



# Introducción al Desarrollo Web

Ing. Marco Aedo López

# PROTOSCOLOS



## Tema 3

***PRESENTADO POR:***  
***Ing. Marco Aedo López***

**Contacto:**

maedol@unsa.edu.pe

marcoaedo@hotmail.com

**Blog:**

du tic.unsa.edu.pe

marcoaedo.blogspot.com

# Objetivos

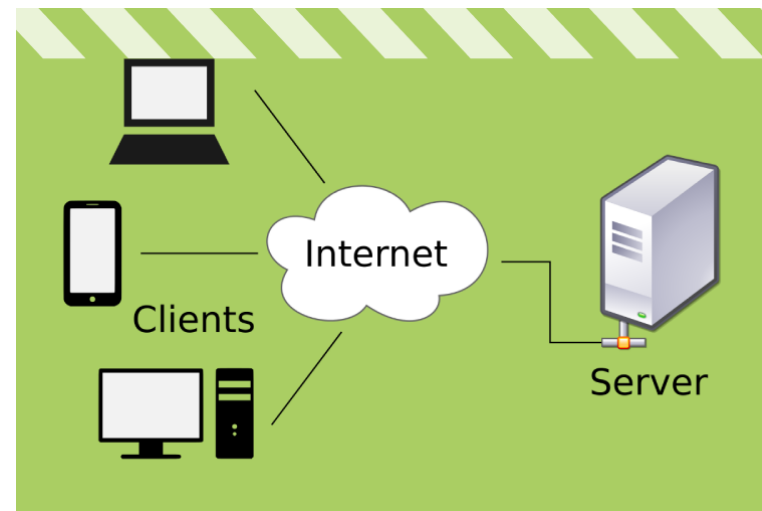
- Entender qué es un protocolo de red
- Comprender los principales protocolos de Internet
- Reconocer componentes clave como URL, direcciones IP y servidores

# Contenido

1. Protocolos
2. HTTP
3. HTTPS
4. TCP/IP
5. URL
6. Servidor DNS
7. SSH
8. FTP
9. Servidor WEB
10. Directorio public\_html
11. Cliente WEB

# 1.1. ¿Qué es un protocolo de red?

- Un protocolo de red es un conjunto de reglas o normas que permiten la comunicación y el intercambio de datos entre dispositivos conectados a una red
- Definen cómo se estructuran, envían y reciben los datos



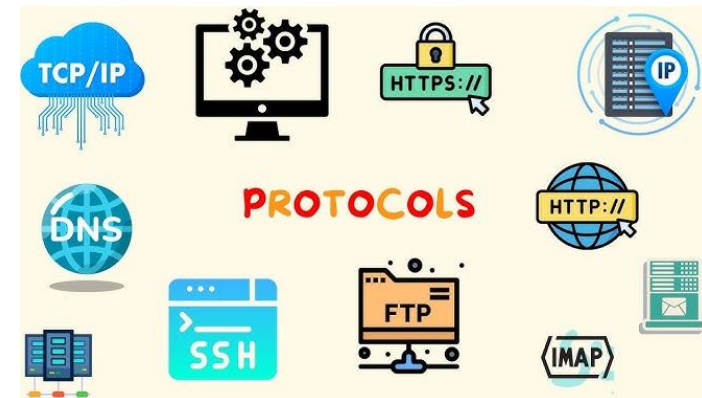
## 1.2. Características generales de los protocolos

- Estandarizados por organismos internacionales (como IETF)
- Permiten la interoperabilidad entre diferentes sistemas
- Pueden ser orientados a conexión (ej. TCP) o no orientados a la conexión (ej. UDP)
- Incluyen funciones como control de errores, cifrado y autenticación



## 1.3. Ejemplos de protocolos de Internet

- HTTP/HTTPS: Transferencia de páginas web (Hipertexto)
- TCP/IP: Transmisión confiable de datos
- FTP: Transferencia de archivos
- SSH: Acceso seguro a sistemas remotos
- DNS: Resolución de nombres de dominio





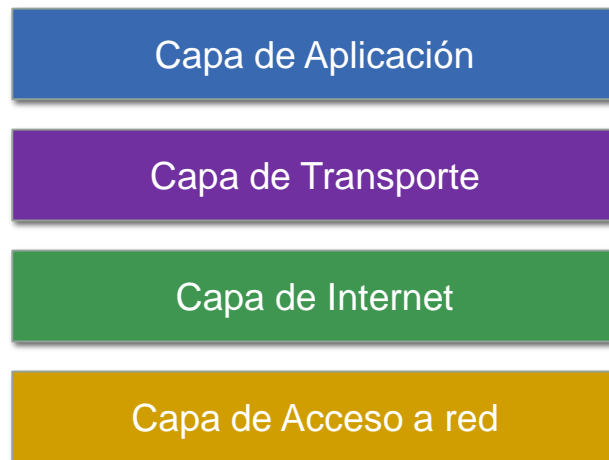
## 1.4. Importancia de los protocolos

- Gracias a los protocolos, Internet funciona como un sistema global interconectado
- Permiten que dispositivos de diferentes fabricantes y plataformas se comuniquen de manera segura, eficiente y estandarizada

