

UT 1: Arquitectura Web



2ºDAW – Despliegue Aplicaciones Web

Introducción

Arquitectura cliente-servidor

El modelo de desarrollo web se apoya, en una primera aproximación desde un punto de vista centrado en el hardware, en lo que se conoce como arquitectura **cliente-servidor** ¹⁾ que define un patrón de arquitectura donde existen dos actores, cliente y servidor, de forma que **el primero es quién se conecta con el segundo para solicitar algún servicio.**

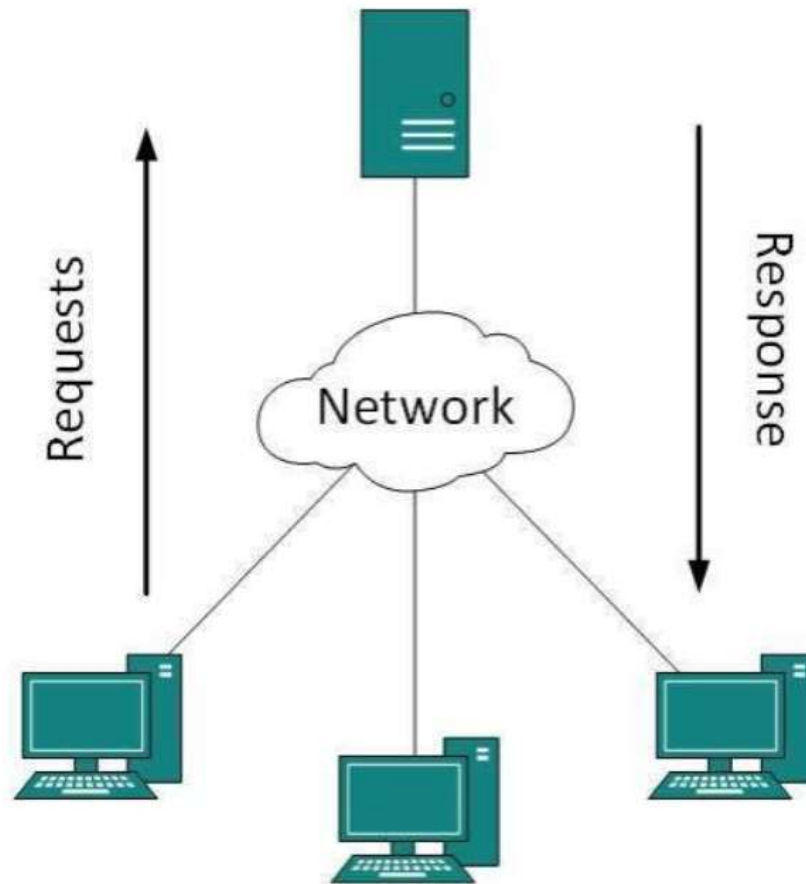
En el caso que nos ocupa, el desarrollo web, los **clientes solicitan** que se les sirva una **web para visualizarla**, aunque también es posible solicitar información si hablamos del caso de los servicios web que también veremos más adelante. En cualquier caso, en ambos casos aparece el mismo escenario, donde un **servidor** se encuentra **ejecutándose ininterrumpidamente a la espera** de que los diferentes clientes realicen una solicitud.

Normalmente a la **solicitud** que hacen los clientes **al servidor** se le llama **petición (request)** y a lo que el **servidor devuelve** a dicho cliente le llamamos respuesta **Response**

También hay que tener en cuenta que esta arquitectura cliente-servidor plantea la posibilidad de **numerosos clientes** atendidos por un **mismo servidor.**

Es decir, el servidor será un software multitarea que será capaz de atender peticiones simultáneas de numerosos clientes.

Introducción



Esta arquitectura básica de cliente y servidor se conoce como **modelo de 2 capas**.

