

**REDES**

**ADOLFO SANZ DE DIEGO**

**MÁSTER UAH**

**2 ACERCA DE**

## 2.1 AUTOR

- Adolfo Sanz De Diego
  - Blog: [asanzdiego.blogspot.com.es](http://asanzdiego.blogspot.com.es)
  - Correo: [asanzdiego@gmail.com](mailto:asanzdiego@gmail.com)
  - GitHub: [github.com/asanzdiego](https://github.com/asanzdiego)
  - Twitter: [twitter.com/asanzdiego](https://twitter.com/asanzdiego)
  - LinkedIn: [in/asanzdiego](https://in/asanzdiego)
  - SlideShare: [slideshare.net/asanzdiego](https://slideshare.net/asanzdiego)

# **3 EL MODELO OSI**

## 3.1 INTRODUCCIÓN

- El modelo OSI (Open Systems Interconnection Basic Reference Model) es el **modelo de referencia para la descripción de las arquitecturas de redes**.
- Antes del modelo OSI cada arquitectura de red dependía del fabricante y de protocolos propietarios.
- Aunque OSI sigue siendo el modelo teórico de referencia, en 1996 se renunció definitivamente a su implementación práctica debido a que, mientras se desarrollaban los trabajos de diseño y estandarización de OSI, el modelo TCP/IP se había ya convertido en el estándar.

## 3.2 CAPAS

- El modelo OSI **se divide en capas** de tal forma que cada capa resuelve un problema concreto y así la siguiente capa ya no ha de preocuparse de dicho problema.
- Esta arquitectura ayuda en el diseño de protocolos al tener estos menos funciones.
- Además los cambios en una capa no afectan al resto de capas.

## 3.3 CAPA FÍSICA

- La primera capa, la física, se encarga de **transmitir bits a lo largo de un medio de transporte**, ya sea de formá inalámbrica, por un cable de cobre o por fibra óptica.

## 3.4 CAPA DE ENLACE

- La segunda capa, la de enlace, se encarga de que la transmisión de datos sea fiable y libre de errores entre equipos directamente conectados.
- En esta capa se usan los protocolos Ethernet, WiFi, etc.



## 3.5 CAPA DE RED

- La tercera capa, la de red, se encarga de la **comunicación a través de una red entre equipos no conectados directamente.**
- En esta capa se usa el protocolo IP (ya sea en su versión 4 o en su versión 6 más moderna).

## 3.6 CAPA DE TRANSPORTE

- La cuarta capa, la de transporte, permite transmitir datos de forma óptima y libre de errores entre dos aplicaciones.
- En esta capa se usan 2 protocolos: el TCP (que es más fiable) y el UDP (que es más rápido).

## 3.7 CAPA DE SESIÓN

- La quinta capa, la de sesión, permite indicar puntos en la comunicación para recuperar o continuar en uno de estos puntos.

## 3.8 CAPA DE PRESENTACIÓN

- La sexta capa, la de presentación, define los **formatos de representación de los datos** (caracteres, números, imágenes, etc.)

## 3.9 CAPA DE APLICACIÓN

- La séptima, la de aplicación, es la que se encarga de **dar servicios a los usuarios**. Existen muchos protocolos en esta capa: HTTP, SMTP, DNS, etc.

# **4 EL MODELO TCP/IP**

## 4.1 INTRODUCCIÓN

- El modelo OSI (Open Systems Interconnection Basic Reference Model) es el modelo de referencia para la descripción de las arquitecturas de redes, pero se ha quedado en modelo teórico ya que el modelo TCP/IP se ha convertido en el estándar.
- Este último, al igual que el primero, **se divide en capas, pero esta vez en vez de 7 son solo 4.**

## 4.2 CAPA DE RED

- La primera capa, la de acceso a la red, indica que **hemos de disponer un enlace entre equipos.**
- En esta capa se usan los protocolos Ethernet, WiFi, etc.



