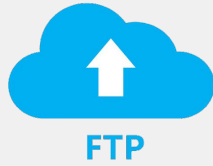


# Servidores Web, DNS y FTP



JOSÉ ANTONIO CASTILLEJO LOBATO  
NADIA LIA CORREA MORALES  
CHISELA MARIEN COLÁS GIL  
ROMÁN MILLÁN DÍAZ  
SANDRA RUIZ JIMÉNEZ


---

# Índice




01

## ¿Qué es Internet?

- Protocolos de Internet
  - Url
- 


02

## Protocolo HTTP

- Métodos de petición HTTP
  - Códigos de estado
  - MIME
- 

03

## Arquitecturas Web

- Componentes principales
  - Tipos de Arquitectura Web
- 


---

# Índice




**04**

## **Servidores Web**

- Como funcionan
  - Servidores populares
- 


**05**

## **DNS**

- Nomenclatura
  - Estructura
  - Funcionalidad
- 

**06**

## **FTP**

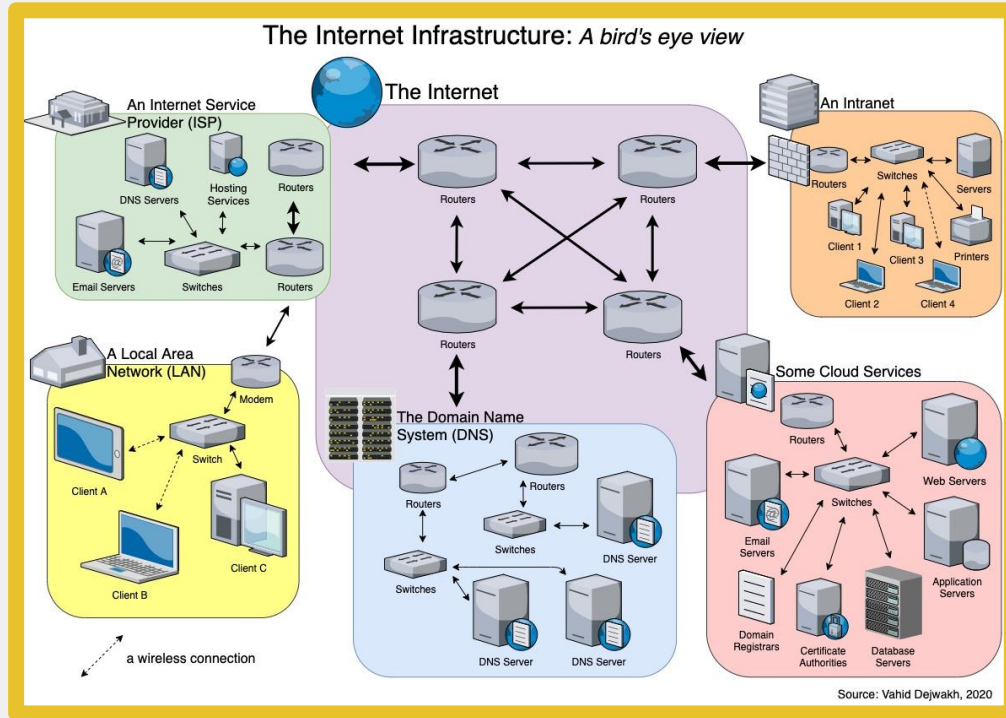
- Usuarios y acceso anónimo
  - Clientes gráficos y de línea de comando
  - Ftp seguro
- 



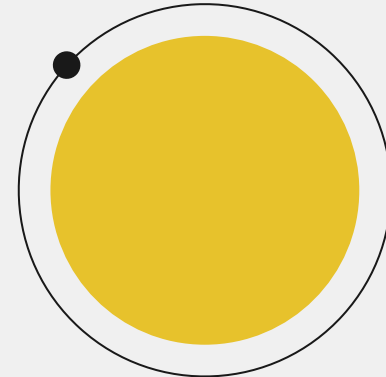
# ¿Qué es Internet?



# ¿Qué es Internet?



Sistema de redes interconectadas mediante distintos protocolos que ofrece una gran diversidad de servicios y recursos, como el acceso a archivos de hipertexto a través de la web.



# 1.1 Protocolos Internet



Son usados para la transmisión de datos en Internet.

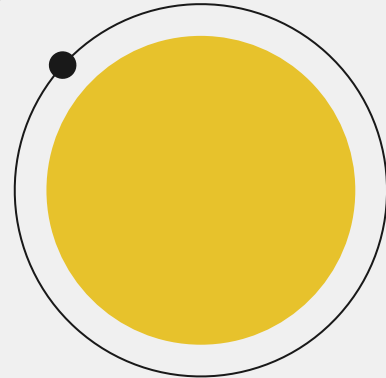
Estos incluyen mecanismos para que los dispositivos se identifiquen y establezcan conexiones entre sí, así como reglas de formato en los mensajes enviados y recibidos.

Hex	Protocol Number	Keyword	Protocol
0x00	0	HOPOPT	IPv6 Hop-by-Hop Option
0x01	1	ICMP	Internet Control Message Protocol
0x02	2	IGMP	Internet Group Management Protocol
0x03	3	GGP	Gateway-to-Gateway Protocol
0x04	4	IP-in-IP	IP in IP (encapsulation)
0x05	5	ST	Internet Stream Protocol
0x06	6	TCP	Transmission Control Protocol
0x07	7	CBT	Core-based trees
0x08	8	EGP	Exterior Gateway Protocol
0x09	9	IGP	Interior Gateway Protocol (any private interior gateway, for example Cisco's IGRP)
0x0A	10	BBN-RCC-MON	BBN RCC Monitoring
0x0B	11	NVP-II	Network Voice Protocol
0x0C	12	PUP	Xerox PUP
0x0D	13	ARGUS	ARGUS
0x0E	14	EMCON	EMCON
0x0F	15	XNET	Cross Net Debugger
0x10	16	CHAOS	Chaos
0x11	17	UDP	User Datagram Protocol
0x12	18	MUX	Multiplexing
0x13	19	DCN-MEAS	DCN Measurement Subsystems
0x14	20	HMP	Host Monitoring Protocol
0x15	21	PRM	Packet Radio Measurement
0x16	22	XNS-IDP	XEROX NS IDP
0x17	23	TRUNK-1	Trunk-1
0x18	24	TRUNK-2	Trunk-2
0x19	25	LEAF-1	Leaf-1

# 1.1 Protocolos Internet



Los más importantes son TCP e IP.  
De manera conjunta (TCP/IP)  
podemos enlazar los dispositivos  
que acceden a la red.



# 1.1 Protocolos Internet



El conjunto de protocolos TCP/IP está organizado en 4 capas conceptuales que están relacionadas con los niveles OSI. La diferencia reside en que TCP/IP organiza y agrupa las capas de manera diferente, haciendo que sea totalmente adaptable a futuras implementaciones, adaptaciones o normativas.

Capas según el modelo OSI		Capas según el modelo TCP/IP	
7	Aplicación <i>Application</i>	4	Aplicación <i>Process</i>
6	Presentación <i>Presentation</i>		
5	sesión <i>Session</i>		
4	Transporte <i>Transport</i>	3	Transporte <i>Host-to-Host</i>
3	Red <i>Network</i>	2	Internet <i>Network</i>
2	Enlace de datos <i>Data Link</i>	1	Acceso al medio <i>Media</i>
1	Física <i>Physical</i>		



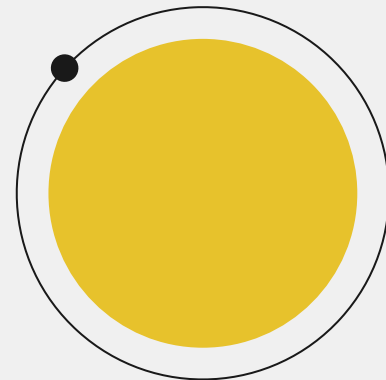
# 1.1 Protocolos Internet



Common TCP/IP Protocols and Ports

Protocol	TCP/UDP	Port Number	Description
File Transfer Protocol (FTP) (RFC 959)	TCP	20/21	FTP is one of the most commonly used file transfer protocols on the Internet and within private networks. An FTP server can easily be set up with little networking knowledge and provides the ability to easily relocate files from one system to another. FTP control is handled on TCP port 21 and its data transfer can use TCP port 20 as well as dynamic ports depending on the specific configuration.
Secure Shell (SSH) (RFC 4250-4256)	TCP	22	SSH is the primary method used to manage network devices securely at the command level. It is typically used as a secure alternative to Telnet which does not support secure connections.
Telnet (RFC 854)	TCP	23	Telnet is the primary method used to manage network devices at the command level. Unlike SSH which provides a secure connection, Telnet does not, it simply provides a basic unsecured connection. Many lower level network devices support Telnet and not SSH as it required some additional processing. Caution should be used when connecting to a device using Telnet over a public network as the login credentials will be transmitted in the clear.