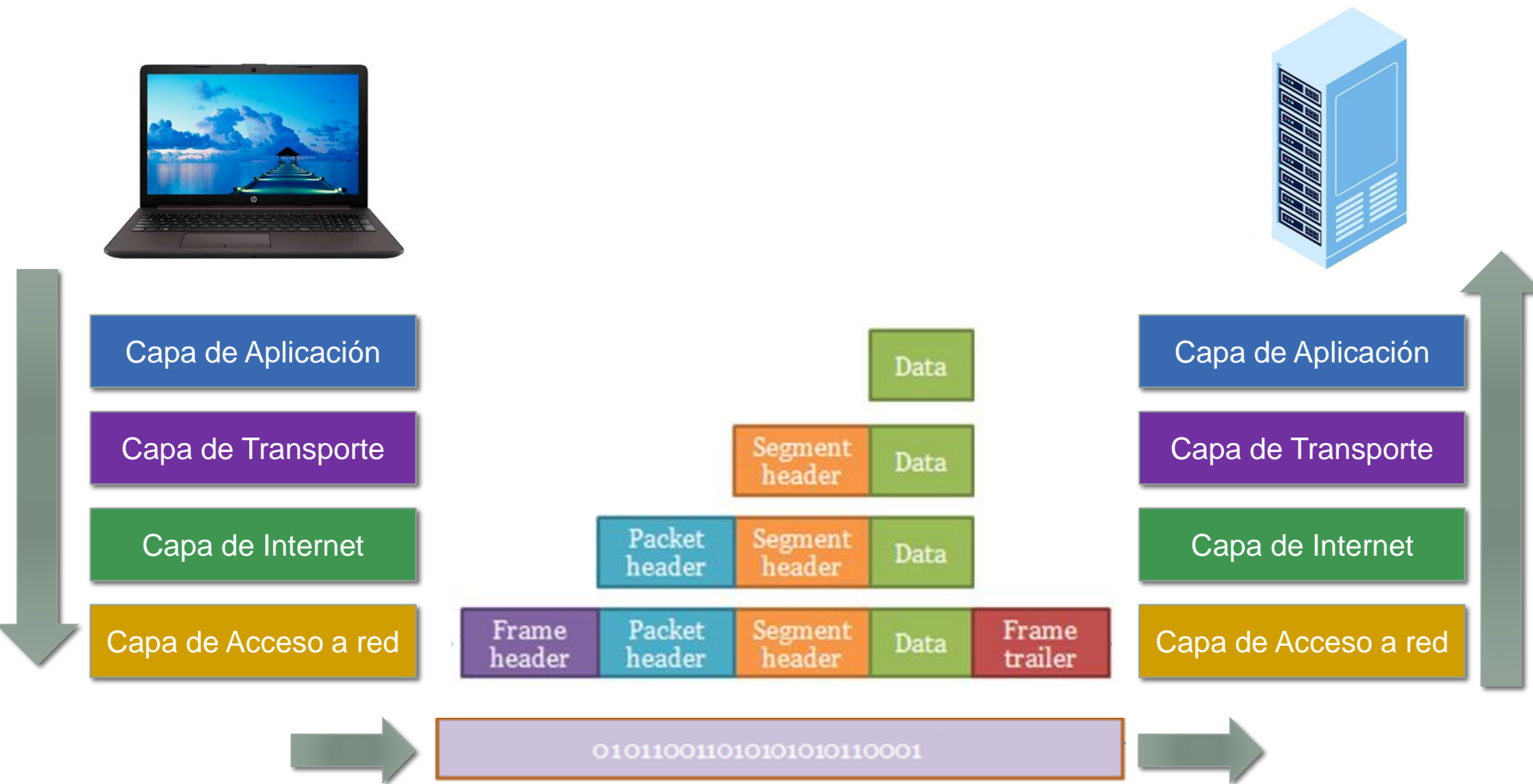


## 1.5. ¿Qué es el modelo TCP/IP?

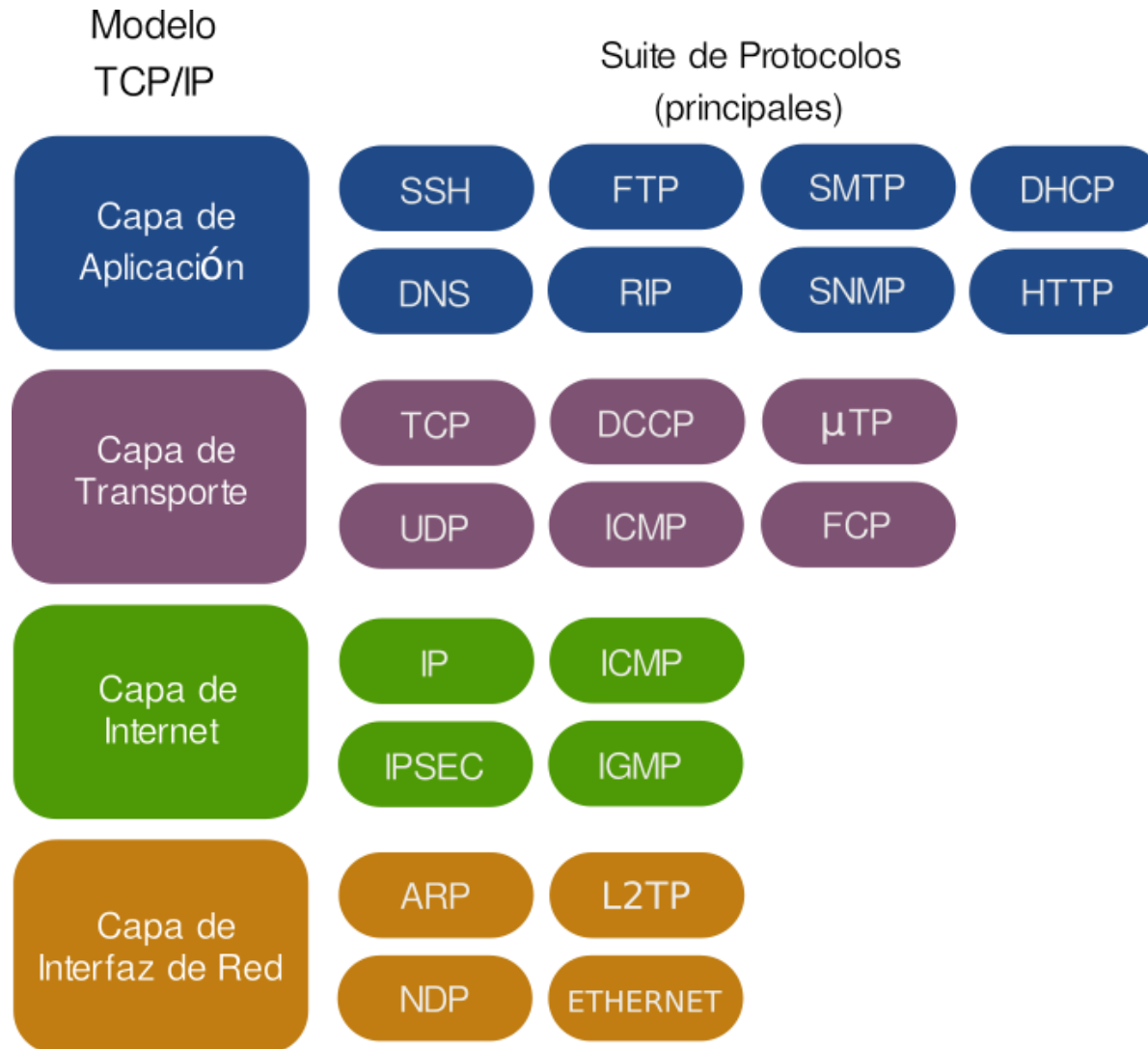
- Es un conjunto de protocolos que define cómo los dispositivos se comunican en Internet
- Tiene 4 capas principales: Aplicación, Transporte, Internet (Red) y Acceso a red (Enlace de datos)