Figure 1: Arquitectura cliente-servidor

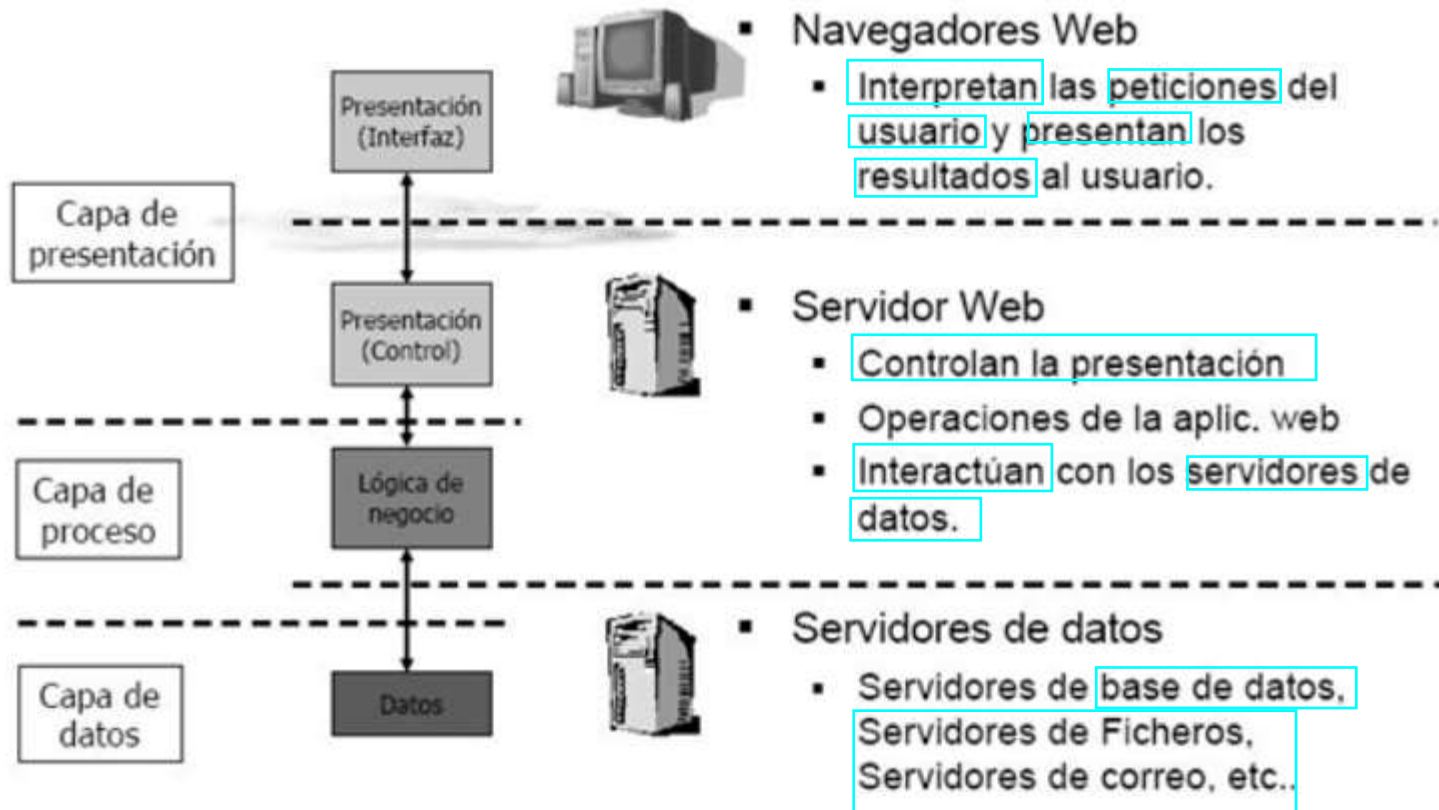
Introducción

Desde un punto de vista de desarrollo una aproximación más detallada para este modelo de ejecución es lo que se conoce como **modelo en 3 capas** ²⁾. Es un modelo donde se muestra más en detalle como se distribuye el software que participa en cualquier desarrollo web.

Sigue estando presente la arquitectura cliente-servidor (todo se basa en ella) pero aparecen más detalles como el software utilizado en cada uno de los dos actores y como interactúan las diferentes tecnologías o aplicaciones.

Es un tipo de arquitectura usada en la gran mayoría de sistemas. Se suele usar en sistemas que implementan un **modelo de negocio** como podría ser una tienda online, una aplicación para gestionar ciertos datos, etc. **En la arquitectura en tres niveles existe un nivel intermedio.** Esto significa que la arquitectura generalmente está compartida por:

Introducción



Arquitecturas de Red

Protocolos

La pila de protocolos en los que se basa Internet es muy amplia, ya sea siguiendo el **modelo OSI o el modelo TCP**. En cualquier caso, en el tema que nos ocupa a nosotros sólo nos fijaremos en la última capa, la **capa de transporte**.

