

# COMUNICACIÓN HTTP

Modelo de comunicación HTTP

Vera Valles, Joaquin  
Joaquin\_veravalles@hotmail.com

## Contents

1.- El protocolo HTTP .....	2
2 Modelo de dos capas .....	3
3. Modelo de tres capas .....	4
4. Modelo de 4 capas. ....	5
5. Los proxies .....	6
6. TCP y UDP .....	7
7.- Fuentes.....	8

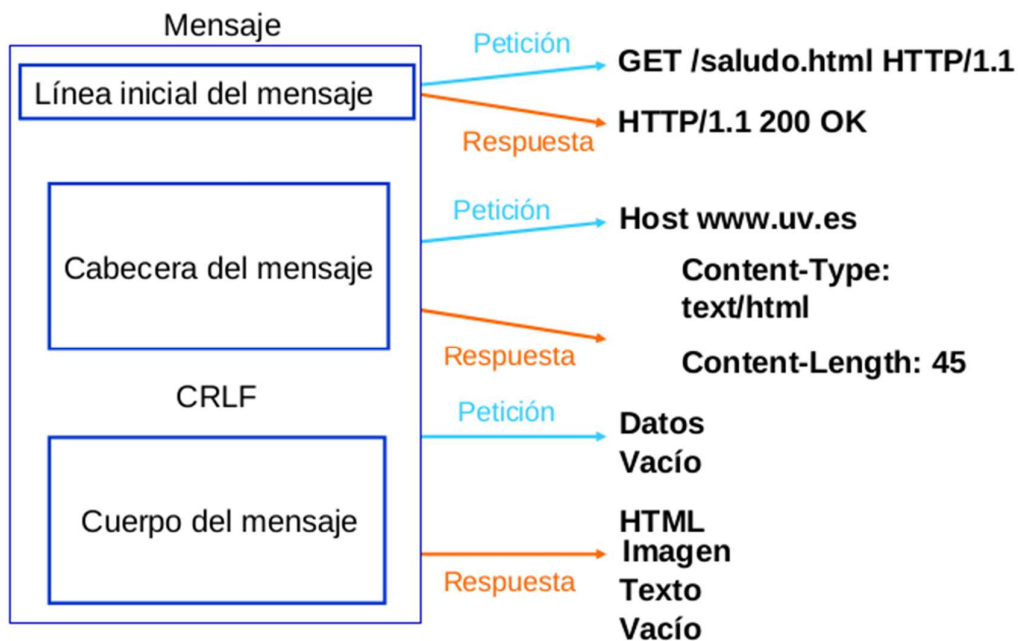
## 1.- El protocolo HTTP

HTTP, de sus siglas en inglés: "Hypertext Transfer Protocol", es el nombre de un protocolo el cual nos permite realizar una petición de datos y recursos, como pueden ser documentos HTML. Es la base de cualquier intercambio de datos en la Web, y un protocolo de estructura cliente-servidor, esto quiere decir que una petición de datos es iniciada por el elemento que recibirá los datos (el cliente), normalmente un navegador Web.

HTTP es un protocolo basado en el principio de cliente-servidor: las peticiones son enviadas por una entidad: el agente del usuario (o un proxy a petición de uno). La mayoría de las veces el agente del usuario (cliente) es un navegador Web, pero podría ser cualquier otro programa, como por ejemplo un programa-robot, que explore la Web, para adquirir datos de su estructura y contenido para uso de un buscador de Internet.

Cada petición individual se envía a un servidor, el cuál la gestiona y responde. Entre cada petición y respuesta, hay varios intermediarios, normalmente denominados proxies, los cuales realizan distintas funciones, como: gateways o caches.

El navegador es siempre el que inicia una comunicación (petición), y el servidor nunca la comienza (hay algunos mecanismos que permiten esto, pero no son muy habituales).



Dependiendo del modelo de la estructura de la página Web, el tipo de conexiones y la cantidad de conexiones cambian:

## 2 Modelo de dos capas

Es uno de los modelos más fáciles de implementar.

