

Introducción a Redes

Módulo 3

Modelo de capas OSI

Modelo de capas OSI

Como se mencionó anteriormente, este es un modelo que **describe las etapas** en un proceso de comunicación, pero no termina el cómo. Un modelo de referencia nos ayuda a comprender los procesos y que se realiza en cada capa y nos proporciona los siguientes beneficios:

- **Facilita la comprensión** al dividir un problema complejo en partes más simples.
- **Evita los problemas** de compatibilidad, por ejemplo de red.
- **Detalla las capas** para su mejor aprendizaje.

- Proporciona a los fabricantes un **conjunto de estándares** que aseguran una mayor **compatibilidad** e **interoperabilidad** entre los distintos tipos de tecnología de red utilizados por las empresas a nivel mundial.



Capas del modelo OSI

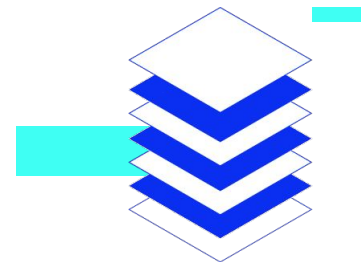
El **modelo de interconexión de sistemas abiertos** (OSI) se divide en **7 capas** que podemos organizar en dos grupos:

Capas de host

- Aplicación.
- Presentación.
- Sesión.
- Transporte.

Capas de medio

- Red.
- Enlace de datos.
- Física.



Cuando describimos los **elementos de una red** mencionamos los elementos que **hacen uso de una red** (tablets, pc, smartphones, etc), es decir los hosts, y los elementos que **proveen acceso a la red** y permiten el intercambio de información (NICs, SWITCHs, medios, etc.) entre hosts.

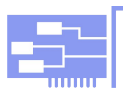
Es importante diferenciar entre estos dos niveles, ya que en ellos suceden procesos distintos.

A nivel de hosts se **generan** los mensajes, a nivel de medio se **transportan** esos mensajes entre los hosts.

Nivel de host



Nivel de red



nic



Capa 7: Aplicación

En esta capa se ubican las **aplicaciones que hacen uso de los servicios de red**. Es donde el usuario u otros servicios generan los mensajes que pasarán a las capas inferiores.

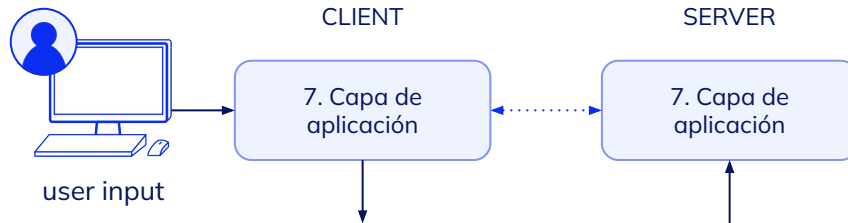
Las aplicaciones de esta capa no solo generan los mensajes, sino que también los interpretan.

El ejemplo típico es el cliente de email: un usuario puede escribir y enviar un correo electrónico usando el cliente *Thunderbird*. El destinatario puede recibir y leer el correo electrónico usando *Outlook*.

Algunos ejemplos de aplicaciones que hacen uso de servicios de red:

- Mensajería instantánea: Messenger, Telegram, Whatsapp, Signal.
- Navegadores web: Firefox, Chrome, Internet Explorer, Edge, Opera.
- Servidores: Mysql, Apache, Vsftpd, Nginx.
- Videojuegos: League of Legends, Counter Strike, Minecraft.
- Streaming y videoconferencia: Zoom, Meets, Jitsi.

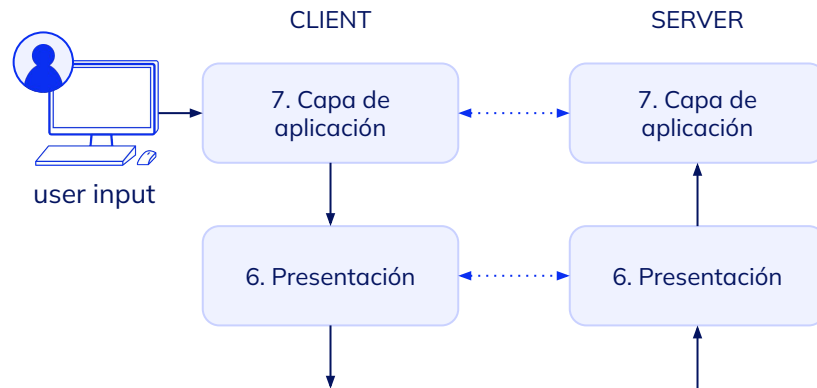
Vale decir que cuando hablamos de mensajes no nos referimos estrictamente a mensajes escritos, como un correo, sino a la **información** que genera la aplicación y que debe ser enviada a otro host.