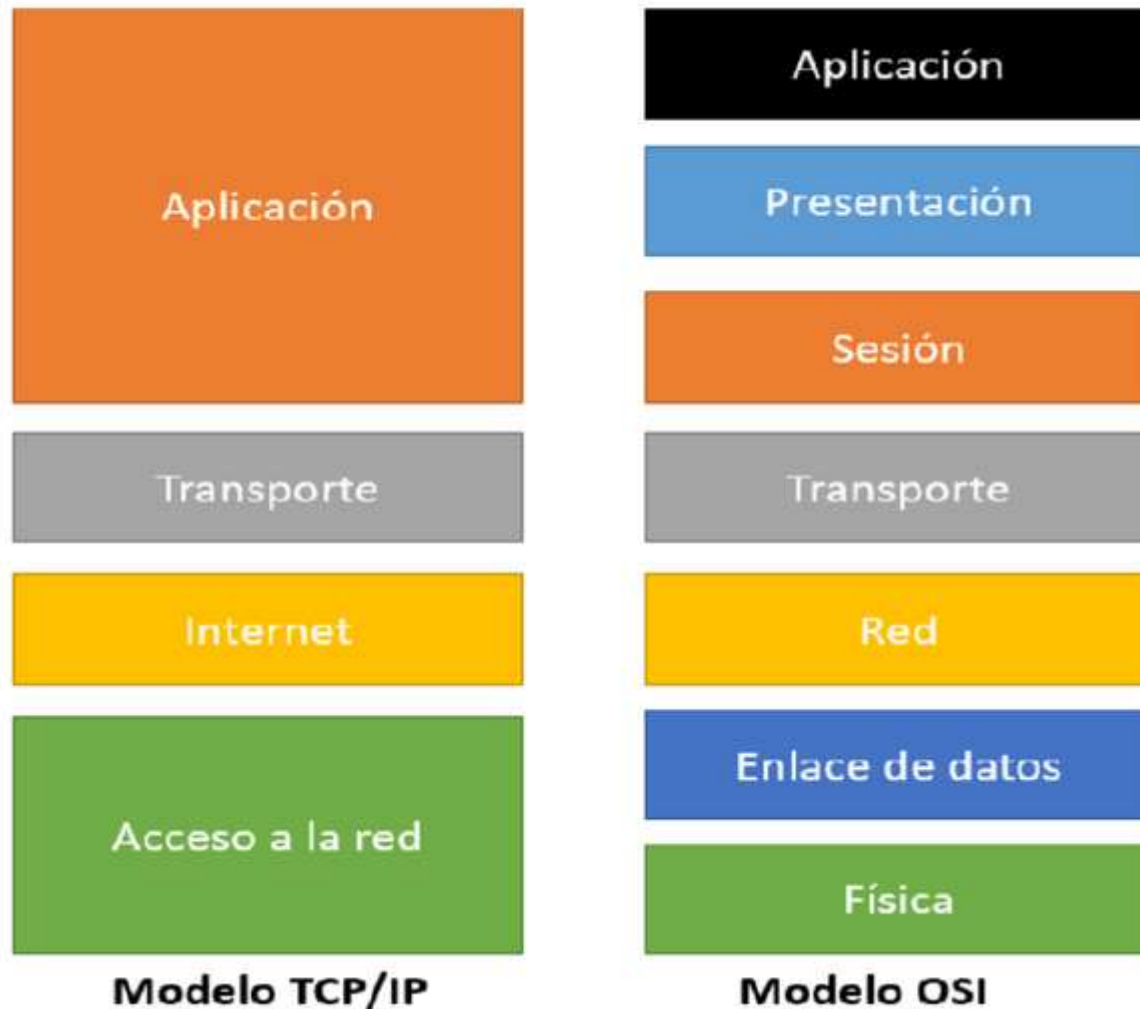
Es en esta capa donde **están los protocolos de la web**, **los que usan navegadores y servidores (web y aplicaciones) para comunicarse**.

Al fin y al cabo, la web no es más que una de las tantas aplicaciones que existen en Internet.

Otras aplicaciones, que también veremos en este curso, son FTP y SSH, entre otras.

Arquitecturas de Red

Modelos

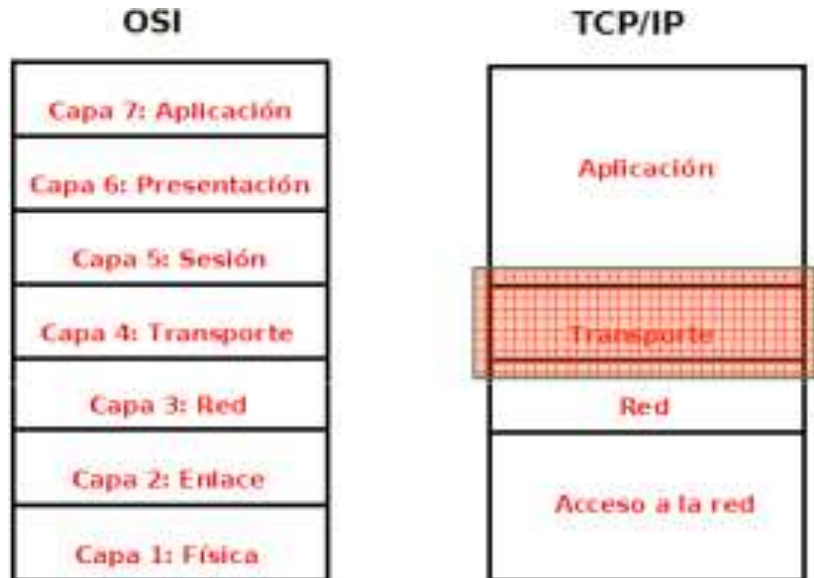


Arquitectura TCP/IP

Nivel de Transporte

Protocolos

- Protocolo TCP y UDP
- Permite diferenciar aplicaciones dentro de un mismo equipo (host).
- Concepto de Puerto
- Funciones adicionales: segmentación de datos, control de errores, control de flujo, QoS, ...