Existen muchas familias de protocolos asociados a TCP/IP en sus diferentes implementaciones y versiones, de hecho hay más de 100 protocolos para diferentes funcionalidades.



# 1.1 Protocolos Internet



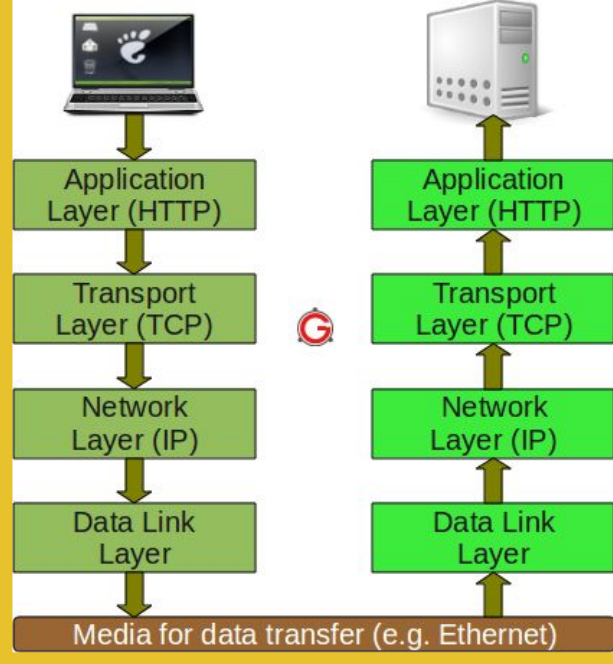
## Las 7 capas del modelo OSI



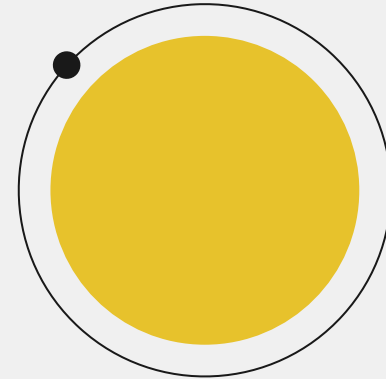
Esquema del modelo OSI

Host (your laptop)

Target (webserver)



Esquema del modelo TCP/IP



## 1.2 URL



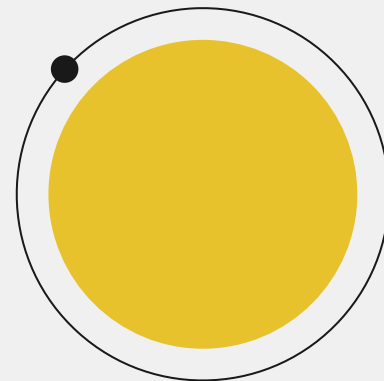
URL (Uniform Resource Locator) es la dirección única y específica que se asigna a cada uno de los recursos que hay en Internet (sitios web, páginas web, textos, fotos, vídeo...) para que puedan ser localizados por el navegador y visualizados por el usuario.

`https://www.ejemplo.com/pagina/subpagina?parametro1=ABCD&parametro2=DEFG#etiqueta`

protocolo      subdominio      dominio      ruta      parámetros (querystring)      etiqueta (hashtag)

`https://www.spanishwithedyta.com/home/SearchForm?Search=subjunctive&searchlocale=en_GB&action_results=Go`

protocolo      subdominio      dominio      ruta      parámetros (querystring)





## 1.2 URL



### Protocolo HTTP

https://

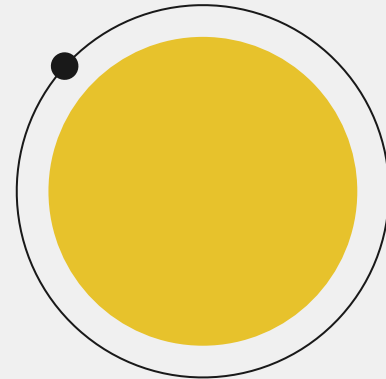
Hypertext Transfer Protocol y el protocolo en sí sirve para transferir información entre diferentes actores dentro de Internet.

### Subdominio

www

Es una extensión del nombre de dominio que se utiliza para organizar diferentes secciones de una web y que funciona de manera independiente a la misma.

*Ej. <https://es.ejemplodesubdominio.com>*





## 1.2 URL



Dominio

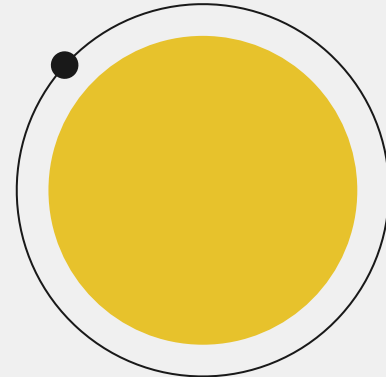
`.ejemplo.com`

El dominio está compuesto por dos elementos: el nombre de dominio y la TLD (top domain level).

Las TLD a veces dan pistas sobre el tipo de página que tenemos delante. Ej: .gob o .gov (gobierno) .org (educación o las ONGs)

Ruta `/pagina/subpagina`

Indica páginas y subpáginas que podemos encontrar en un sitio web.





## 1.2 URL



Parámetro (o query string)

`?parametro1=ABCD&parametro2=DEFG`

En una URL puede haber varios parámetros. Y cuando es el caso, éstos se separan con el símbolo &.

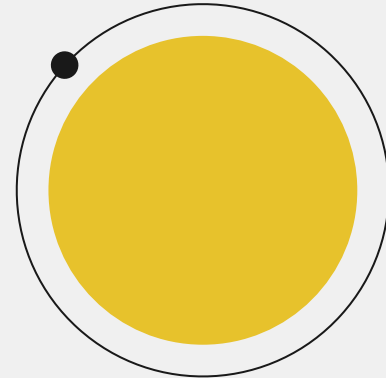
Los parámetros contienen el envío de datos.

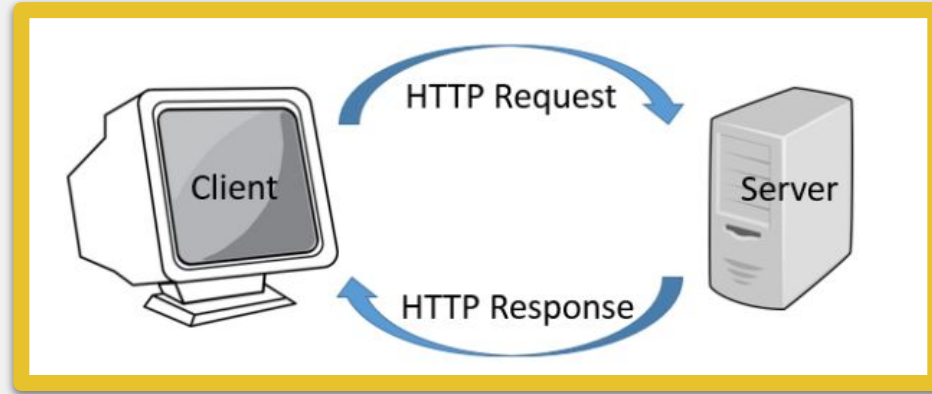
Etiqueta

`#etiqueta`

Las etiquetas en una URL aparecen después del hashtag #.