## 4.3 CAPA DE INTERNET

- La segunda capa, la capa de Internet, permite la **comunicación de paquetes a través de la red**.
- En esta capa se usa el protocolo IP (ya sea en su versión 4 o en su versión 6 más moderna).

## 4.4 CAPA DE TRASPORTE

- La tercera capa, la de transporte, permite transmitir datos de forma óptima y libre de errores entre dos aplicaciones.
- En esta capa se usan 2 protocolos: el TCP (que es más fiable) y el UDP (que es más rápido).

## 4.5 CAPA DE APLICACIÓN

- La cuarta capa, la de aplicación, es la que se encarga de dar servicios a los usuarios.
- Existen muchos protocolos en esta capa: HTTP, SMTP, DNS, etc.

# **5 FUNCIONAMIENTO TCP/IP**

## 5.1 CABECERA PAQUETE

- Primero se divide el mensaje en paquetes y a cada uno de estos paquetes se le añade una cabecera TCP con información sobre su orden, y un checksum para poder verificar en destino que dicho paquete es correcto.

## 5.2 CABECERA IP

- Acto seguido se le añade la cabecera IP con el remitente y el destinatario del paquete.

## 5.3 ENVÍO

- Luego se envían los paquetes de forma individual por la red hasta el destinatario.

## 5.4 RECEPCIÓN

- El destinatario **reconstruye el mensaje en orden**, y si detecta algún error, vuelve a pedir el reenvío de los paquetes perdidos o dañados.



# 6 TCP VS UDP

## 6.1 EL PROTOCOLO TCP

- El protocolo TCP es **orientado a conexión**, pues crea una sesión entre el origen y el destino.
- Su entrega es confiable, pues se encarga de volver a enviar los datos perdidos o dañados.
- La reconstrucción de los datos se hace de forma ordenada mediante la numeración y secuenciación de segmentos.
- Controla el flujo de datos, pues regula la cantidad de datos que se transmite.
- Es un protocolo con estado, pues realiza un seguimiento de la sesión.

## 6.2 EL PROTOCOLO UDP

- El protocolo UDP no es orientado a conexión.
- Su entrega no es confiable.
- No reconstruye los datos de forma ordenada.
- No hay control de flujo.
- Es un protocolo sin estado.
- Lo bueno que tiene es que su cabecera es mucho más pequeña, y por lo tanto es un protocolo más ligero y rápido.

# **7 CONCEPTOS VARIOS**

## 7.1 DIRECCIÓN IP

- Es una serie de caracteres (numéricos en la versión 4.0 y alfanuméricos en la versión 6.0) separado por puntos, que identifica un ordenador dentro de una red.

## **7.2 COMANDO IPCONFIG**

- Comando de Windows para saber nuestra IP.

## 7.3 COMANDO IFCONFIG

- Comando de Linux para saber nuestra IP.

## 7.4 SERVIDOR DNS

- Un servidor DNS, convierte los nombres de los recursos en direcciones IP.



## 7.5 DIRECCIÓN MAC

- Es la dirección física de un dispositivo y viene puesta de fábrica.

## 7.6 SERVIDOR DHCP

- Asigna una dirección IP para cada dirección MAC de cada uno de los dispositivos conectados a un red, .

## 7.7 COMANDO PING

- El comando ping funciona mandando paquetes a un dispositivo remoto, el cual responde con un eco, de tal forma que **cuando recibimos el eco, sabemos con certeza que hay conectividad** entre la máquina que ha hecho ping y el dispositivo remoto.
- Además calculando el tiempo que tarda en retornar el eco, podemos calcular la distancia al dispositivo remoto.

## 7.8 PROTOCOLO HTTP

- El protocolo HTTP es el de transferencia de hipertexto (en inglés: Hypertext Transfer Protocol o HTTP) y permite las transferencias de información en la World Wide Web. Es un protocolo sin estado, es decir, no guarda ninguna información sobre conexiones anteriores, por ello es necesario el uso de cookies, que se guardan en el sistema cliente, y permiten guardar información entre peticiones diferentes.