Arquitectura TCP/IP

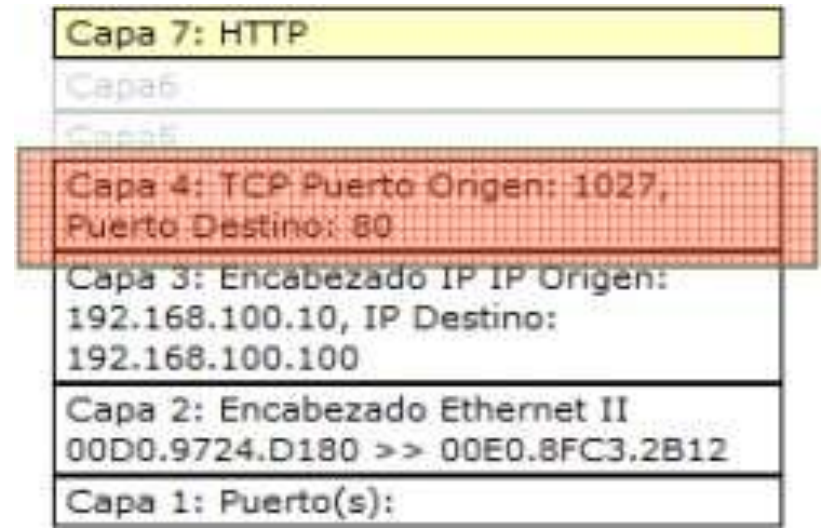
Nivel de Transporte

Puertos

- Números enteros positivos (16 bits) (0-65535) que identifican procesos de un equipo que envían y reciben información a través de la red.
- Puertos bien conocidos conocidos (“well-known ports”): 0 al 1023
- Puertos registrados (1024 - 49151)
- Puertos dinámicos (49152 - 65535)

Asignación de puertos

- Estática: definidos en configuración de la aplicación.
- Dinámica:
Sistema operativo.
Puertos disponibles



Arquitectura TCP/IP

Nivel de Transporte – Protocolo UDP

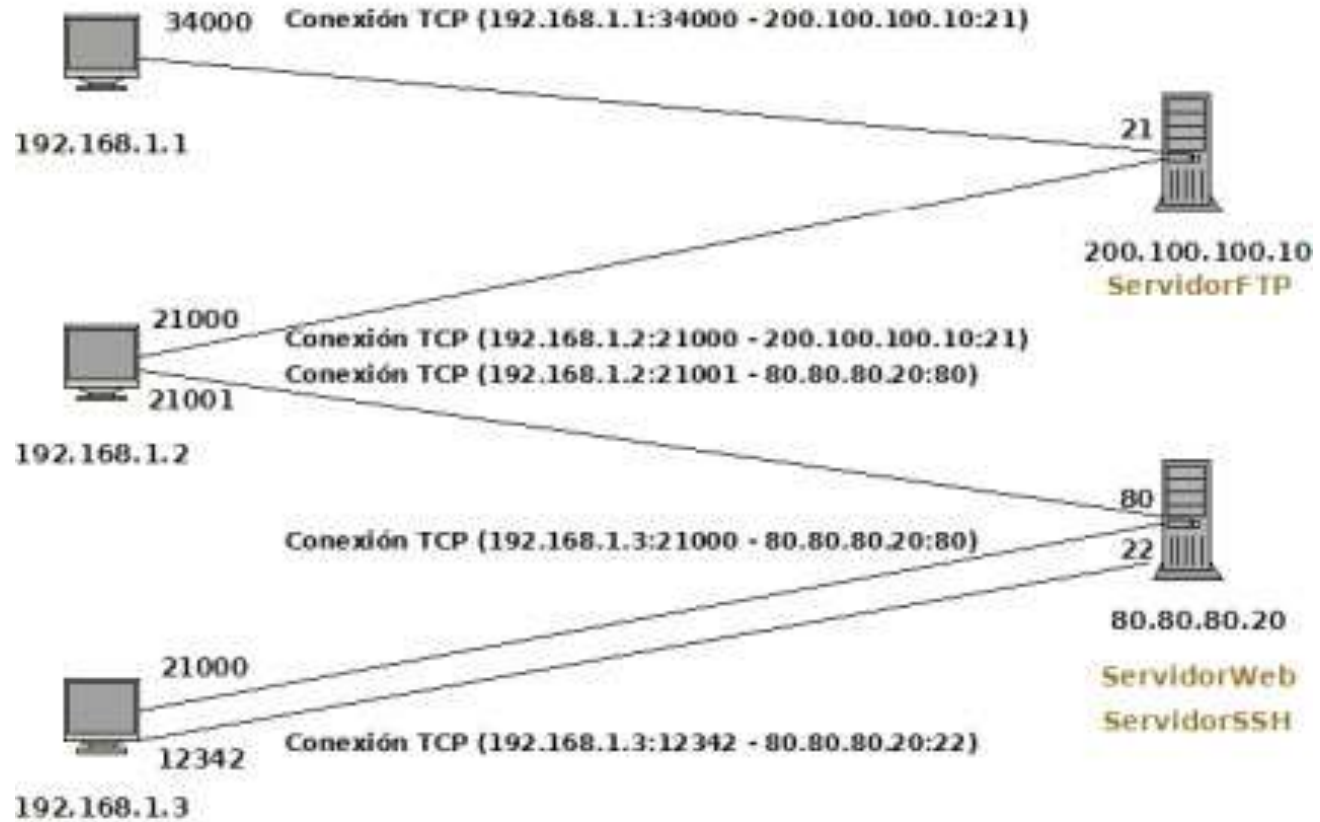
- No orientado a conexión.
No hay conexiones.
No hay establecimiento de conexión.
- No fiable -> No realiza control de errores.
- Envío de datos más rápido que TCP.
- Envío de datos más rápido que TCP.
- Datagramas UDP.