Permiten hacer scroll hasta un elemento en concreto. Ej: si mandamos una URL que contenga una etiqueta, ésta le dirigirá a la parte exacta de la página en cuestión.



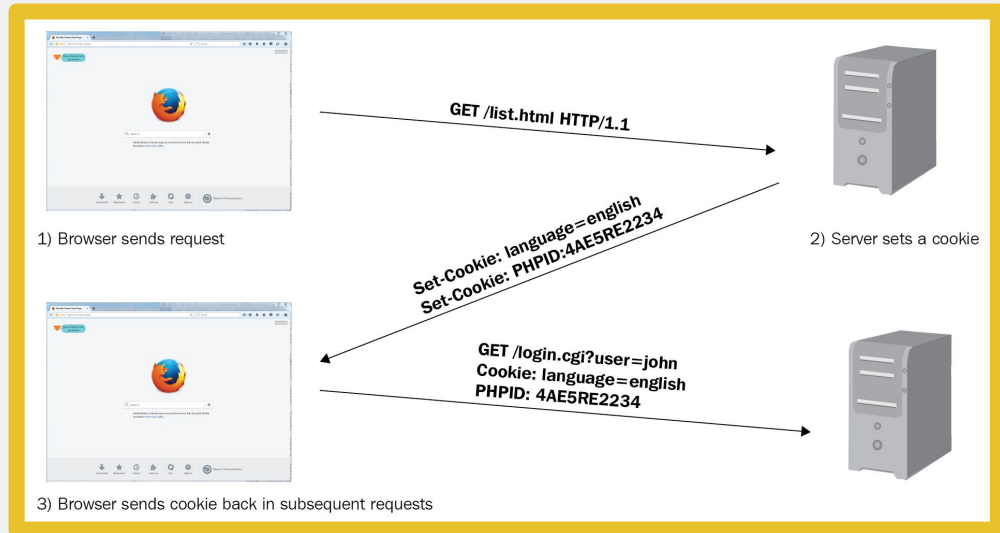


# Protocollo HTTP



## 2.1 HyperText Transfer Protocol

- Protocolo **sin estado** → no guarda información sobre conexiones anteriores
- **COOKIES** → información que servidor almacena en cliente.





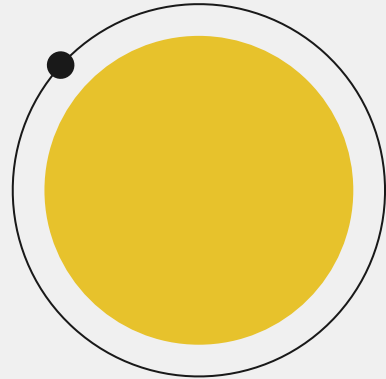


## 2.2 Métodos de petición HTTP

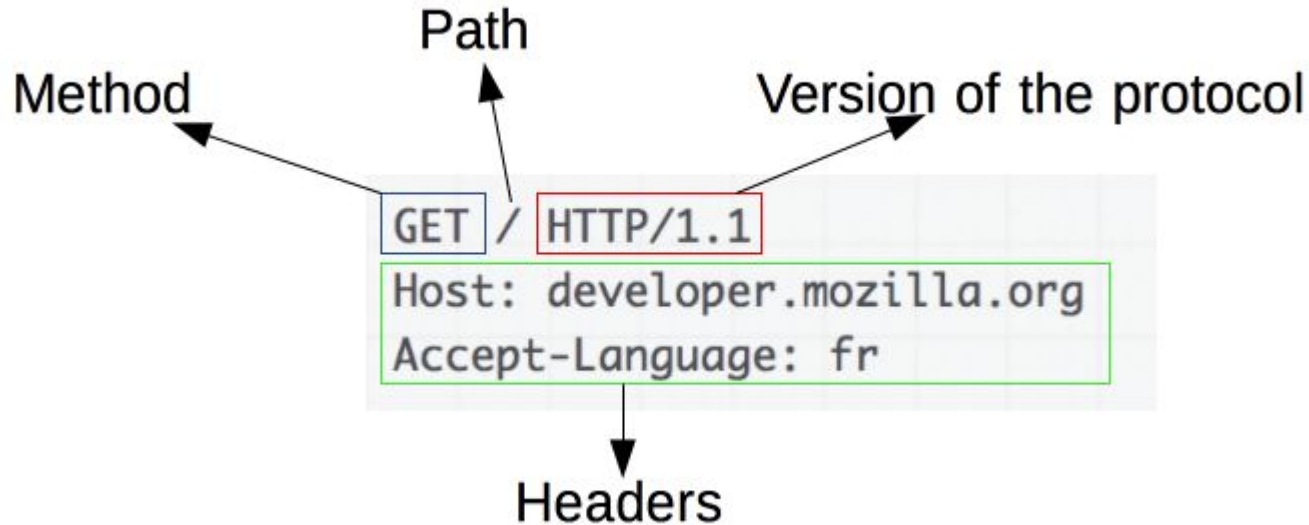
---



**GET**  
**POST**  
**PUT**  
**DELETE**  
**PATCH**  
**HEAD**  
**CONNECT**  
**OPTIONS**  
**TRACE**



## 2.2 Métodos de petición HTTP





## 2.2.1 Método GET



---

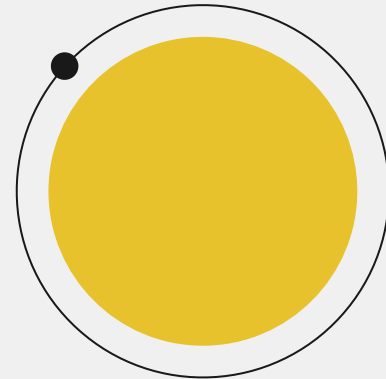
Solicita un recurso específico

Es cacheable

**Sintaxis:** GET /index.html

**Ejemplo:**

GET/RegisterStudent.asp?user=value1&pass=value2





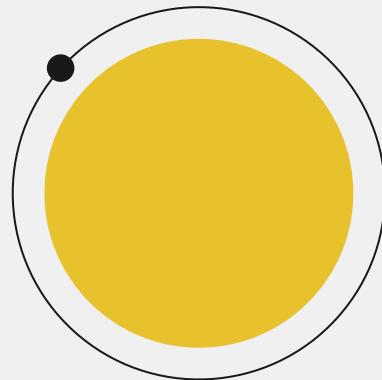
## 2.2.2 Método POST



- 
- Modificar recursos existentes.
  - Publicar mensaje en un tablón de anuncios, grupo de noticias, lista de correos, etc.
  - Agregar nuevo usuario → modal de suscripciones.
  - Proveer datos → envío de un formulario.
  - Extender BBDD → operación de concatenación.

Solo cacheable si incluye nueva info.

**Sintaxis:** **POST** /index.html





## 2.2.3 Método PUT



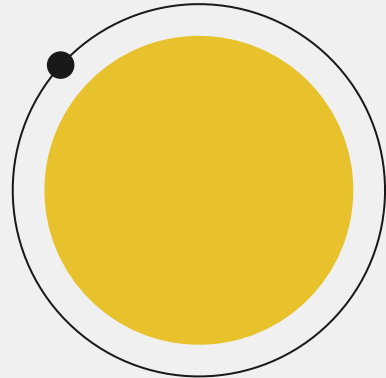
---

Crea un nuevo elemento o reemplaza una representación del elemento de destino con los datos de la petición

PUT vs POST: PUT es idempotente → llamarlo de forma sucesiva tiene el mismo efecto, POST puede tener efectos adicionales

No es cacheable.