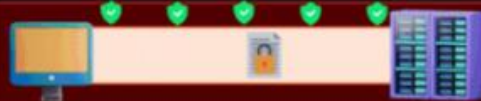






# 1.6. Estructura del modelo TCP/IP



# 1.6. Estructura del modelo TCP/IP

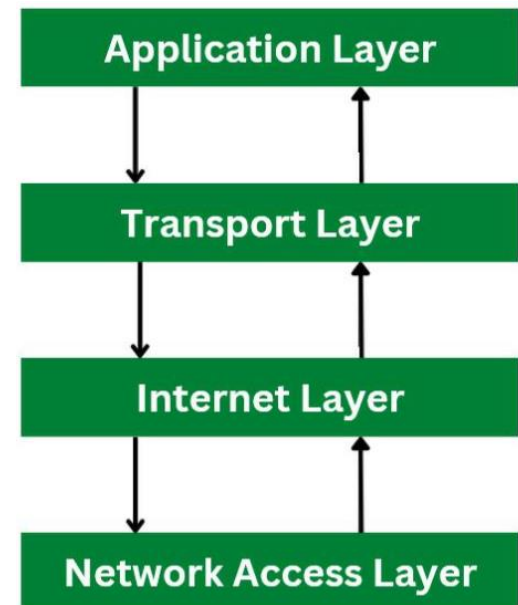


# Networking Protocols Explained

| Protocol  | Illustration   | Description   |
|---|--|---|
| <b>HTTP</b><br>Hyper Text Transfer Protocol         |    | <ul style="list-style-type: none"> <li>• Web transfer protocol</li> <li>• Request-response</li> <li>• Not Secure</li> </ul> |
| <b>HTTPS</b><br>Hyper Text Transfer Protocol Secure |    | <ul style="list-style-type: none"> <li>• Secure communication</li> <li>• Uses TLS/SSL encryption</li> </ul>                 |
| <b>FTP</b><br>File Transfer Protocol                |    | <ul style="list-style-type: none"> <li>• File transfer protocol</li> <li>• Transfers files between computers</li> </ul>     |
| <b>TCP</b><br>Transmission Control Protocol         |    | <ul style="list-style-type: none"> <li>• Reliable communication</li> <li>• Connection-oriented</li> </ul>                   |
| <b>IP</b><br>Internet Protocol                      |    | <ul style="list-style-type: none"> <li>• Internet protocol</li> <li>• Routes data packets</li> </ul>                        |
| <b>UDP</b><br>User Datagram Protocol                |  | <ul style="list-style-type: none"> <li>• Unreliable communication</li> <li>• Connectionless</li> </ul>                      |
| <b>SMTP</b><br>Simple Mail Transfer Protocol        |  | <ul style="list-style-type: none"> <li>• Simple mail transfer</li> <li>• Sends email messages</li> </ul>                    |
| <b>SSH</b><br>Secure Shell                          |  | <ul style="list-style-type: none"> <li>• Secure shell</li> <li>• Remote access</li> </ul>                                   |

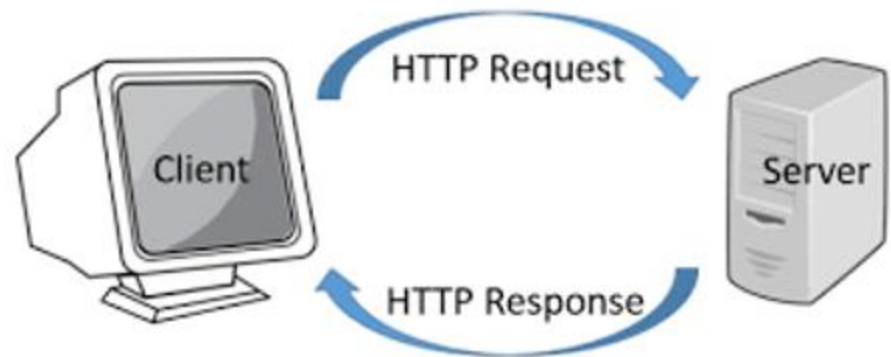
## 1.7. Importancia del modelo TCP/IP

- El modelo TCP/IP permite que millones de dispositivos diferentes se comuniquen de forma estandarizada, haciendo posible el funcionamiento global de Internet



