuando lo que importa es la velocidad y no la perfección de los datos.

Se emplea en aplicaciones en tiempo real, como las videollamadas, el streaming de música y vídeo, o los videojuegos en línea, donde perder algún paquete no afecta demasiado a la experiencia, pero la rapidez es fundamental.

2.4. Protocolos de aplicación

Un **protocolo de aplicación** es un conjunto de reglas que define cómo las aplicaciones que utilizan una red deben comunicarse para ofrecer servicios al usuario final. Se encuentra en la capa superior del modelo TCP/IP y constituye el punto de contacto más directo entre la red y las personas que la usan.

Gracias a estos protocolos es posible realizar tareas cotidianas como acceder a una página web, enviar un correo electrónico, descargar un archivo o establecer una videollamada.

2.4.1 El protocolo DHCP

El **DHCP (Dynamic Host Configuration Protocol)** es un protocolo de aplicación cuya función principal es **simplificar la configuración de los dispositivos en una red**. En lugar de que cada usuario tenga que asignar manualmente una dirección IP, una máscara de subred, una puerta de enlace y otros parámetros de red a su ordenador o móvil, el protocolo DHCP permite que todo esto se haga de forma automática y dinámica.

Cuando un dispositivo se conecta a una red, lo primero que hace es enviar una petición de configuración utilizando un mensaje especial llamado **DHCP Discover**. Esta señal busca a un servidor DHCP dentro de la red. El servidor responde con un mensaje denominado **DHCP Offer**, en el que ofrece una dirección IP disponible junto con la configuración necesaria para que el dispositivo pueda comunicarse correctamente: máscara de subred, puerta de enlace predeterminada, direcciones de servidores DNS, e incluso el tiempo durante el cual la dirección será válida (lo que se denomina *lease time* o tiempo de concesión).

El cliente, al recibir la oferta, envía un mensaje de confirmación llamado **DHCP Request**, en el que indica que acepta la dirección propuesta. Finalmente, el servidor responde con un **DHCP Acknowledgment (ACK)**, confirmando el proceso y asignándole oficialmente esa dirección IP al dispositivo. De esta manera, con un intercambio de mensajes muy sencillo —Discover, Offer, Request y Acknowledgment—, el dispositivo queda configurado y listo para usar la red sin intervención manual del usuario.

Otra característica importante es que la asignación de direcciones suele ser temporal. Al cumplirse el tiempo de concesión, el dispositivo debe renovar la dirección o solicitar una nueva. Esto permite reutilizar direcciones IP y aprovechar mejor el espacio disponible, especialmente en redes muy dinámicas como las de oficinas, aeropuertos o cafeterías.

La principal ventaja de DHCP es la **automatización**. En redes grandes, como las de empresas o universidades, sería prácticamente imposible gestionar las direcciones manualmente para cientos o miles de dispositivos. Con DHCP, el servidor mantiene un control centralizado del rango de direcciones disponibles y **evita conflictos como que dos dispositivos usen la misma IP.**

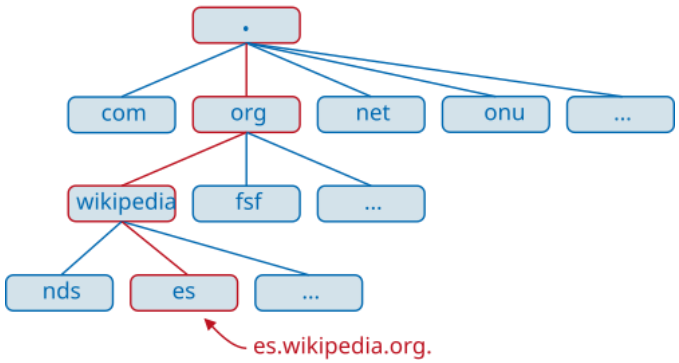
En la práctica cotidiana, cada vez que un teléfono móvil se conecta a una red WiFi, es **DHCP quien le proporciona automáticamente la dirección IP y todos los**

parámetros necesarios. Sin este protocolo, el usuario tendría que introducirlos manualmente, lo que resultaría poco práctico e incluso confuso para la mayoría.