Nivel de Transporte – Protocolo TCP

- Orientado a conexión.
Conexiones.
Establecimiento y finalización de conexiones
- Fiable: Control de errores, Control de flujo, Control de congestión ...
- Segmentos TCP

Arquitectura TCP/IP

Nivel de Transporte – Protocolo TCP

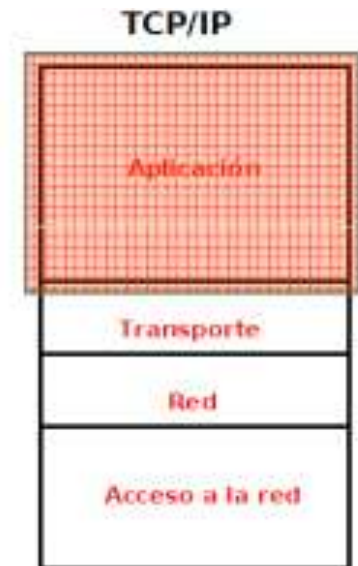
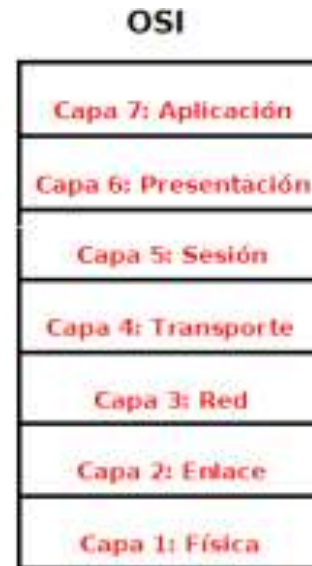


Arquitectura TCP/IP

Nivel de Aplicación


Ofrece servicios de red a los usuarios

- Modelo de funcionamiento/comunicación
 Cliente/Servidor - P2P (Peer To Peer) - Híbrido.
- Aplicaciones:
 Clientes.
 Servidores.
- Protocolos: HTTP, FTP, DNS,
 DHCP, SSH, SMTP, ...



Arquitecturas de Red

Protocolos

Puesto que en esta asignatura nos centramos en la web y, aunque en menor medida, en los protocolos (de aplicación) que de alguna manera son de utilidad para trabajar con ella, nos centraremos exclusivamente en ellos 

- **HTTP:** HyperText Transfer Protocol. Protocolo de comunicación para la web
- **HTTPS:** HTTP Secure. Protocolo seguro de comunicación para la web. Surge de aplicar una capa de seguridad, utilizando SSL/TLS, al protocolo HTTP
- **Telnet:** Es un protocolo que establece una línea de comunicación basada en texto entre un cliente y un servidor. Desde su aparición se utilizó ampliamente como vía de comunicación remota con el sistema operativo ya que permitía la ejecución remota de comandos. Con el tiempo ha ido cayendo en desuso a favor de un protocolo seguro que lo sustituye, SSH.
- **SSH:** Secure Shell. Protocolo seguro de comunicación ampliamente utilizado para la gestión remota de sistemas, ya que permite la ejecución remota de comandos. Surge como reemplazo para el protocolo no seguro Telnet.
- **SCP:** Secure Copy. Es un protocolo seguro (basado en RCP, Remote Copy) que permite transferir ficheros entre un equipo local y otro remoto o entre dos equipos remotos. Utiliza SSH por lo que garantiza la seguridad de la transferencia así como de la autenticación de los usuarios.
- **FTP:** File Transfer Protocol. Es un protocolo que se utiliza para la transferencia de archivos entre un equipo local y otro remoto. Su principal problema es que tanto la autenticación como la transferencia se realiza como texto plano, por lo que se considera un protocolo no seguro.
- **SFTP:** SSH FTP. Es una versión del protocolo FTP que utiliza SSH para cifrar tanto la autenticación del usuario como la transferencia de los archivos. Es la opción segura al uso de un protocolo como FTP