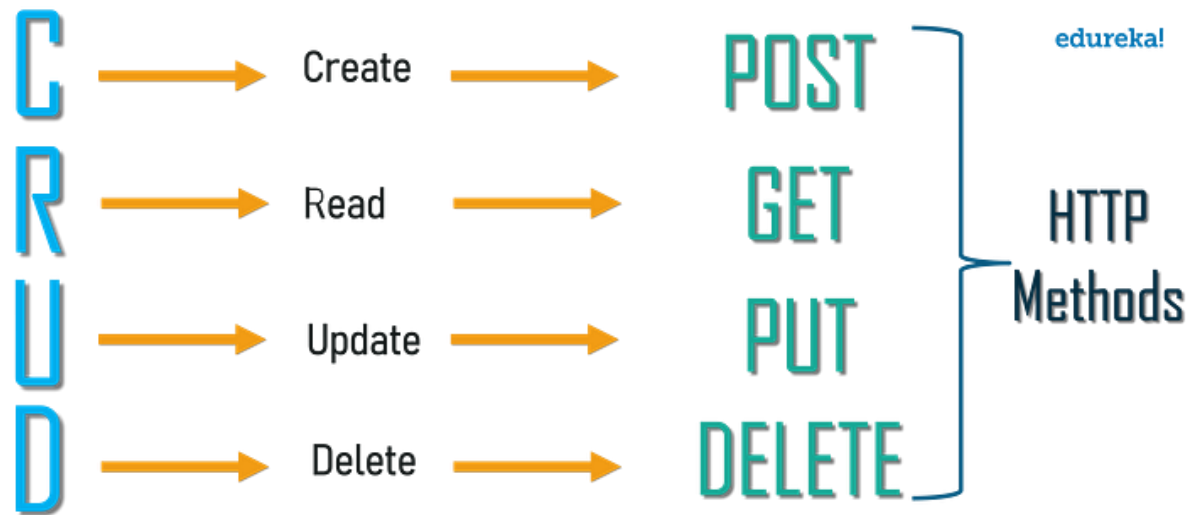
## 2. HTTP

- HyperText Transfer Protocol
- Protocolo que permite la transferencia de información (principalmente documentos HTML, imágenes, videos, etc.) entre un cliente (por ejemplo, un navegador web) y un servidor web



## 2. HTTP

- Métodos HTTP (o *HTTP request methods*) indican la acción que el cliente desea que el servidor realice sobre un recurso identificado por una URL
- Se usan en el ciclo **CRUD** (Create, Read, Update, Delete)

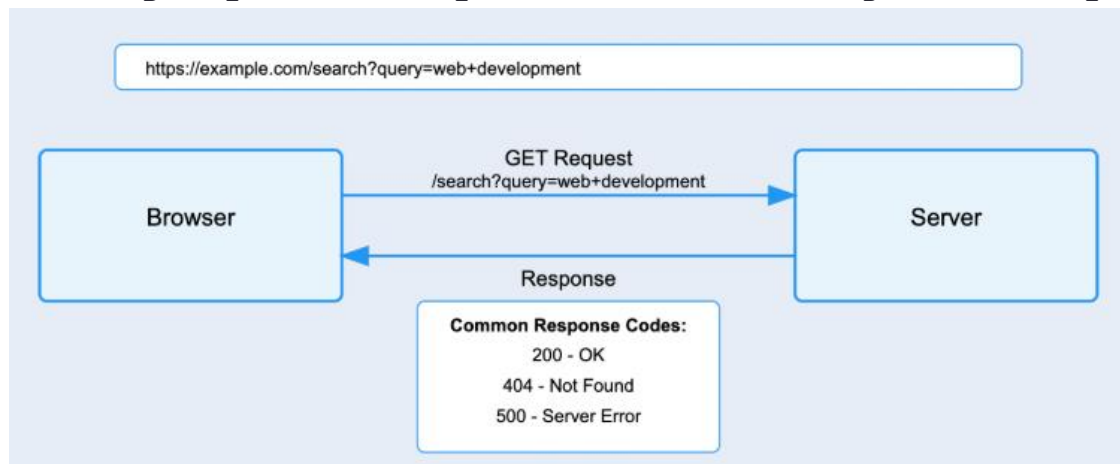




## 2. HTTP

- GET

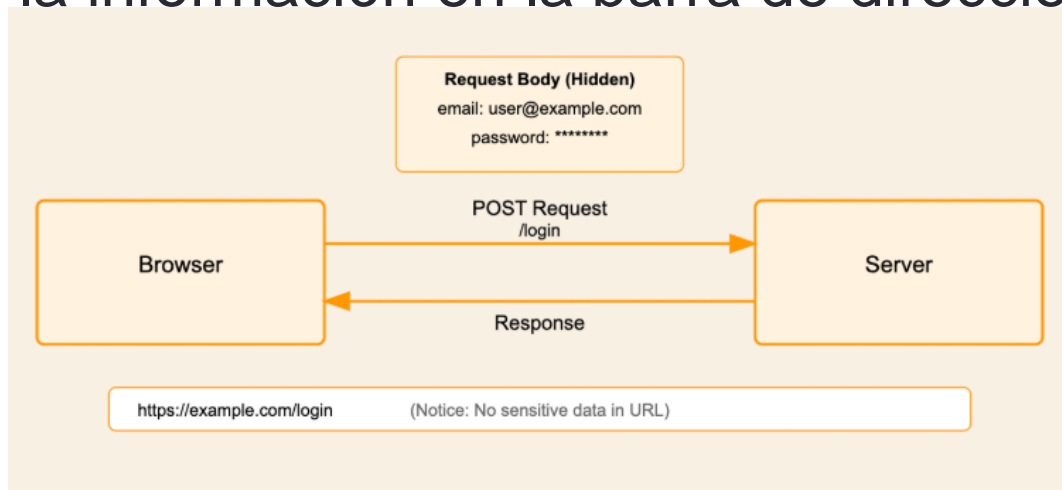
- Función: solicitar y obtener un recurso desde el servidor (HTML, imagen, archivo, datos, etc.)
- No modifica información en el servidor
- Los parámetros se envían en la URL (?clave=valor)
- Ejemplo:
  - `https://ejemplo.com/productos?categoria=ropa`
  - `https://ejemplo.com/productos?categoria=ropa&orden=asc`



## 2. HTTP

### • POST

- Función: enviar datos al servidor para crear un nuevo recurso
- Modifica datos en el servidor (no es solo consulta)
- Los datos van en el cuerpo (body) de la petición, no en la URL
- Más seguro que GET para datos sensibles, porque no expone la información en la barra de direcciones