2.4.2. El protocolo DNS

El **DNS (Domain Name System)** es uno de los pilares fundamentales de Internet, ya que actúa como un sistema de traducción entre los nombres fáciles de recordar por los humanos y las direcciones numéricas que utilizan los ordenadores para comunicarse. En lugar de memorizar largas direcciones IP como *142.250.190.78*, los usuarios pueden simplemente escribir en su navegador un nombre como *www.google.com*. El DNS se encarga de transformar ese nombre en la dirección IP correspondiente, de manera que el navegador sepa a qué servidor conectarse.

El funcionamiento de DNS se basa en una gran **base de datos distribuida y jerárquica** repartida por todo el mundo. Esta jerarquía se organiza en niveles: en la parte superior se encuentran los **servidores raíz**, que conocen la ubicación de los servidores de dominio de primer nivel (como *.com*, *.org*, *.es*). Cada uno de estos, a su vez, sabe dónde encontrar los servidores responsables de dominios concretos, como *google.com* o *wikipedia.org*.



Cuando un usuario escribe una dirección web en su navegador, como por ejemplo *www.google.com*, el ordenador no entiende directamente ese nombre. Lo que realmente necesita es la dirección IP del servidor donde está alojada la página, que es un número parecido a *142.250.190.78*. El **DNS (Domain Name System)** funciona como una guía telefónica de Internet: traduce los nombres fáciles de recordar en los números que entienden los ordenadores.

El proceso de traducción ocurre de forma automática y muy rápida. Normalmente es el propio sistema operativo o el router de casa quien pregunta a un servidor DNS, que suele pertenecer al proveedor de Internet o a servicios públicos muy conocidos como los de **Google (8.8.8.8)** o **Cloudflare (1.1.1.1).** Ese servidor busca la dirección IP correspondiente y se la devuelve al ordenador. Para que la próxima vez sea más rápido, la respuesta se guarda durante un tiempo en una memoria temporal llamada **caché**.

La importancia de DNS es enorme aunque pase desapercibido. Sin él, las personas tendríamos que memorizar direcciones IP para cada página que queremos visitar, lo cual sería imposible en un Internet con millones de sitios web.

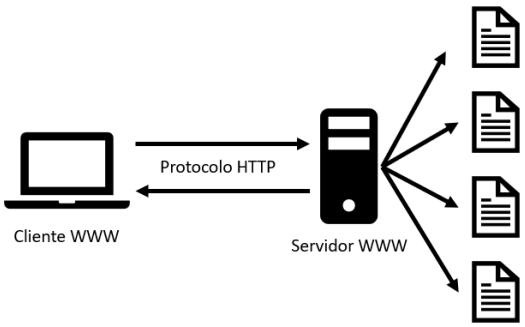
No obstante, debido a su papel crítico, **DNS también puede ser un punto vulnerable**. Ataques como la suplantación de respuestas (**DNS spoofing**) o la denegación de servicio (**DDoS sobre servidores DNS**) pueden afectar a la disponibilidad de páginas web. Para reducir estos riesgos, se han desarrollado mejoras como **DNSSEC**, que añade mecanismos de validación, o **DoH** y **DoT**, que cifran las consultas para proteger la privacidad de los usuarios.

En resumen, lo fundamental para un principiante es entender que DNS es el traductor que hace posible que podamos navegar escribiendo nombres en lugar de números, y que aunque funciona de manera invisible y automática, es una pieza clave y delicada de todo Internet.

2.4.3 El protocolo HTTP

Entre todos los protocolos de aplicación, el **HTTP (HyperText Transfer Protocol)** es uno de los más importantes y conocidos, ya que hace posible la navegación por la World Wide Web (WWW). HTTP regula la comunicación entre un cliente (normalmente un navegador web) y un servidor que almacena y distribuye páginas y recursos. La relación sigue un **modelo cliente-servidor**: el cliente envía solicitudes y el servidor responde entregando los datos solicitados.

El intercambio se basa en mensajes de petición y de respuesta. Una petición habitual es el método "**GET**", mediante el cual el navegador solicita un recurso concreto, como un documento HTML o una imagen. El servidor procesa la petición y envía una respuesta que incluye el recurso junto con información de control, como el tipo de contenido o el estado de la operación.