Capa 6: Presentación

Su objetivo es encargarse de la **representación de la información**, de manera que aunque distintos equipos puedan tener diferentes representaciones internas de caracteres, números, sonido o imágenes, los datos lleguen de manera reconocible.

En esta capa es donde se comienzan a **aplicar los protocolos de aplicación**, continuemos con el ejemplo del correo electrónico y el de acceso a contenido web que utilizan protocolos abiertos.



Protocolo para transferencia simple de correo (SMTP)

Es el protocolo usado para el envío de correos electrónicos.

Redacción de mensaje

Mensaje escrito por el usuario mediante una aplicación de correo electrónico como Outlook.

Destinatario: `jorge@example.com`

Cuerpo del mensaje: "Hola ¿Qué tal tus vacaciones?"



Mensaje que se envía bajo el protocolo SMTP

La aplicación enviará un mensaje mucho más **complejo**, y normalmente, invisible al usuario.

Es un lenguaje que hablan entre aplicaciones, no entre usuarios.

```
S: 220 Servidor SMTP
C: HELO miequipo.midominio.com
S: 250 Hello, please to meet you
C: MAIL FROM: <yo@midominio.com>
S: 250 Ok
C: RCPT TO: <jorge@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: Subject: Campo de asunto
C: From: yo@midominio.com
C: To: jorge@example.com
C:
C: Hola ¿Qué tal tus vacaciones?
C: .
C: <CR><LF>.<CR><LF>
S: 250 Ok: queued as 12345
C: quit
S: 221 Bye
```

El Protocolo de transferencia de hipertexto (HTTP)

Mensaje enviado desde un navegador web

El mensaje que se envía es la url que se ingresa en la barra de direcciones de un navegador.

```
url: https://educacionit.com.ar
```

Mensaje que se envía bajo el protocolo HTTP

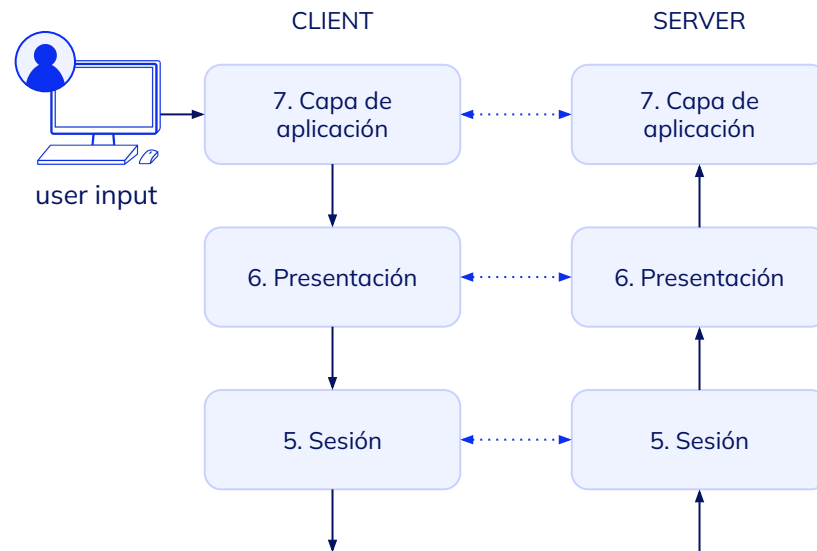
El mensaje que recibirá el servidor es la solicitud (GET) del contenido del directorio raíz (/) alojado en un servidor web bajo el nombre de host 'educacionit.com.ar'.

```
GET / HTTP/1.1  
Host: educacionit.com.ar  
Accept-Language: es
```

Capa 5: Sesión

Establece, administra y finaliza las conexiones entre las aplicaciones locales y las remotas. Esta capa también permite:

- Cifrar los datos y comprimirlos.
- Establece y finaliza las conexiones.
- Proporciona sus servicios a la capa de presentación.
- Sincroniza el diálogo entre las capas de presentación de los dos hosts y administra su intercambio de datos.