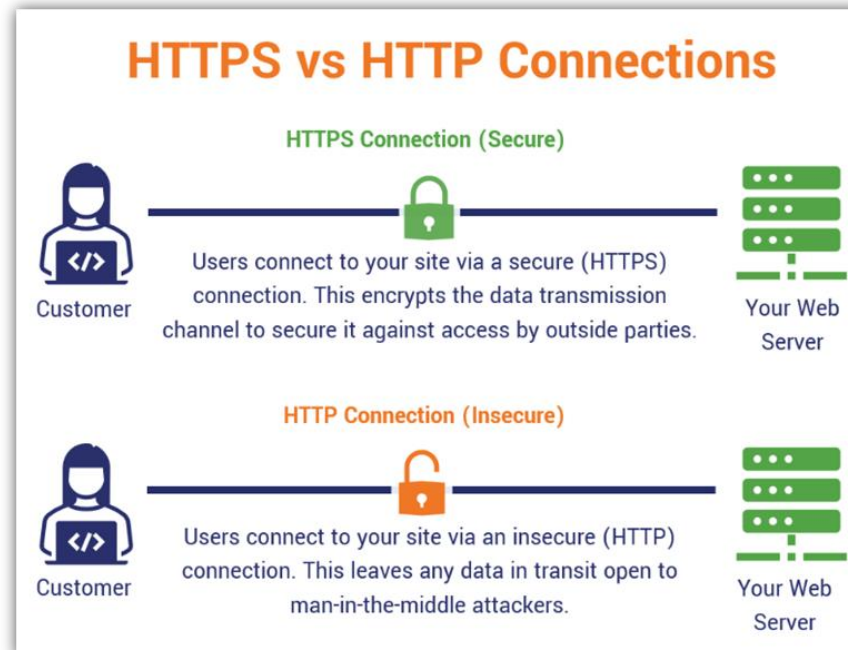


## 2. HTTP

| Método | Función principal                 | ¿Modifica datos? | Uso en REST                              |
|--------|-----------------------------------|------------------|--|
| GET    | Obtener un recurso                | ✗ No             | Lectura ( <i>Read</i> )                  |
| POST   | Crear un nuevo recurso            | ✓ Sí             | Creación ( <i>Create</i> )               |
| PUT    | Reemplazar un recurso existente   | ✓ Sí             | Actualización completa ( <i>Update</i> ) |
| PATCH  | Modificar parcialmente un recurso | ✓ Sí             | Actualización parcial ( <i>Update</i> )  |
| DELETE | Eliminar un recurso               | ✓ Sí             | Eliminación ( <i>Delete</i> )            |

# 3. HTTPS

- Versión segura de HTTP que utiliza cifrado SSL/TLS
- Usa cifrado para proteger la comunicación entre el cliente y el servidor web



## 4.1. TCP

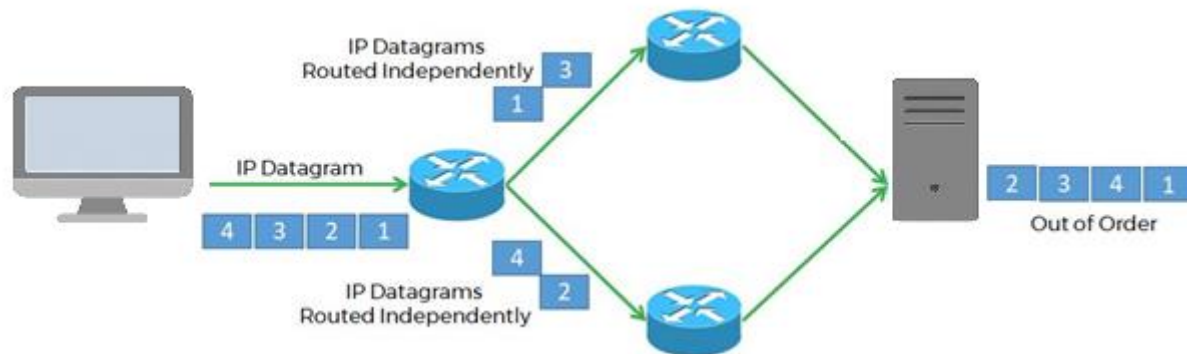
- Es un protocolo de la capa de transporte del modelo TCP/IP
- Garantiza que los datos lleguen completos, en orden y sin errores desde un emisor hasta un receptor
- Orientado a la conexión
  - Establece conexión (three-way handshake)
  - Transfiere datos de manera fiable
  - Cierra la conexión cuando termina la comunicación

### THREE - WAY HANDSHAKE (TCP)



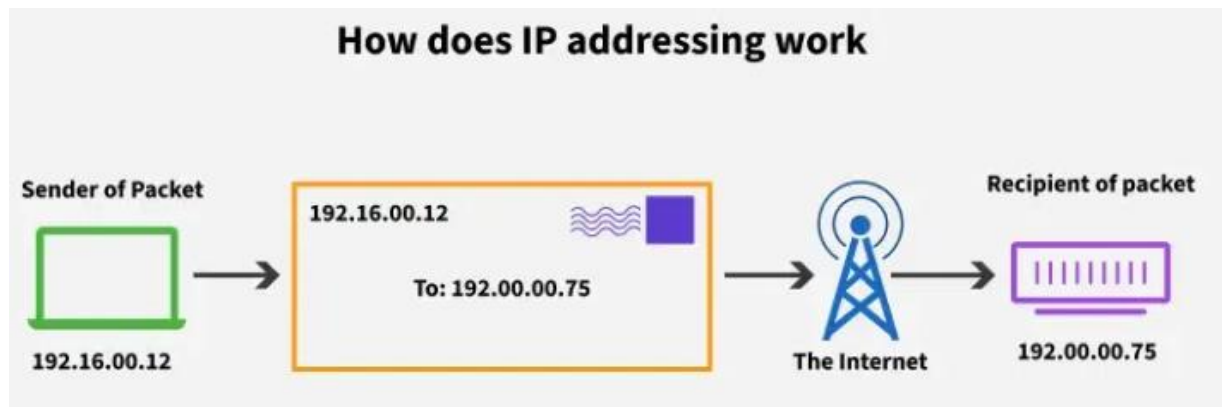
## 4.2. IP

- Internet Protocol
- Protocolo principal de la capa de Internet del modelo TCP/IP
- Envía los paquetes de datos desde el origen hasta el destino a través de diferentes redes