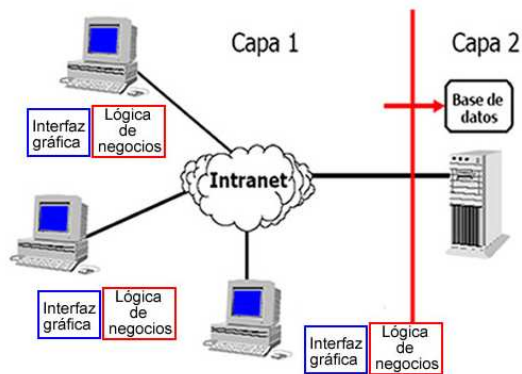
La mayor parte de la lógica de este modelo de arquitectura corre en la parte del cliente (fat client).

Las 2 capas son:

- Cliente: La lógica de negocio está inmersa dentro de la aplicación. Esta aplicación se realiza en la interfaz de cliente, es decir que la lógica de negocio corre parte del cliente.
- Servidor: Es la parte que gestiona los datos y que los contiene. Suele ser una base de datos normal y corriente.

Este modelo de aplicaciones Web tiene una serie de desventajas:

- Es difícilmente escalable
- Número de conexiones reducida
- Alta carga de la red.
- La flexibilidad es restringida
- La funcionalidad es limitada.



### 3. Modelo de tres capas

Es una arquitectura diseñada para superar las dificultades de las arquitecturas de 2 capas, es decir, que es mucho más potente que las arquitecturas de 2 capas.

La diferencia en estos modelos es que no cae tanta funcionalidad en el cliente y se crea una capa intermedia (Capa de servidor) entre la capa de cliente y la capa de datos.

Ahora, el modelo 3 capas, las capas quedarían con las siguientes funcionalidades:

1. Capa de presentación (parte en el cliente y parte en el servidor)
  - Recoge la información del usuario y la envía al servidor (cliente)
  - Manda información a la capa de proceso para su procesamiento
  - Recibe los resultados de la capa de proceso
  - Generan la presentación
  - Visualizan la presentación al usuario (cliente)
2. Capa de proceso (servidor web)
  - Recibe la entrada de datos de la capa de presentación
  - Interactúa con la capa de datos para realizar operaciones
  - Manda los resultados procesados a la capa de presentación
3. Capa de datos (servidor de datos)
  - Almacena los datos
  - Recupera datos
  - Mantiene los datos
  - asegura la integridad de los datos