Arquitecturas de Red

Protocolos

Puesto que en esta asignatura nos centramos en la web y, aunque en menor medida, en los protocolos (de aplicación) que de alguna manera son de utilidad para trabajar con ella, nos centraremos exclusivamente en ellos:



Figure 3: Protocolos en Internet (Fuente:Wikipedia)

Arquitecturas de Red

Protocolos

Protocolo HTTP

El protocolo HTTP es un protocolo para la transferencia de páginas web (hipertexto) entre los clientes (navegadores web) y un servidor web. Cuando un usuario, a través del navegador, quiere un documento (página web), éste lo solicita mediante una petición HTTP al servidor.

Éste le contestará con una respuesta HTTP y el documento, si dispone de él.

Hay que tener en cuenta que, al contrario que el resto de los protocolos que estamos viendo en esta parte, HTTP no tiene estado.

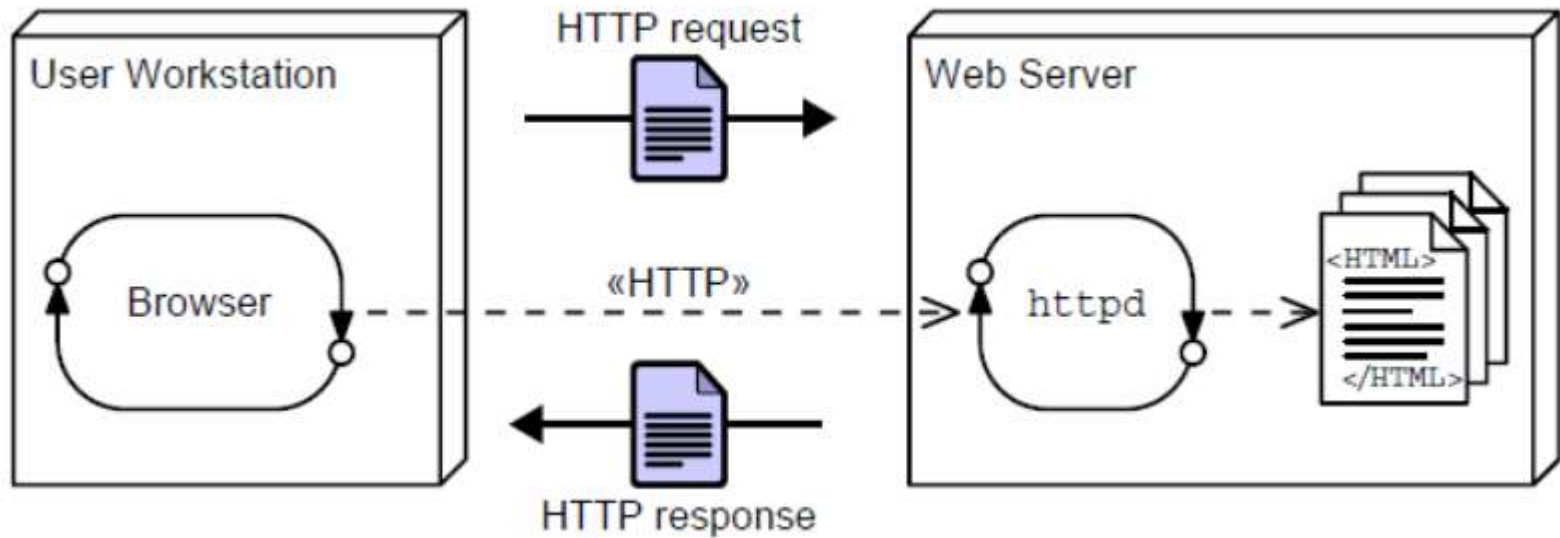
Eso significa que un servidor web no almacena ninguna información sobre los clientes que se conectan a él. Así, cada petición/respuesta supone una conexión única y aislada.