**Sintaxis:** PUT /nuevo.html HTTP/1.1





## 2.2.4 Método DELETE



---

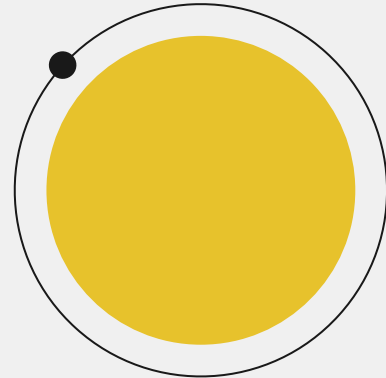
Borra un recurso en específico.

No es cacheable.

**Sintaxis:** DELETE /file.html HTTP/1.1

```
DELETE /file.html HTTP/1.1
```

```
Host: example.com
```





## 2.2 Códigos de estado

---

Respuestas del servidor a la petición.

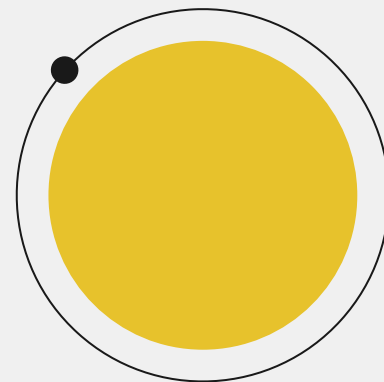
*Ejemplos:*

```
HTTP/1.1 204 No Content
```

```
Content-Location: /existing.html
```

```
HTTP/1.1 200 OK
```

```
Date: Wed, 21 Oct 2015 07:28:00 GMT
```



## 2.2 Códigos de estado







## 2.3 MIME (*Multipurpose Internet Mail Extensions*)

---

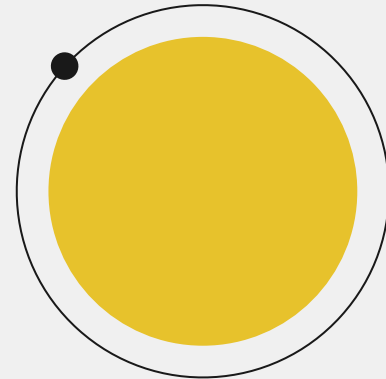
Indica naturaleza y formato de documento, archivo o conjunto de datos.

Navegador usa MIME para saber cómo procesar un documento.

Adjuntado al encabezado del objeto de respuesta.

**Estructura:** tipo/subtipo

- Tipo: categoría (discreto/multiparte).
- Subtipo: específico para el tipo.





## 2.3.1 Tipos discretos

---

Tipo	Descripción	Ejemplo de subtipos típicos
<b>text</b>	Texto	text/plain, text/html, text/css, text/javascript
<b>image</b>	Imagen	image/gif, image/png, image/jpeg, image/bmp, image/webp
<b>audio</b>	Audio	audio/midi, audio/mpeg, audio/webm, audio/ogg, audio/wav
<b>video</b>	Video	video/webm, video/ogg
<b>application</b>	Datos binarios	application/octet-stream application/pkcs12 application/vnd.msppowerpoint application/xhtml+xml application/xml application/pdf



## 2.3.2 Tipos multiparte

---

Categoría de documento que está rota en distintas partes.

Diferentes tipos de MIME.

**multipart/form-data** → formularios HTML y el método POST

**multipart/byteranges** → código de estado 206

**Resto** → HTTP no maneja documentos multiparte de una manera específica  
→ mensaje se transmite al navegador, con ventana Guardar como...



# Arquitecturas Web



# Componentes principales

## Servidor Web:

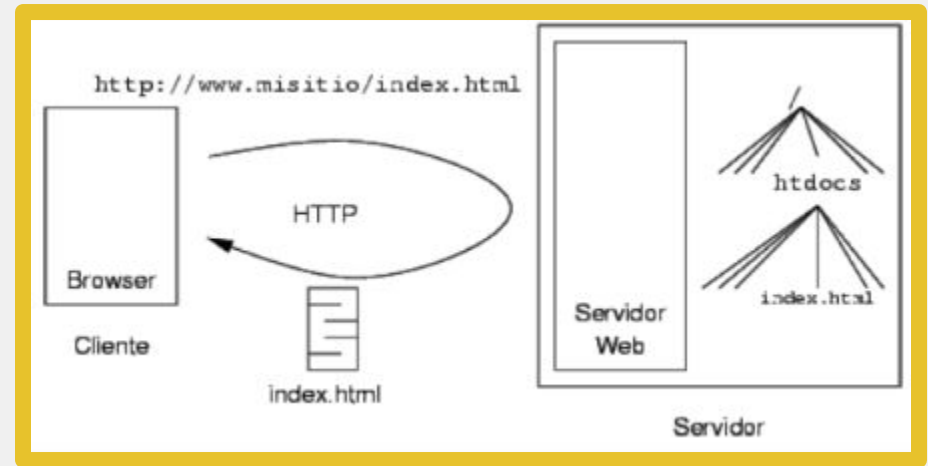
Esperando permanentemente las solicitudes de conexión y distribuye páginas de información formateada

## Conexión de red:

Requerimientos mediante el uso del protocolo HTTP.

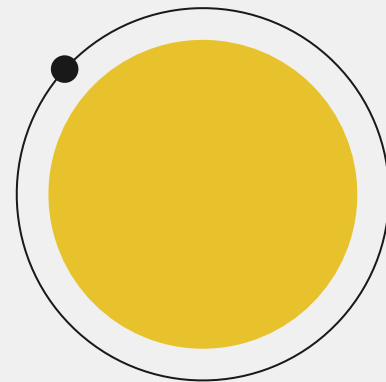
## Cliente(s):

Con quien el usuario interactúa para solicitar el envío de recursos al servidor. Interpreta los recursos.





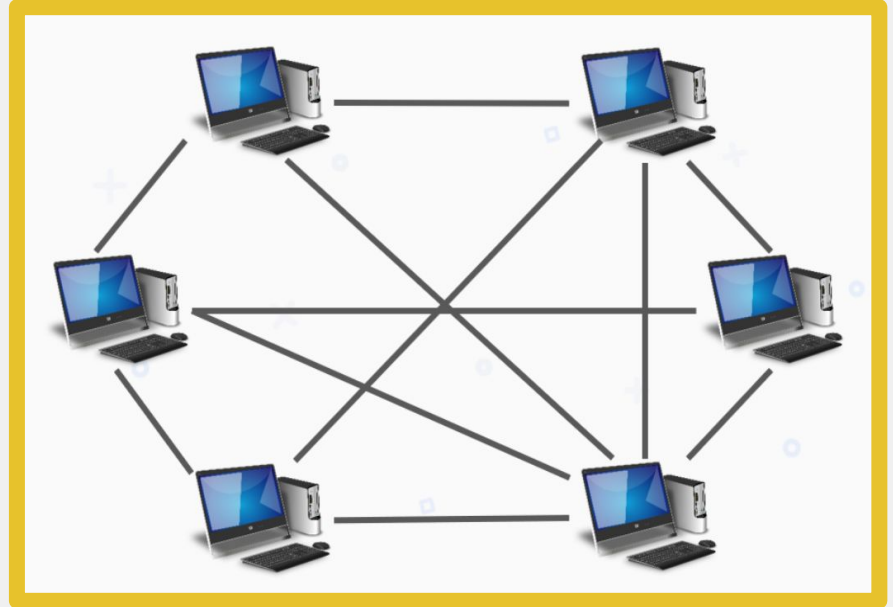
## 3.1 Tipos de Arquitectura Web



### 3.1.1 Arquitectura Peer to Peer (P2P)

Red de computadoras donde todos los dispositivos conectados a la red actúan como cliente y servidor al mismo tiempo.

No necesita de un servidor central que administre la red (aunque puede existir), si no que todos los nodos de la red pueden comunicarse entre sí.