En resumen, en esta capa se establecen los **mecanismos** que permiten que **las aplicaciones envíen y reciban los mensajes** entre ellas.

El tiempo que transcurre entre la apertura de la comunicación y el cierre de esta se conoce como **sesión**. La capa de sesión garantiza que la sesión permanezca abierta el tiempo suficiente como para transferir todos los datos que se están intercambiando.

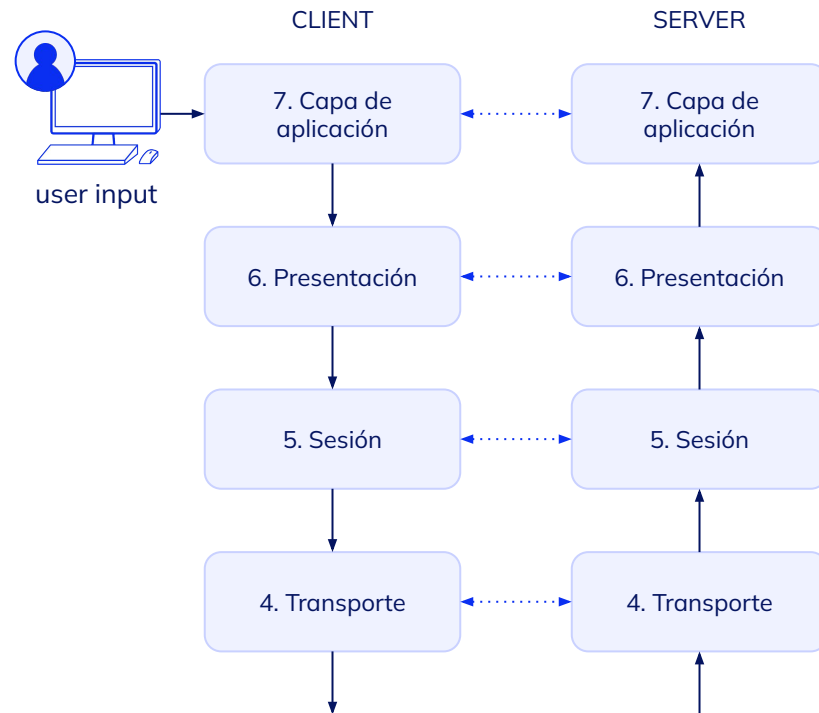


Capa 4: Transporte

La capa de transporte es la responsable de las **comunicaciones de extremo a extremo** entre dos dispositivos.

El mensaje que viene de la capa de sesión no se envía como tal, sino que se fragmenta en lo que se denomina “**segmentos**”, piezas más pequeñas que juntas componen el mensaje.

Los segmentos se reciben en el destino y el proceso de armado del mensaje a partir de los segmentos se produce en la misma capa en el extremo opuesto.



La capa de transporte es también la responsable del **control de flujo** y del **control de errores**.

El **control de flujo** sirve para determinar la velocidad óptima de transmisión que garantice que un emisor con velocidad de conexión alta no sobrecargue el segmento con menor capacidad de recepción, es decir, cuya conexión sea más lenta.

La **capa de transporte** realiza un control de errores en el extremo receptor, consistente en asegurarse de que todos los datos recibidos estén completos, y solicitará el reenvío en caso de que no.