## 4.2. IP

- No orientado a conexión: IP solo se encarga de enviar los paquetes, sin garantizar que lleguen, ni en orden, ni sin errores
- Enrutamiento: Se encarga de decidir por qué camino enviar los paquetes hasta el destino
- Direcciones IP: Cada dispositivo conectado a la red tiene una dirección IP única

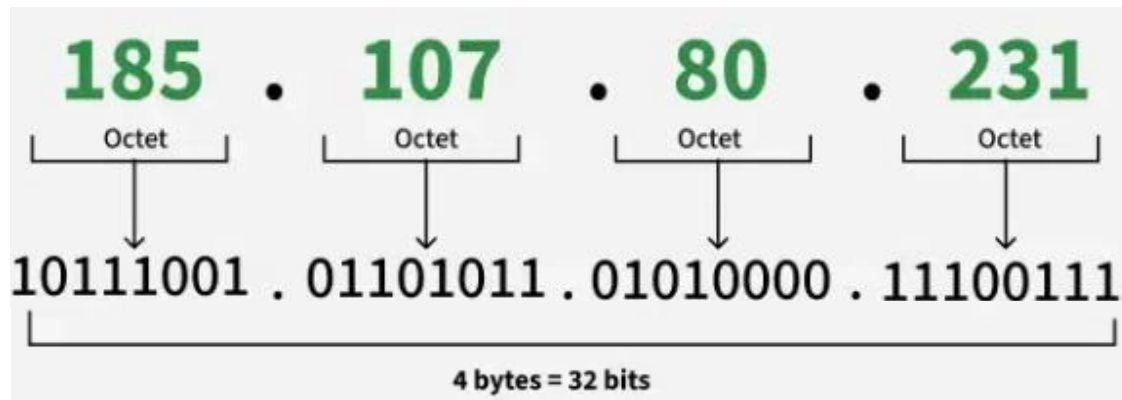


## 4.3. Dirección IP

- Identificador único que se asigna a cada dispositivo conectado a una red que usa el protocolo IP
- Sirve para que los dispositivos puedan enviar y recibir datos entre sí
- IPv4 e IPv6



## 4.4. Dirección IPv4 e IPv6



## IPv6 address in Hexadecimal notation

2001:0DB8:0000:2F3B:02AA:00FF:FE28:9C5A



00100000000000001000011011011100000000000000000000010111100111011000000101010101000000000111111111111110001010001001110001011010

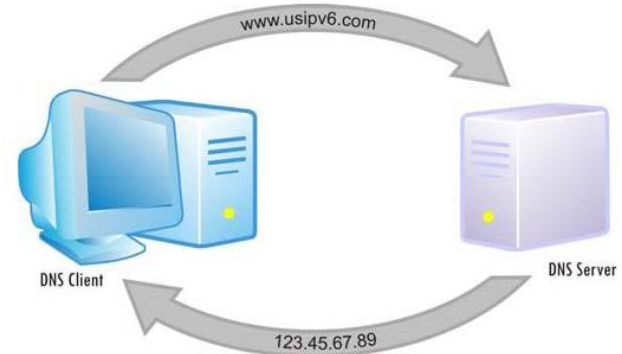
# 5. URL

- Localizador Uniforme de Recursos, dirección única que se usa para localizar un recurso en la web
- Permite a navegadores, apps y usuarios localizar y acceder a recursos en la web