#### 4. Modelo de 4 capas.

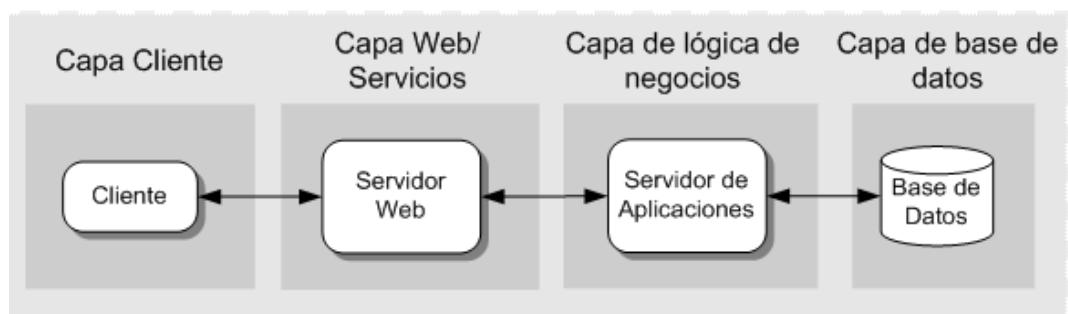
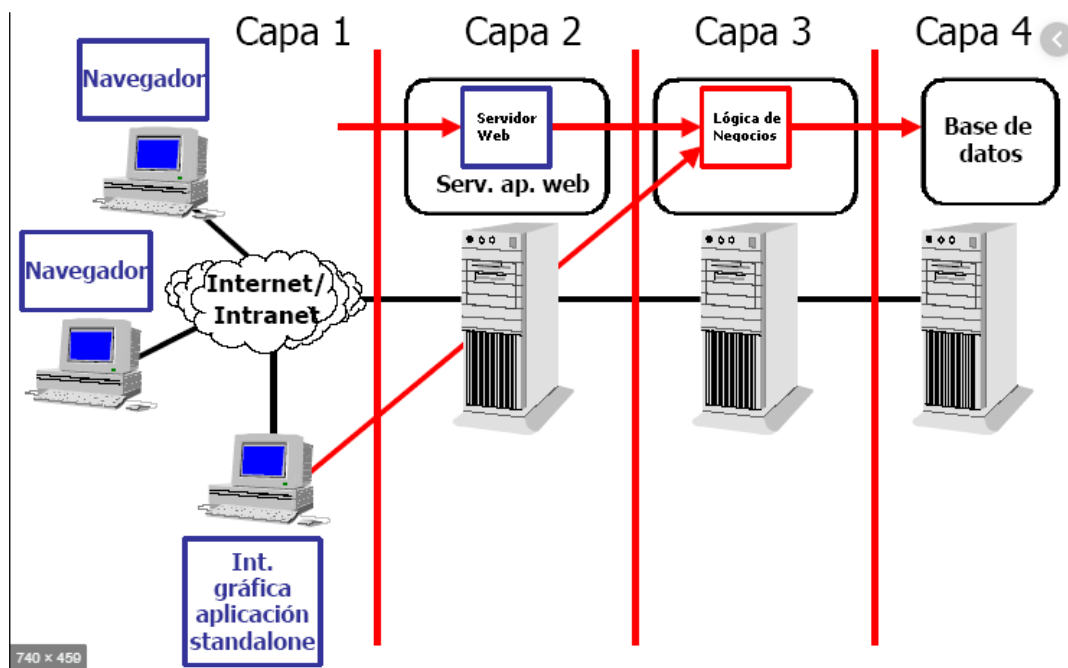
Es una arquitectura que solo es utilizada en las empresas más grandes. Es el modelo más difícil y costoso de implementar pero es el mejor preparado y es el más escalable y funcional.

Ahora, en la capa de cliente solo se queda la interfaz gráfica y la ejecución de códigos en cliente (Visualizaciones, etc...)

La capa de Servidor ahora se divide en 2 partes y se crea la capa de lógica de negocios, anteriormente gestionada por el servidor.

Los clientes pueden llamar a la parte del servidor Web cuando necesite una interfaz o a la capa de lógica de negocios cuando intente algo de la lógica de negocios.

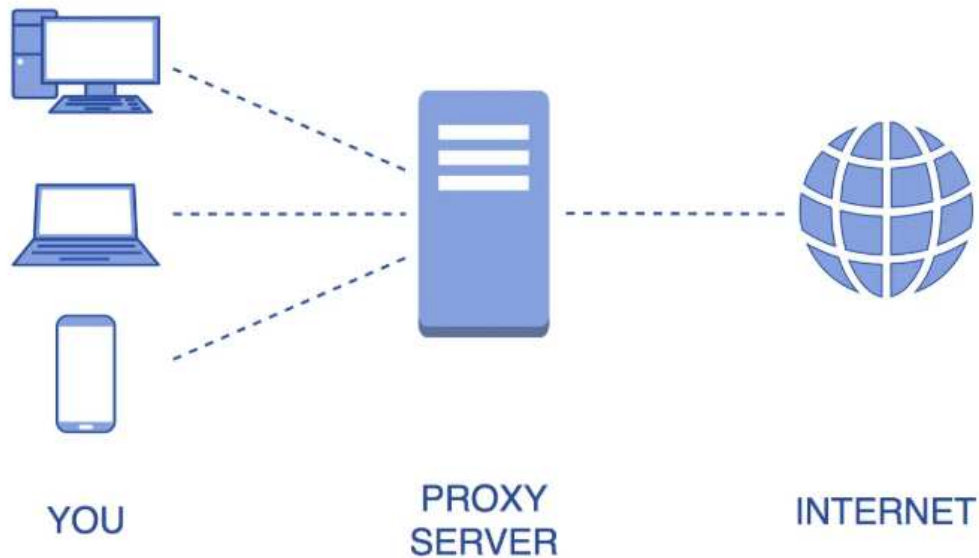
El servidor también puede llamar a la capa de negocios.