Capa 3: Red

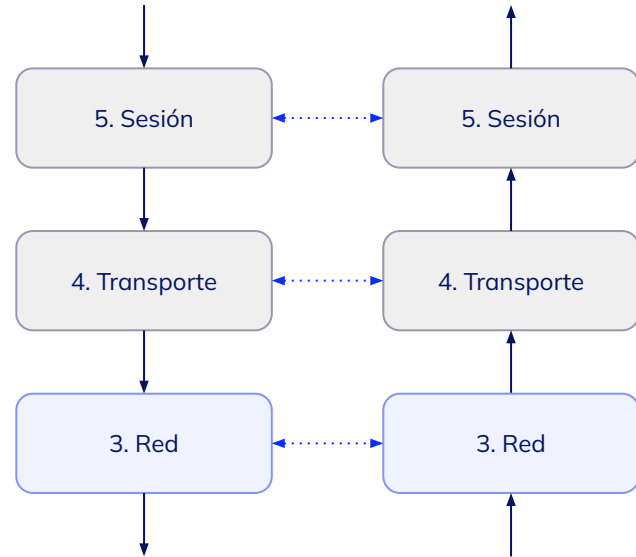
En este punto entramos en el rango de capas de medio o “*media layers*”.

Esta capa es responsable del **direccionamiento lógico** y el **dominio del enrutamiento**. Su misión es conseguir que los datos lleguen desde el origen al destino aunque no tengan conexión directa.

En esta capa los segmentos que llegan de la capa de transporte se encapsulan en lo que se denomina “paquete de datos”, y llevan consigo dos datos fundamentales: la **dirección del host emisor** y la **dirección del host receptor**.



Esta información permitirá a los protocolos y dispositivos de enrutamiento hacer llegar los paquetes entre hosts que estén en redes distintas, o en el caso que se encuentren en el mismo segmento lógico, hacer llegar el paquete al host correspondiente dentro del mismo dominio de broadcast junto con los métodos de direccionamiento físicos descritos en capas inferiores.



Capa 2: Enlace de datos

Proporciona **direccionamiento físico** y **procedimientos de acceso a medios**.

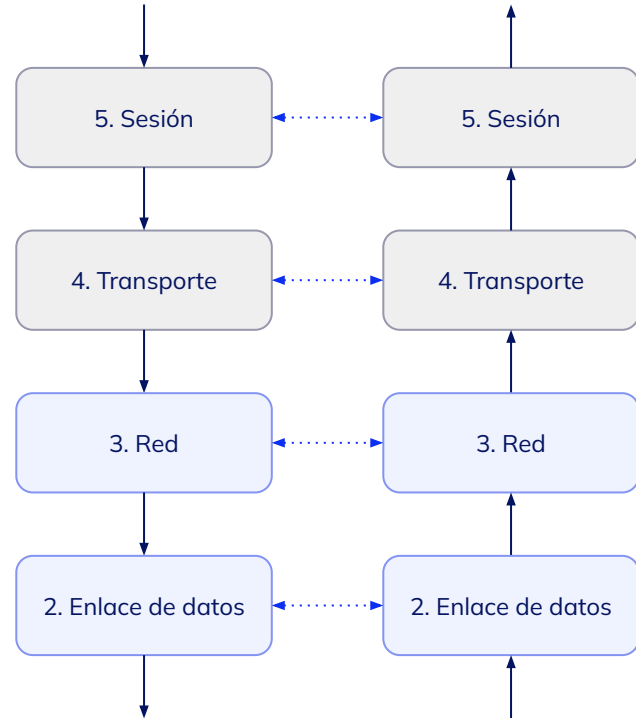
La capa de enlace de datos es muy similar a la capa de red, salvo que lo que hace es facilitar la transferencia de datos entre dos dispositivos ubicados en una misma red física.

La capa de enlace de datos toma los paquetes de la capa de red y genera trozos más pequeños denominados **tramas**. Estas tramas, así como los paquetes, tienen direcciones de origen y destino, pero en lugar de ser una dirección lógica es una dirección física, que es como se identifican las

interfaces de red dentro de un segmento físico de red que se puede representar a partir de la topología.

El **direccionamiento lógico** permite a un paquete llegar entre segmentos de red físicos y lógicos distintos. Una trama solo llega a interfaces de red a las que se pueda acceder físicamente, que estén conectados entre sí, por ejemplo, los hosts que orbitan un switch.

Al igual que la capa de red, la capa de enlace de datos es también la responsable del control de flujo y de errores respecto de esa comunicación dentro de la red (la capa de transporte solo realiza esto último respecto de comunicaciones entre redes).



Capa 1: Física

Define todas las **especificaciones eléctricas y físicas de los dispositivos**.

Básicamente son los **medios de transmisión de datos**, por ejemplo:

- Cables de cobre.
- Cables de fibra óptica.
- Ondas de radio (wifi, bluetooth, etc.)