## 3.1.1 Arquitectura Peer to Peer (P2P)

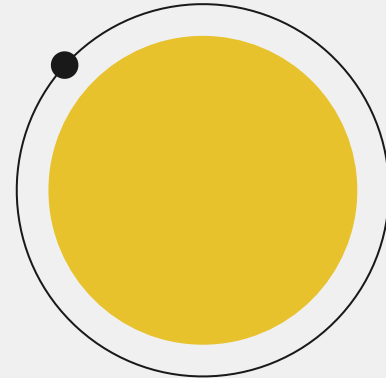
---

### Ventajas

Alta escalabilidad  
Tolerancia a fallos  
Descentralización  
Privacidad  
Equilibrio de carga

### Desventajas

Alta complejidad  
Control  
Seguridad  
Tráfico



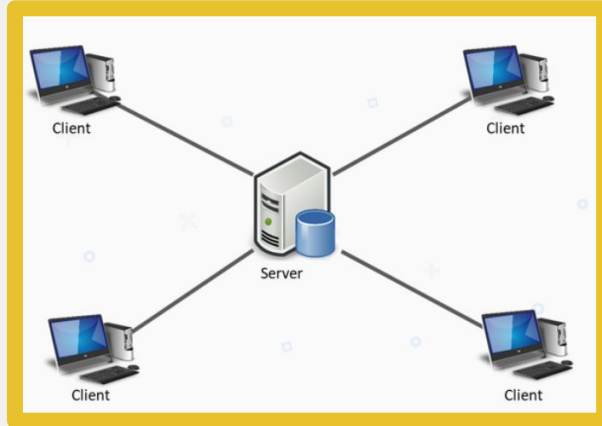


## 3.1.2 Arquitectura Cliente-Servidor

### Compuesta:

Proveedor - brinda servicios o recursos

Consumidor - consume servicios o recursos



### Funcionamiento básico:

- Usuario - Navegador web (cliente) solicita la carga de una página web con URL
- Servidor recibe petición, busca en sistema de almacenamiento, envía el contenido por el mismo medio
- Navegador web - recibe el código y lo interpreta mostrando al usuario



## 3.1.2 Arquitectura Cliente-Servidor

---

### Ventajas

Centralización del control

Seguridad

Facilidad de instalación (cliente)

Separación de responsabilidades

Portabilidad

### Desventajas

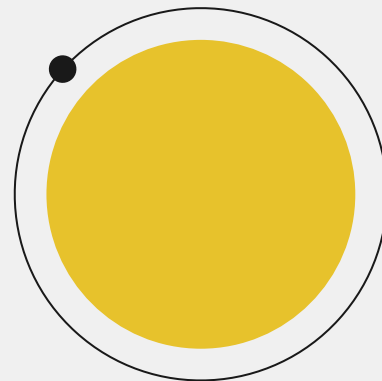
Actualizaciones (cliente)

Concurrencia

Todo o nada

Protocolos de bajo nivel

Complicada depuración



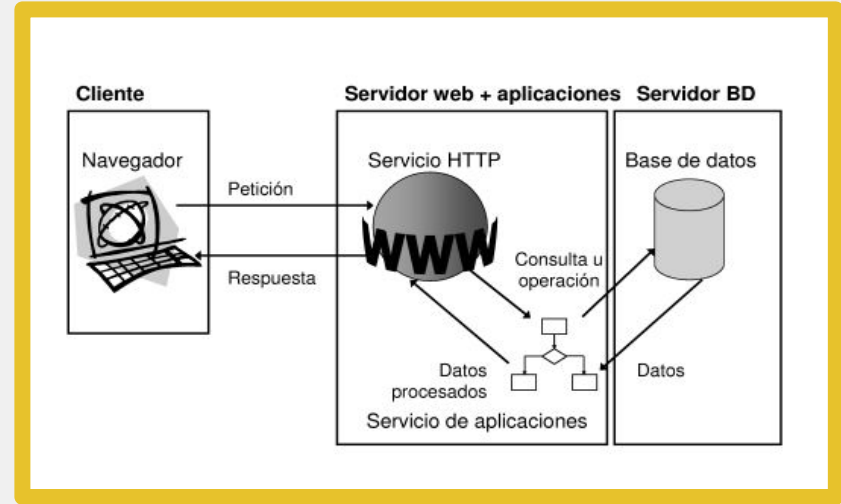
## 3.1.3 Arquitectura con Servidor de Aplicaciones

### Servidor de aplicaciones:

- Corazón de un gran sistema distribuido
- Proporciona servicios que soportan la ejecución y disponibilidad de las aplicaciones desplegadas.

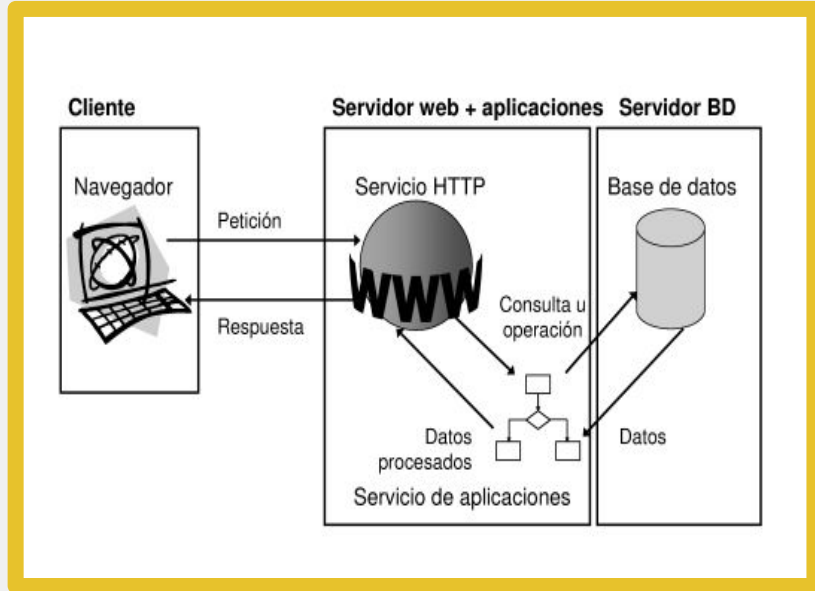
Separa la lógica de datos y los datos a un servidor de base de datos específico.

El servidor web y el servidor de aplicaciones pueden residir en una misma máquina.



La base de datos puede residir en la misma máquina que el servidor web o en otro host conectado en red.

### 3.1.3 Arquitectura con Servidor de Aplicaciones



El servidor web - petición de un recurso, si es un recurso dinámico, transfiere la parte correspondiente al servidor de aplicaciones.

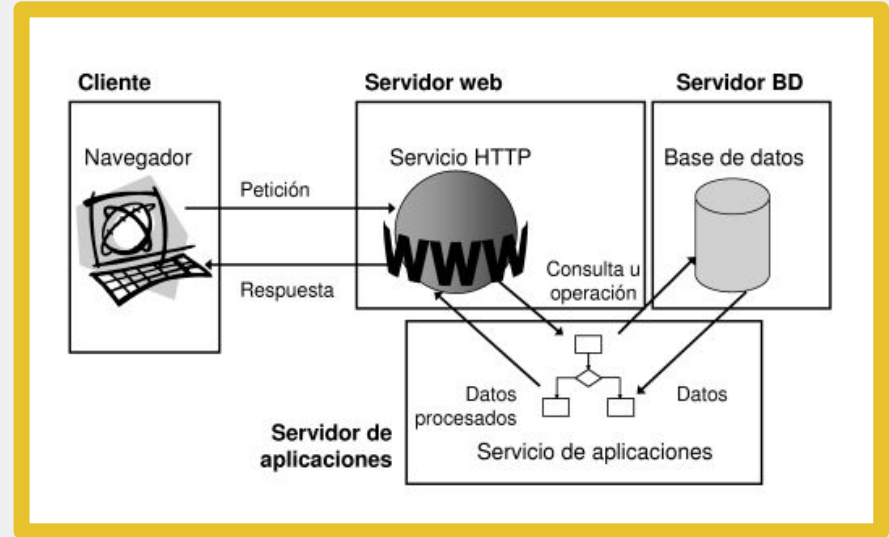
El servidor de aplicaciones devolverá el recurso ejecutado. Frecuentemente el servidor tendrá que conectarse con una base de datos con el fin de obtener los datos solicitados.