## 5. Los proxies

Dada la arquitectura en capas de la Web, la mayoría de estos dispositivos solamente gestionan estos mensajes en los niveles de protocolo inferiores: capa de transporte, capa de red o capa física, siendo así transparentes para la capa de comunicaciones de aplicación del HTTP, además esto aumenta el rendimiento de la comunicación. Aquellos dispositivos, que sí operan procesando la capa de aplicación son conocidos como proxies. Estos pueden ser transparentes, o no (modificando las peticiones que pasan por ellos), y realizan varias funciones:

- caching (la caché puede ser pública o privada, como la caché de un navegador)
- filtrado (como un anti-virus, control parental, ...)
- balanceo de carga de peticiones (para permitir a varios servidores responder a la carga total de peticiones que reciben)
- autenticación (para el control al acceso de recursos y datos)
- registro de eventos (para tener un histórico de los eventos que se producen)

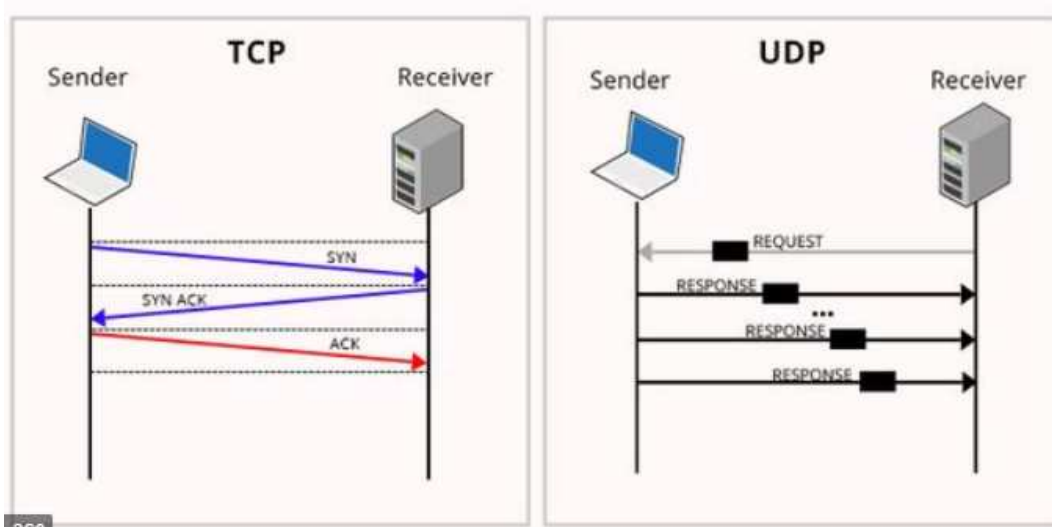


## 6. TCP y UDP

Una conexión de TCP se establece en un vínculo de 3 tiempos ("3-way handshake"), que es un proceso de iniciar y verificar una conexión. Una vez se establece la conexión, puede comenzar la transferencia de data. Luego de la transferencia, la conexión se termina cerrando los circuitos virtuales establecidos.

UDP usa un modelo sencillo de transmisión sin vínculo ("handshake") implícito que garantice confiabilidad, orden o integridad de data. Por ello, UDP provee un servicio poco confiable y la data puede llegar fuera de orden, duplicada o incompleta sin que haya aviso. UDP asume que las correcciones y verificación de errores o no es necesario o se hace por medio de la aplicación activa, evitando usar recursos a nivel de la interfase de la red misma. A diferencia de TCP, UDP es compatible con transmisiones amplias de paquetes (envíos a todos los puntos de una red local) y multi-envíos (enviados a todos los suscriptores.)

En resumen, TCP implica Seguridad y UDP implica velocidad.





## 7.- Fuentes

<https://es.diffen.com/tecnologia/TCP-vs-UDP>

<http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/http.html>

<https://developer.mozilla.org/es/docs/Web/HTTP/Overview>