La capa física define las especificaciones eléctricas, mecánicas, de procedimiento y funcionales para activar, mantener y desactivar el enlace físico entre sistemas finales.

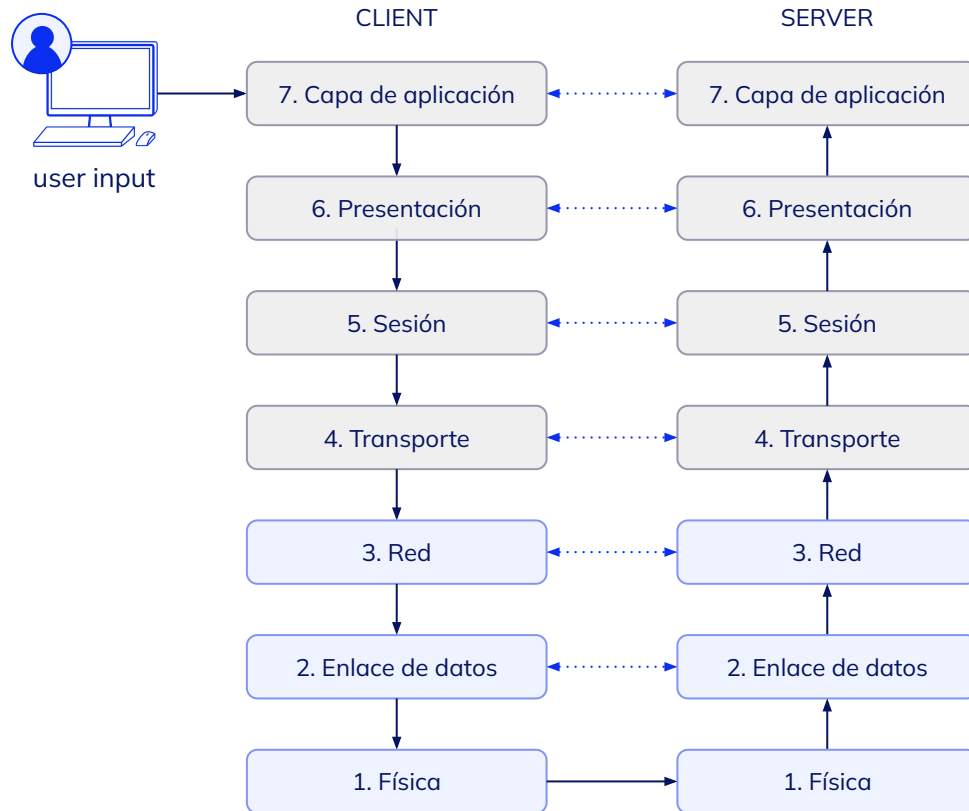
Las características tales como niveles de voltaje, temporización de cambios de voltaje, velocidad de datos físicos, distancias de transmisión máximas, conectores físicos y otros atributos similares son definidos por las especificaciones de la capa física.

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal que posibilita la transmisión de voz y datos entre diferentes dispositivos.

ADSL Consiste en una transmisión analógica de datos digitales apoyado en el par simétrico de cobre que lleva la línea telefónica convencional.

USB es un estándar industrial desarrollado en los años 1990 que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores, periféricos y dispositivos electrónicos. Consiste en una transmisión analógica de datos digitales apoyada en el par simétrico de cobre que lleva la línea telefónica convencional.





Conclusión

El **modelo OSI** describe el proceso de comunicación, no como se desarrolla ese proceso ni protocolos intervinientes ni tecnologías utilizadas.

De esta forma el modelo permite aislar el proceso de comunicación en capas que pueden ser observadas individualmente. Si una aplicación que debe enviar un mensaje a otra no lo está logrando, inspeccionar lo que se sucede en cada capa puede llevarnos a la identificación del problema y su resolución, desde la verificación del correcto conexionado e integridad de cables (capa 1) hasta las configuraciones de red de la aplicación (capa 4).

El modelo que se describe es **simétrico**, debe funcionar de igual manera en ambos sentidos y cada capa procesa la información generada en la misma capa del extremo opuesto.



**¡Sigamos
trabajando!**