Una característica fundamental de HTTP es que se trata de un **protocolo sin estado**. Esto significa que cada transacción es independiente y el servidor no guarda memoria de lo que ocurrió en solicitudes previas. Para mantener continuidad en la interacción (por ejemplo, en una sesión de usuario) se emplean mecanismos adicionales como **cookies o identificadores de sesión.**

El protocolo HTTP funciona habitualmente en el **puerto 80**, mientras que su versión segura, conocida como HTTPS, opera en el **puerto 443**. HTTPS incorpora cifrado mediante **TLS/SSL**, lo cual garantiza la confidencialidad de la información transmitida, así como su integridad frente a posibles manipulaciones durante el tránsito.

Ejemplo de uso: En la práctica, cada vez que un usuario escribe en su navegador la dirección de un sitio, como "www.wikipedia.org", el navegador genera una petición HTTP hacia el servidor correspondiente. El servidor responde con el código HTML de la página y el navegador interpreta ese código para mostrarlo de forma comprensible y visual al usuario final.

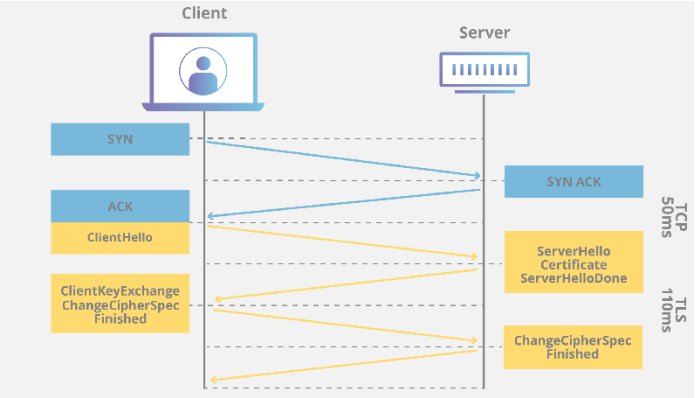
2.4.4. TLS/SSL: Añadimos seguridad

TLS (Transport Layer Security) y su antecesor **SSL (Secure Sockets Layer)** son protocolos diseñados para proteger las comunicaciones en la red. **Su papel es añadir una capa de seguridad entre el transporte y la aplicación** (por lo que realmente no pertenece a una sola capa, sino que está en mitad de las dos), de modo que cualquier dato que viaje por Internet lo haga de forma cifrada y con garantías de autenticidad e integridad.

Gracias a TLS/SSL, aunque alguien intercepte la información que se transmite, no podrá leerla ni modificarla. Además, mediante certificados digitales, estos protocolos permiten verificar que el servidor al que nos conectamos es realmente quien dice ser, evitando fraudes y suplantaciones.

El uso más visible es en la web, con **HTTPS**, pero TLS también se aplica en muchos otros servicios: en el correo electrónico (SMTP, IMAP y POP3 sobre TLS), en la transferencia de archivos (FTPS), en conexiones remotas seguras como algunas VPN, e incluso en aplicaciones de mensajería y telefonía por Internet (VoIP).

En cuanto a su funcionamiento, **TLS y SSL establecen primero una especie de "apretón de manos" (handshake)**. Durante este proceso, el cliente y el servidor acuerdan qué algoritmos de cifrado van a usar y se intercambian claves de forma segura. Una vez hecho esto, toda la comunicación posterior viaja cifrada con esas claves, de manera que solo el cliente y el servidor pueden entenderla.



Es importante diferenciar entre ambos: **SSL fue la primera versión, creada en los años 90, pero hoy en día está obsoleta por tener vulnerabilidades de seguridad. La versión moderna y segura es TLS, que es la que se utiliza actualmente en casi todos los servicios de Internet.** Aun así, por costumbre se sigue usando el término "SSL" en muchas ocasiones, incluso cuando en realidad se está empleando TLS.

En la práctica cotidiana, cada vez que vemos el candado en el navegador al entrar en un banco, en una red social o en una tienda online, es TLS quien está trabajando en segundo plano para que nuestra contraseña o nuestros datos personales viajen seguros.

Para que el uso de **TLS** resulte más ilustrativo, podemos fijarnos en cómo lo aplican organizaciones de confianza. Un buen ejemplo es la **Universitat Politècnica de València (UPV)**, cuya página web oficial *https://www.upv.es* utiliza este protocolo de seguridad para proteger las comunicaciones entre los usuarios y sus servidores.