Finalmente será el servidor web el que envíe al cliente el resultado final.

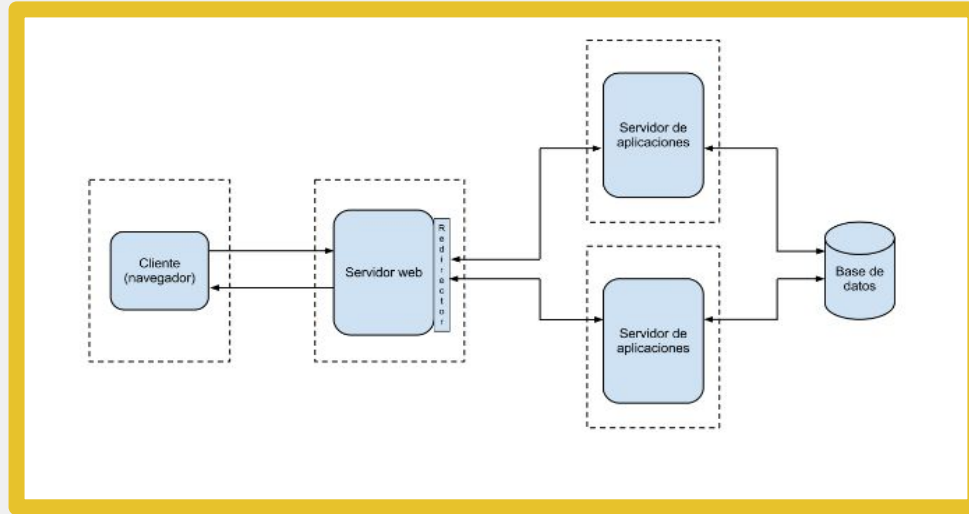
### 3.1.4 Arquitectura con Servidor de Aplicaciones Externo

Las tres funcionalidades básicas del servidor web se separan en tres servidores específicos.

La parte correspondiente al servidor web suele tener menos carga de trabajo que el servidor de aplicaciones.



### 3.1.5 Arquitectura con Varios Servidores de Aplicaciones



Usada cuando se estima que la carga de trabajo del servidor de aplicaciones será elevada.

Sistema con varios servidores de aplicaciones unidos a un mismo servidor web que requiere menos rendimiento.

---



# Servidores Web



## 4.1 ¿Qué es un servidor web?

---

Software que forma parte del navegador.

Utilizados para la distribución y entrega de contenido web.

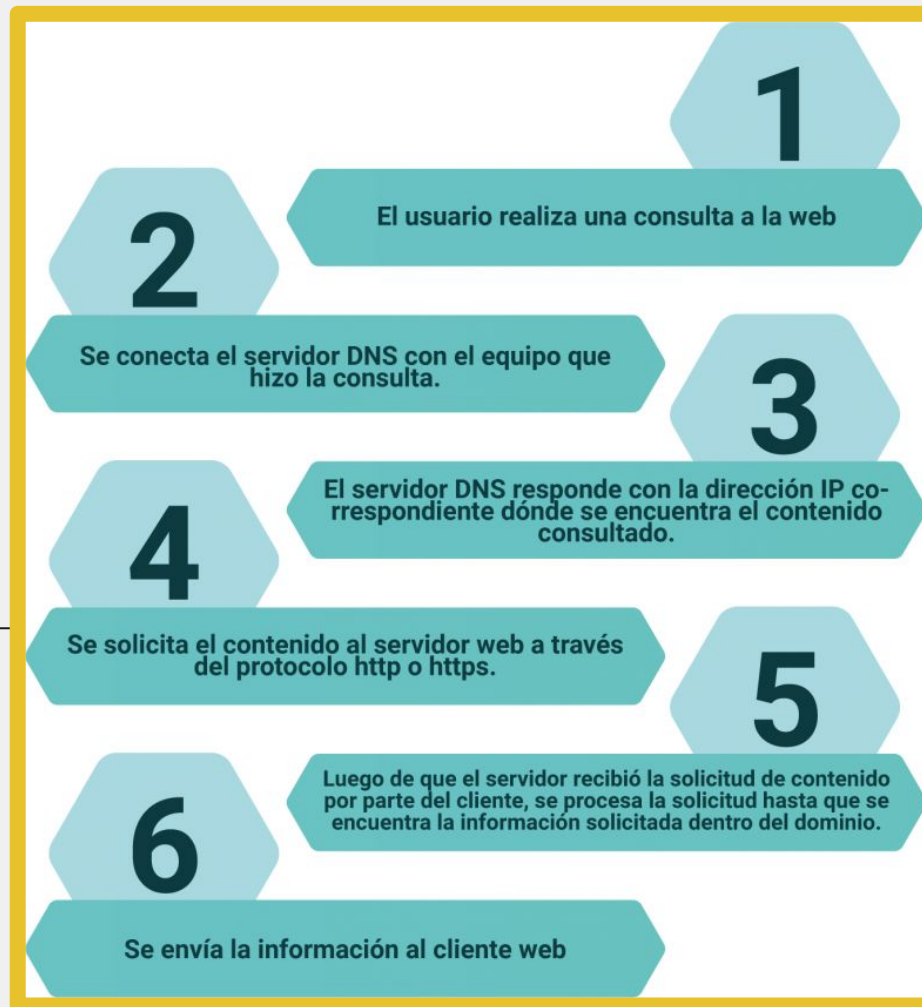
**Función:** Mostrar la información cuando los usuarios la solicitan.

---





## 4.2 ¿Cómo funciona?



## 4.3 Servidores web populares

---



- Código abierto y gratuito.
- Arquitectura basada en eventos.
- Gestiona sitios web de alto tráfico.



- SO Windows.
- Arquitectura modular.
- Modelo de proceso único.



- Código abierto y gratuito.
  - Arquitectura basada en módulos.
  - Funciones de autenticación y validación de usuarios.
-



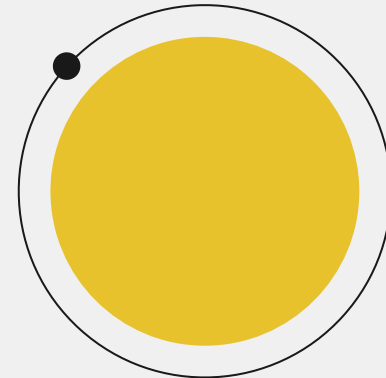
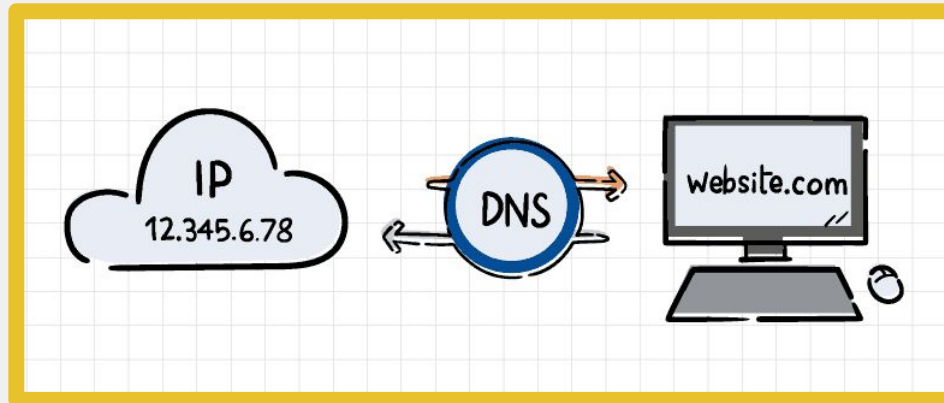
**DNS**