En cualquier caso, utilizando tecnologías en el lado servidor es posible escribir aplicaciones web que puedan establecer sesiones o cookies para almacenar ese estado y “recordar” de alguna manera a los clientes en sucesivas conexiones

Arquitecturas de Red

Protocolos

Protocolo HTTP



Arquitecturas de Red

Protocolos

Protocolo HTTP

A continuación, a modo de ejemplo, podemos ver una **petición HTTP** que un **navegador (Firefox)** ha realizado a un sitio web (*misitio.com*), solicitando el documento *index.html*.

```
GET /index.html HTTP/1.1
Host: www.misitio.com
User-Agent: cliente
Referer: www.google.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Connection: keep-alive
[Línea en blanco]
```

Arquitecturas de Red

Protocolos

Protocolo HTTP

Y el **servidor web le contesta** con el contenido del documento para que el **navegador** que lo ha solicitado lo pueda **renderizar para que el usuario** lo visualice en su pantalla:

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2003 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221

<html lang="es">
<head>
<meta charset="utf-8">
<title>Mi título</title>
</head>
<body>
<h1>Bienvenido a mi sitio.com</h1>
. . .
. . .
</body>
</html>
```

Arquitecturas de Red

Protocolos

Protocolo **SSL/TLS**

!MPA

SSL (Secure Sockets Layer) y TLS (Transport Layer Security) son protocolos de cifrado que se utilizan para cifrar las comunicaciones en Internet. En ocasiones se hace referencia en ambos casos al uso de **SSL**, pero la realidad es que **TLS** es el sucesor de **SSL** debido a las diferentes vulnerabilidades que han ido surgiendo de este último.

En este caso, la aplicación de esta capa de seguridad, cifrando las comunicaciones del protocolo HTTP, da lugar a lo que se conoce como **HTTPS**, que veremos a continuación.



Arquitecturas de Red

Protocolos

Protocolo SSL/TLS

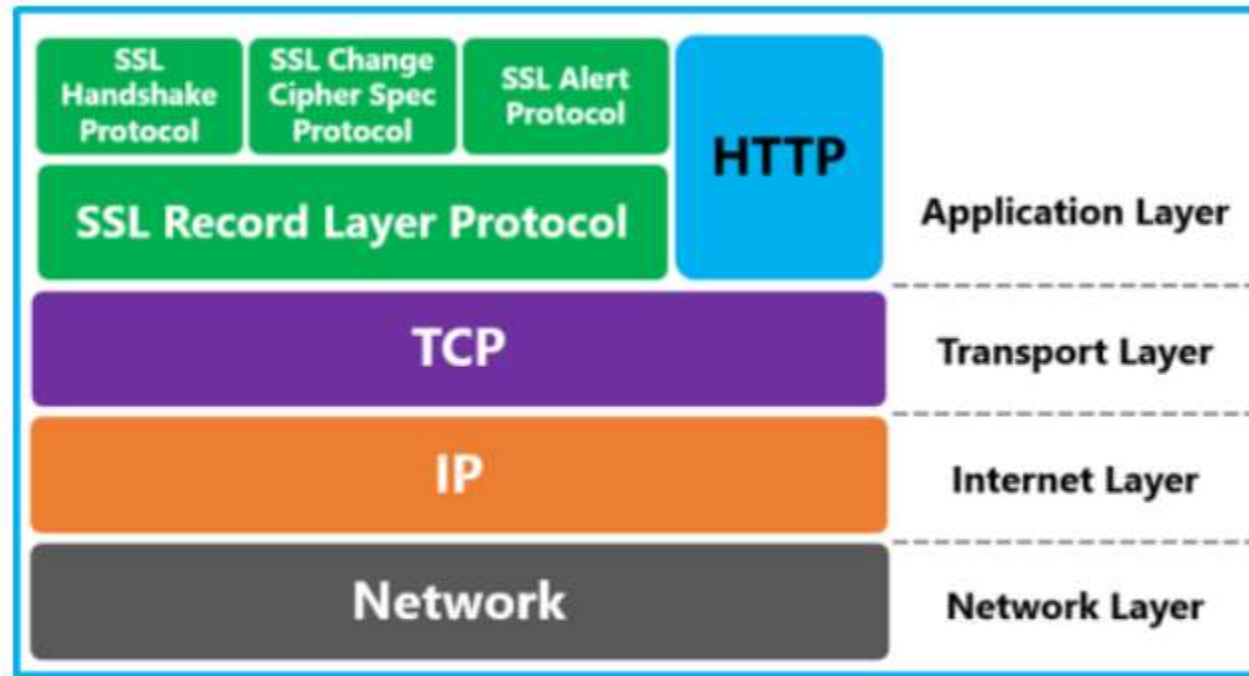


Figure 6: SSL/TLS en el modelo TCP

Arquitecturas de Red

Protocolos

Protocolo HTTPS

El protocolo HTTPS (HTTP Secure) es un protocolo de comunicación segura a través de Internet. Este protocolo se basa en la comunicación del protocolo HTTP pero con una capa de seguridad adicional cifrando el contenido con TSL ó SSL.