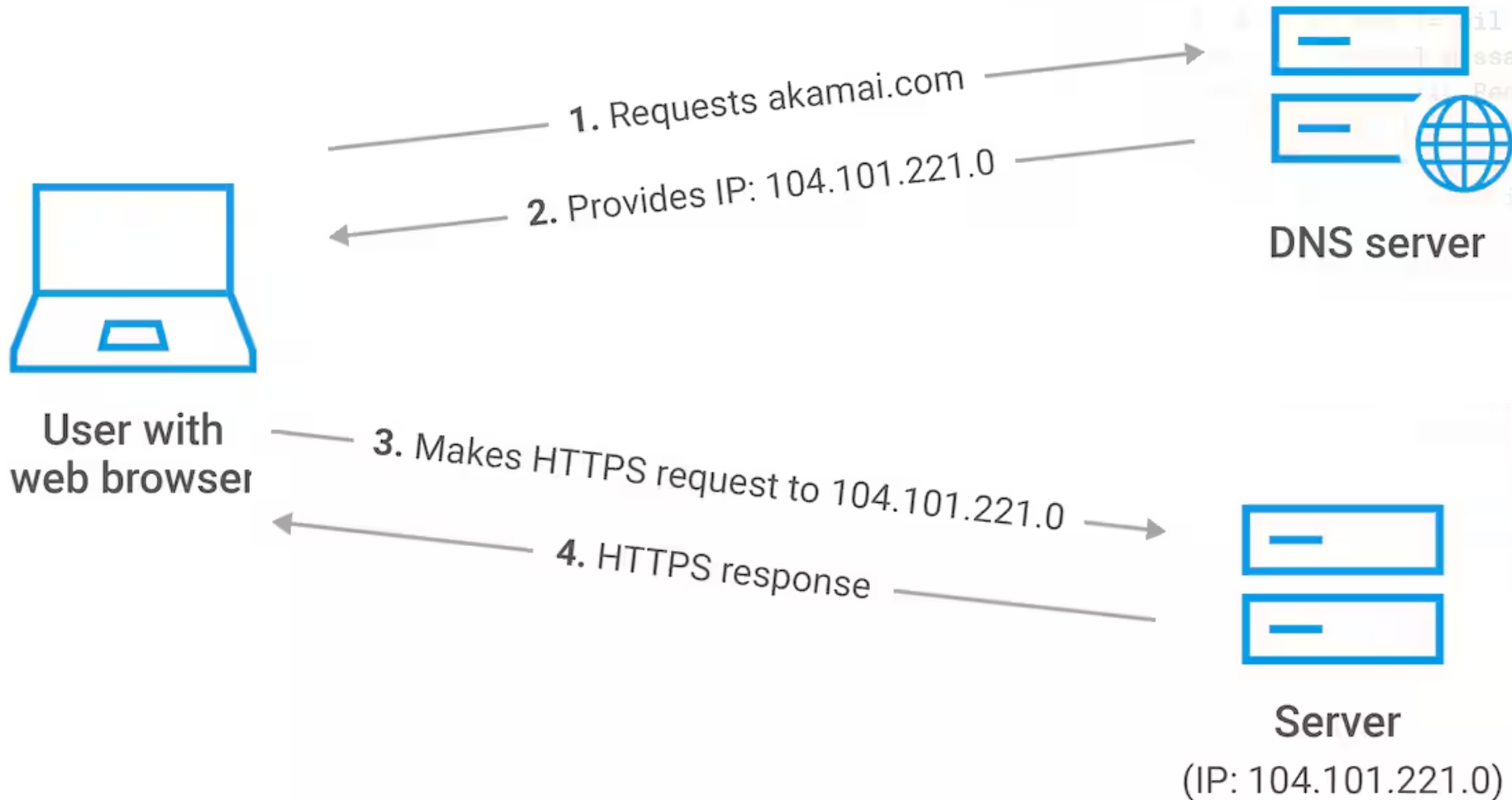


## 6. Servidor DNS

- Guía telefónica de Internet
- Servidor que traduce los nombres de dominio fáciles de recordar (como **mipagina.com**) en direcciones IP (como **142.250.72.206**) que usan los dispositivos para comunicarse
- Las personas recordamos mejor los nombres, pero las computadoras y redes trabajan con direcciones IP



## 6. Servidor DNS



# 7. SSH

- Secure Shell
- Es un protocolo de red que permite conectarse de forma segura a otro equipo a través de una red no segura (por ejemplo, Internet)
- Su uso principal es administrar servidores de manera remota (entrar a una consola/terminal en otro equipo)



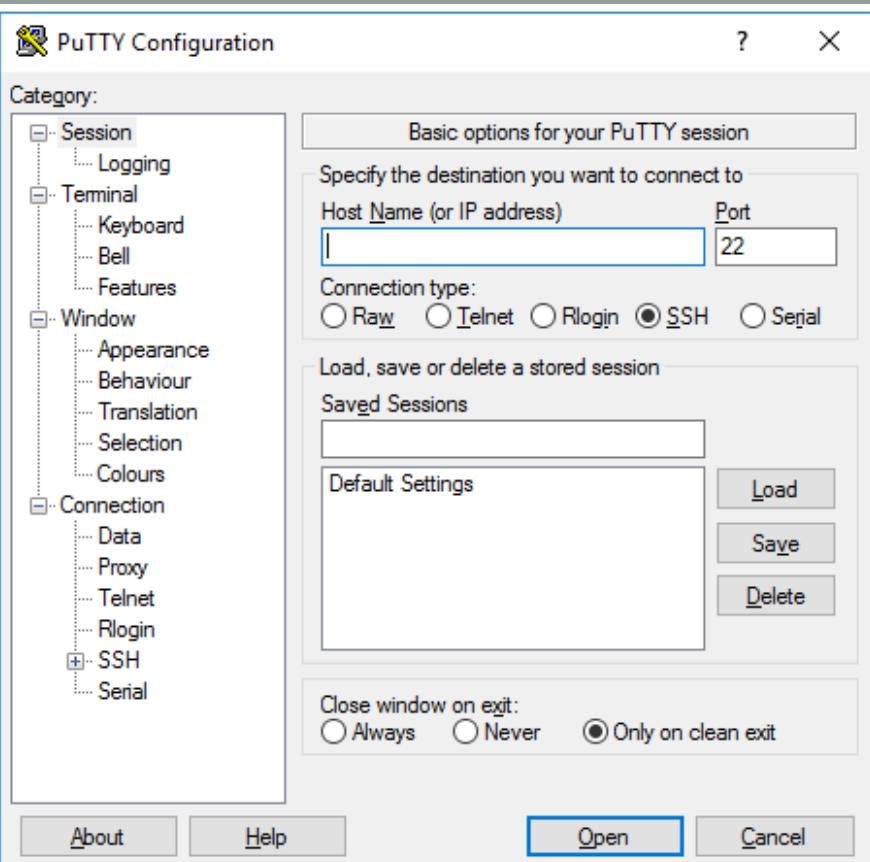
# 7. SSH

- Ejemplo: tienes un servidor en la IP 203.0.113.10 y tu usuario es marco
- En un terminal y usando bash:

```
ssh marco@203.0.113.10
```

- Te pedirá la contraseña, y luego tendrás acceso a la línea de comandos de ese servidor como si estuvieras físicamente frente a él