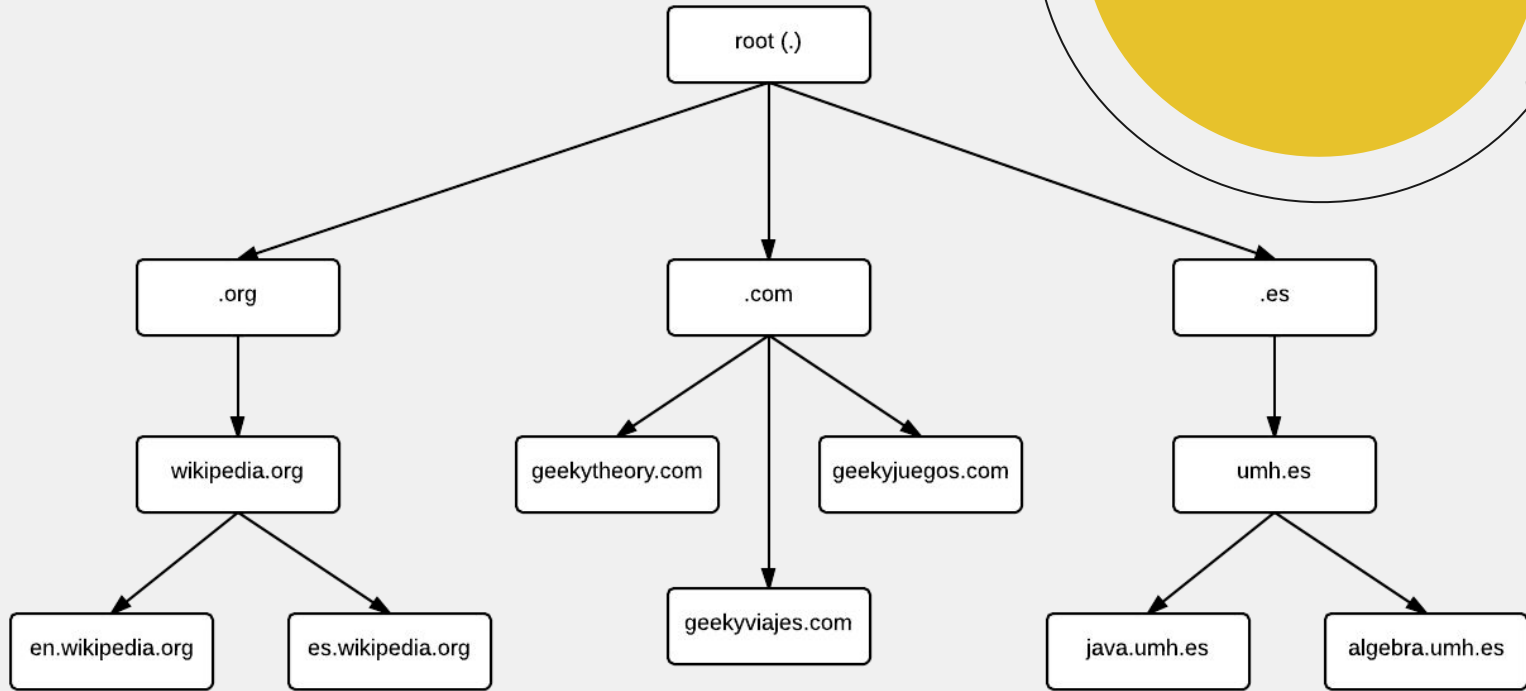


## 5.1 Qué significa DNS

### DNS: Domain Name System

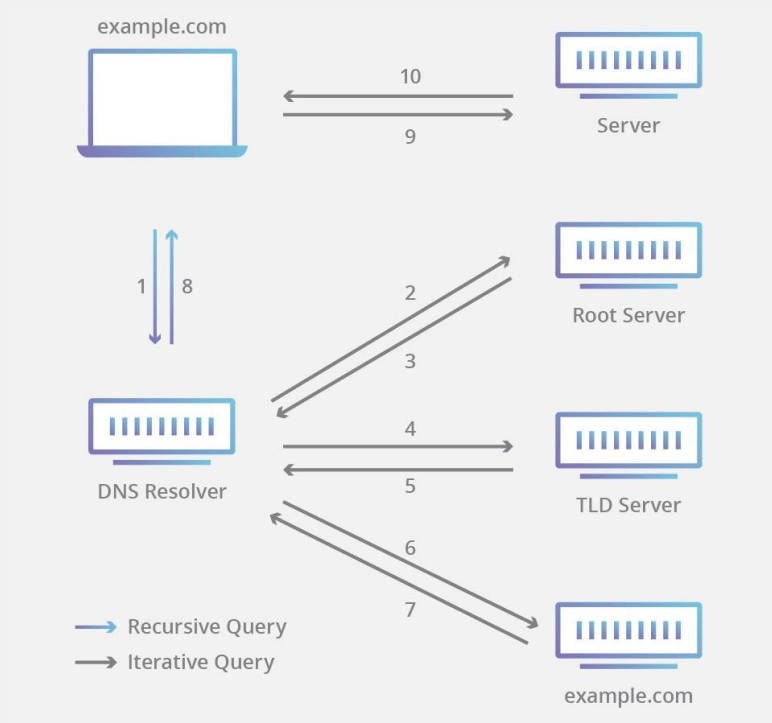
- Traduce las direcciones en lenguaje humano a direcciones IP
- BIND es el software de servidor DNS más utilizado





## 5.2 Estructura DNS

## 5.3 Pasos de una búsqueda DNS



- **Consulta recursiva**

Obliga a responder con el recurso solicitado o un mensaje de error

- **Consulta iterativa**

Permite derivar la consulta hasta encontrar la mejor respuesta



**FTP**



**FTP**





## 6.1 ¿Qué es FTP?

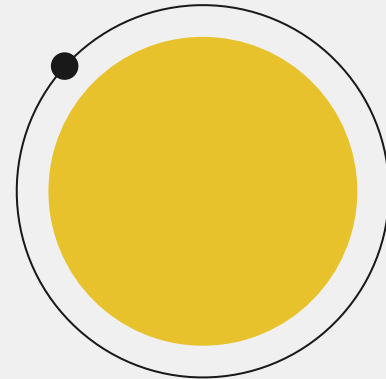
---

*Protocolo de Transferencia de Archivos.*

**Permite transferir archivos directamente de un dispositivo a otro.**

Funciona entre ordenadores que estén conectados a **Internet**.

Archivos se comparten de forma directa y sin ningún intermediario.



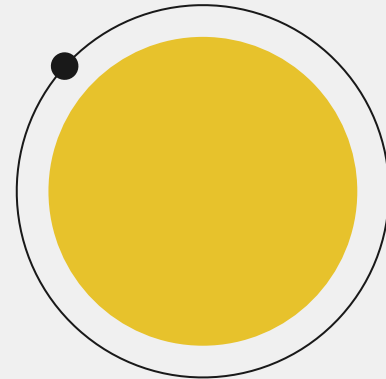




## 6.2 Usuarios y acceso anónimos

---

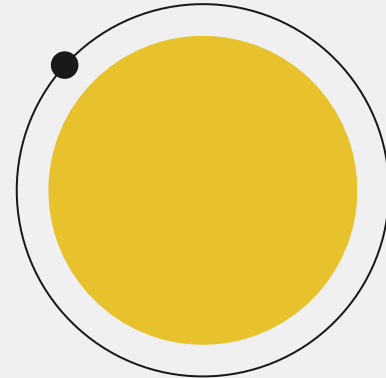
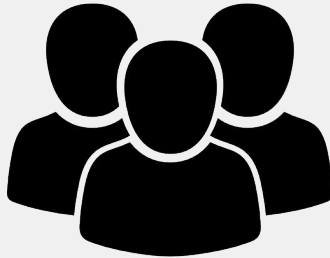
- **Usuario real:** establece sesiones en el servidor FTP.
- Permisos de acceso a archivos y directorios.
- Toda la estructura del disco estará visible.