Su principal utilidad es el cifrado de los mecanismos de autenticación en la web, justamente cuando el usuario envía sus credenciales al servidor para validar su sesión. Es el momento más crítico en una comunicación, aunque actualmente se está utilizando abiertamente durante toda la comunicación entre cliente y servidor en la web por privacidad e integridad.

Con respecto a la integridad, HTTPS proporciona un mecanismo de autenticación con respecto al sitio web y servidor web que estamos visitando, evitando así ataques como el del *Man in the Middle*. Es la manera en la que podemos estar seguros de que el sitio con el que estamos comunicandonos es el que creemos que es.

Arquitecturas de Red

Protocolos

Protocolo HTTPS

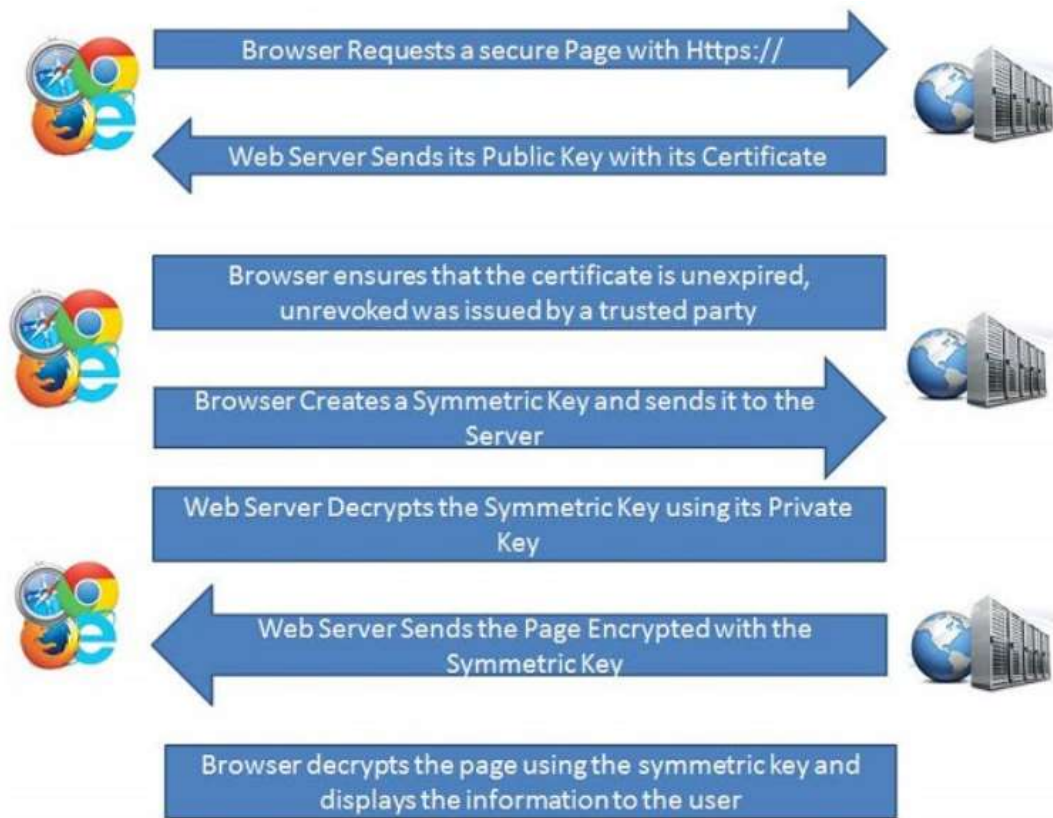


Figure 7: Protocolo HTTPS

Arquitecturas de Red

Protocolos

Protocolo SSH y SCP

Protocolo SSH

El protocolo SSH (**S**ecure **S**hell), al igual que Telnet, proporciona una forma de comunicación entre dos máquinas remotas, basada en texto pero en este caso tanto la autenticación como la propia comunicación se encuentran cifradas. Su uso más habitual está en la gestión remota de máquinas Unix/Linux, tal y como ocurriría en su momento con Telnet.

Además, otros protocolos como FTP (por eso llamado SFTP) o SCP se apoyan en este protocolo de cifrado para cifrar sus comunicaciones.

Protocolo SCP

El protocolo SCP (**S**ecure **C**oPy), basado en el protocolo SSH, permite copiar ficheros entre dos equipos, tanto del equipo local al remoto como en el sentido contrario.

Arquitecturas de Red

Protocolos

Protocolo FTP

El protocolo **FTP (File Transfer Protocol)** es un protocolo para la transferencia de ficheros entre dos máquinas: una máquina cliente y otra máquina remota o servidor donde se alojan dichos ficheros. La idea es que la máquina remota sirva como repositorio de información y sean los múltiples clientes los que se conectan a ella para coger o subir ficheros.

Protocolo SFTP

El protocolo **SFTP (SSH FTP)** es la versión segura del protocolo **FTP**. Esta vez, a diferencia de lo que ocurre con el protocolo **FTP**, las comunicaciones se cifran de la misma forma que en el protocolo **SSH**.