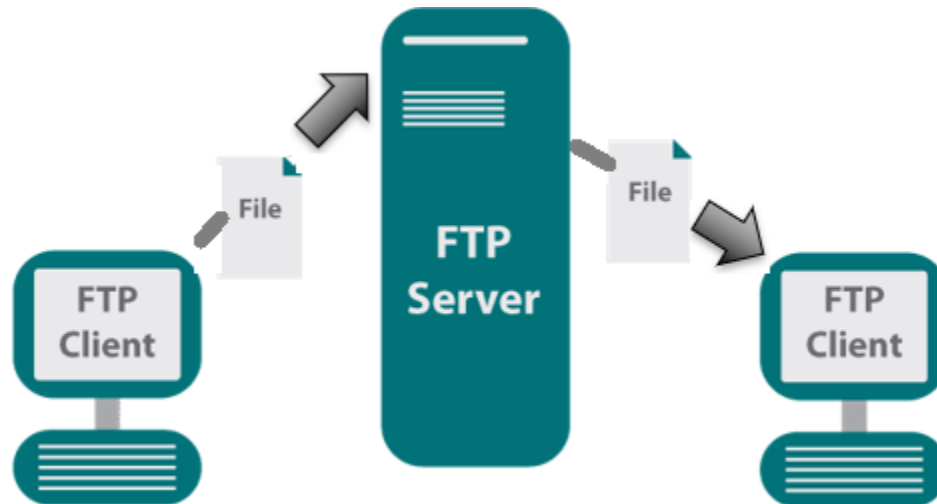




```
root@localhost:~  
[root@localhost ~]# ls /var/log  
anaconda.log          cron                  messages              secure-20160907  
anaconda.program.log  cron-20160822        messages-20160822    secure-20160911  
anaconda.storage.log  cron-20160828        messages-20160828    spooler  
anaconda.syslog       cron-20160907        messages-20160907    spooler-20160822  
anaconda.xlog         cron-20160911        messages-20160911    spooler-20160828  
anaconda.yum.log      dmesg                mobileid              spooler-20160907  
audit                dmesg.old            mysqld.log            spooler-20160911  
boot.log              gdm                  ntpstats              tallylog  
boot.log-20160822     httpd                pm-powersave.log     wpa_supplicant.log  
boot.log-20160828     lastlog              ppp                   wtmp  
boot.log-20160907     maillog              prelink               Xorg.0.log  
boot.log-20160911     maillog-20160822     sa                    Xorg.0.log.old  
btmptmp               maillog-20160828     secure                yum.log  
btmptmp-20160901      maillog-20160907     secure-20160822      yum.log-20140318  
ConsoleKit            maillog-20160911     secure-20160828      yum.log-20160613  
[root@localhost ~]#
```

# 8. FTP

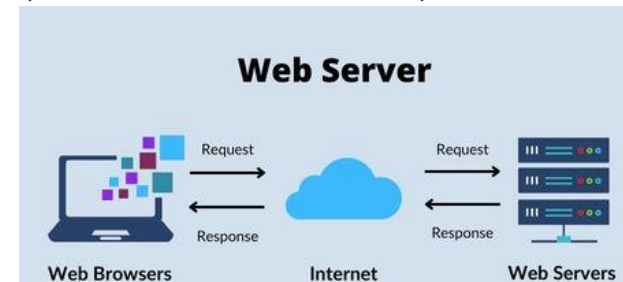
- File Transfer Protocol
- Protocolo para transferir archivos entre cliente y servidor, a través de internet
- Permite subir (upload) y descargar (download) archivos





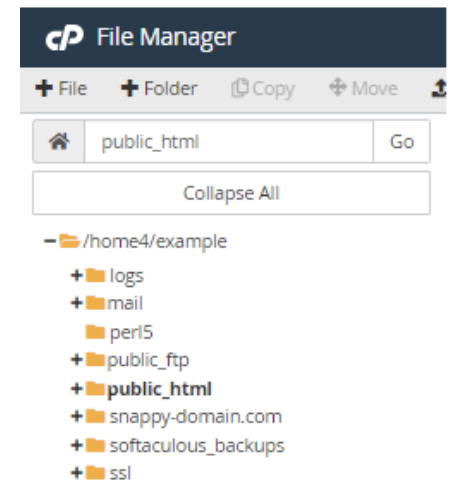
# 9. Servidor WEB

- Servidor que almacena y sirve páginas web a los usuarios
- Recibe solicitudes de los navegadores (clientes) usando principalmente el protocolo HTTP o HTTPS
- Envía de vuelta páginas web, imágenes, archivos o cualquier recurso web solicitado
- Ejecutar scripts (como Python, PHP, etc.) para generar contenido dinámico
- Apache HTTP Server, Nginx, Cloudflare, Microsoft IIS, LiteSpeed



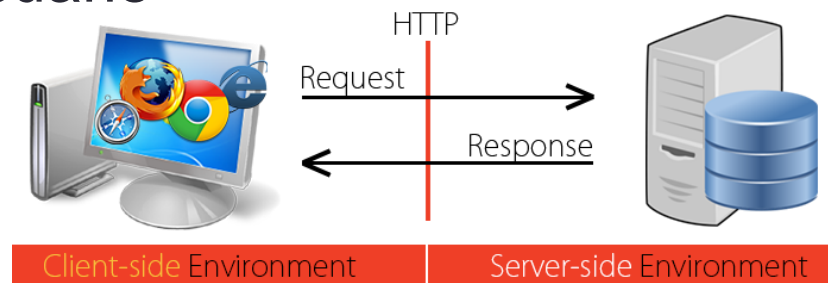
# 10. Directorio public\_html

- Es un directorio especial que existe en muchos servicios de hosting (por ejemplo, cPanel)
- Su función es ser el directorio público del servidor web
- Espacio donde debes subir tu sitio web para que esté disponible en Internet

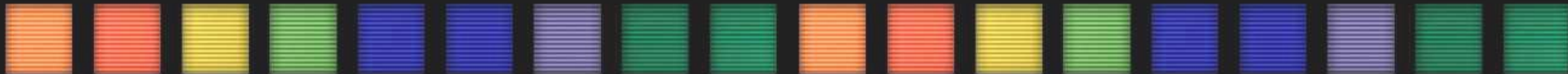


# 11. Cliente WEB

- Aplicación que se conecta a un servidor web para solicitar, recibir y mostrar contenido de la web
- El navegador web (Chrome, Firefox, Edge, Safari, etc.)
  - Envía peticiones HTTP o HTTPS
  - Recibe respuestas del servidor (HTML, CSS, imágenes, etc.)
  - Las interpreta y muestra al usuario



# PROTOCOLOS



**Gracias**