## 6.2 Usuarios y acceso anónimos

---

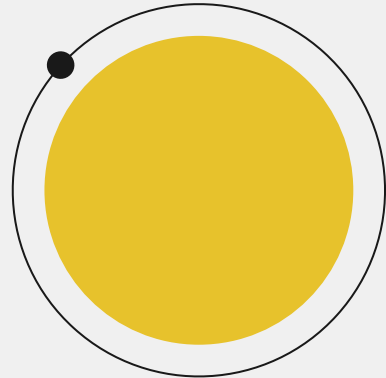
- **Usuarios invitados:** necesitan cuenta para iniciar sesión en el servidor FTP.
- No pueden hacer sesiones en terminales.
- Servidor FTP → restringe vista de estructura del disco del servidor.



## 6.2 Usuarios y acceso anónimos

---

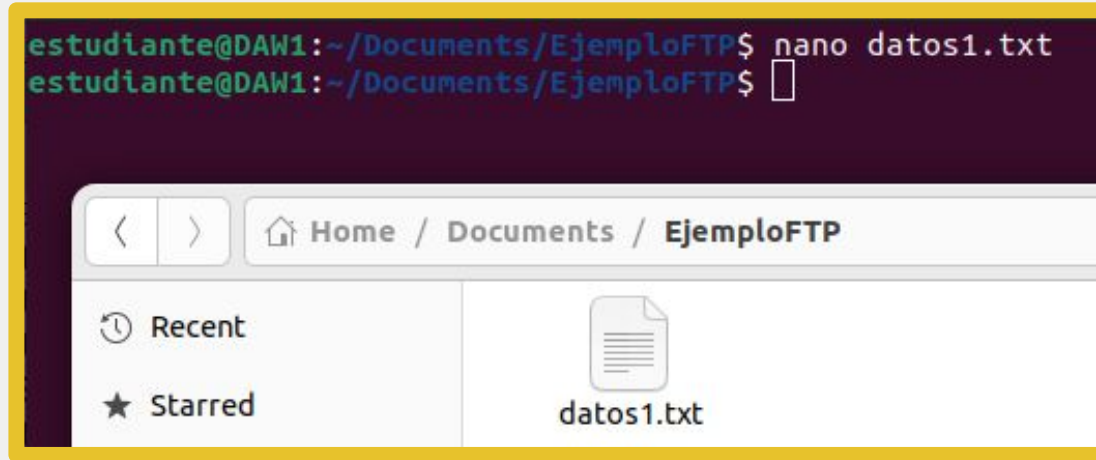
- **Usuarios anónimos:** inician sesión como anonymous
- Diferencia con usuario invitado → una única área del archivo compartida por todos los usuarios anónimos



## 6.2.1 Clientes en línea de comando

### 1º Creamos el repositorio

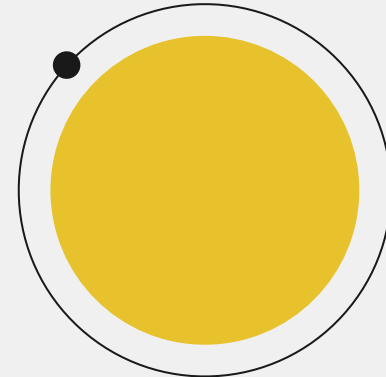
Creamos un repositorio llamado ***EjemploFTP*** y dentro de él creamos un fichero txt con el contenido que queramos.



The image shows a terminal window and a file explorer. The terminal window has a dark purple background and displays the following commands and output:

```
estudiante@DAW1:~/Documents/EjemploFTP$ nano datos1.txt
estudiante@DAW1:~/Documents/EjemploFTP$
```

The file explorer shows the directory structure: Home / Documents / EjemploFTP. It lists a file named 'datos1.txt' with a document icon. The interface includes a 'Recent' section and a 'Starred' section.





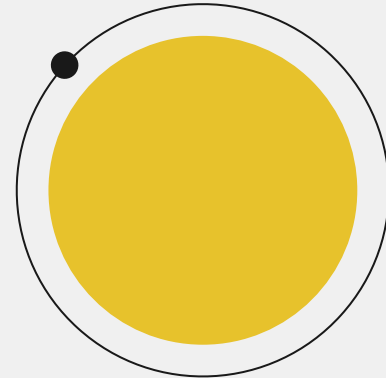
## 6.2.1 Clientes en línea de comando

---

### 2º Establecer conexión con servidor ftp de la red iris

Identificarse: usuario anonymous y sin contraseña.

```
estudiante@DAW1:~/Documents/EjemploFTP$ ftp
ftp> open
(to) ftp.rediris.es
Trying 130.206.13.2:21 ...
Connected to ftp.rediris.es.
220- Bienvenido al servicio de replicas de RedIRIS.
220- Welcome to the RedIRIS mirror service.
220 Only anonymous FTP is allowed here
Name (ftp.rediris.es:estudiante): anonymous
230- RedIRIS - Red Académica y de Investigación Española
230- RedIRIS - Spanish National Research Network
230-
230- ftp://ftp.rediris.es -==> http://ftp.rediris.es
230 Anonymous user logged in
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> █
```





## 6.2.1 Clientes en línea de comando

---

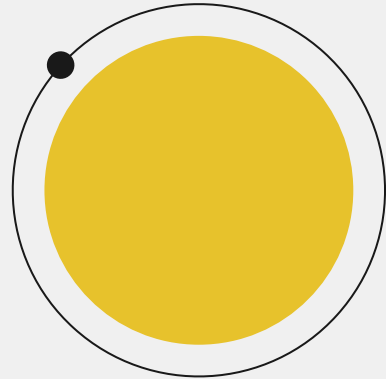
### 3º Descargar welcome.msg del servidor

Subir datos1.txt con put (como somos anonymous no nos deja subirlo al servidor)

Cerrar conexión → bye

```
ftp> get welcome.msg
local: welcome.msg remote: welcome.msg
229 Extended Passive Mode Entered (|||39794|)
150 Accepted data connection
100% |*****| 93 78.22 KiB/s 00:00 ETA
226-File successfully transferred
226 0.000 seconds (measured here), 0.90 Mbytes per second
93 bytes received in 00:00 (66.97 KiB/s)
ftp>
```

```
local: datos1.txt remote: datos1.txt
229 Extended Passive Mode Entered (|||30518|)
550 Anonymous users may not overwrite existing files
ftp> bye
221-Goodbye. You uploaded 0 and downloaded 1 kbytes.
221 Logout.
```

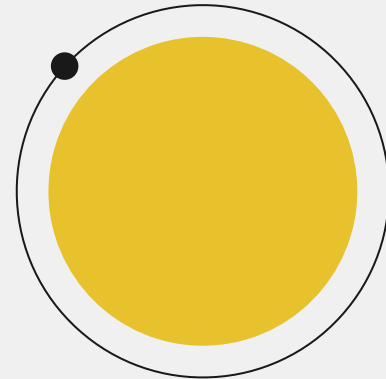




## 6.2.2 Clientes gráficos

---

- **FileZilla**
- **WinSCP**
- **Cyberduck**

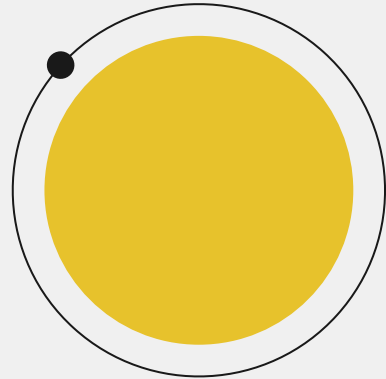


## 6.3 FTP Seguro



**Conexión autenticada** = ID usuario encriptado + contraseña encriptado + certificados

**Conexión FTPS** → cliente comprueba certificado de servidor firmado por una autoridad certificada







**¡Gracias!**

**¿Preguntas?**