

# CONCURSO TECNOLOGÍAS INFORMATICAS A

POSTULANTE A AYUDANTE GRADUADO  
REGULAR: **ING. MARIANO MOREL**

# FORMACIÓN Y EXPERIENCIA DOCENTE



## FORMACIÓN

- TÉCNICO E INGENIERO ELECTRÓNICO
- CURSO FULL-STACK



## CARGOS

- AYUDANTE GRADUADO EXCLUSIVO
- AYUDANTE GRADUADO SIMPLE



## DOCENCIA

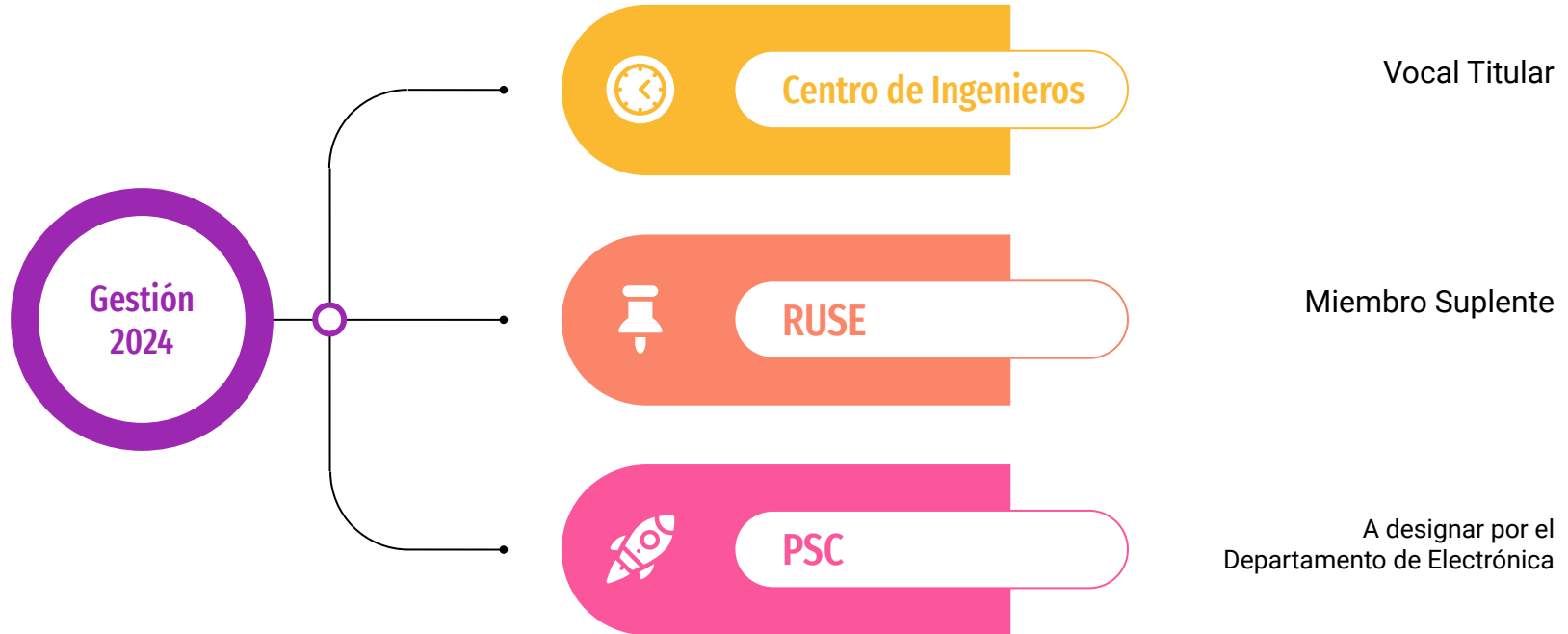
- PLAN FINES
- TEORÍA DE REDES I
- CIRCUITOS I
- TECNOLOGÍAS INFORMÁTICAS A
- PROYECTO TRANSVERSAL



## CAPACITACIÓN

- MAESTRÍA EN MICROELECTRÓNICA (UBA)

# GESTIÓN



# TEMAS A DESARROLLAR



## Conceptos de protocolo HTTP

HTTP es el protocolo de comunicación utilizado para transferir datos en la web

Programa que escucha en un puerto y responde a las solicitudes HTTP de los clientes



## Conceptos de Servidor HTTP



## Registros de Dominios

Proceso de adquirir un nombre de dominio único para un sitio web

Implica desde la creación del contenido hasta la configuración del servidor



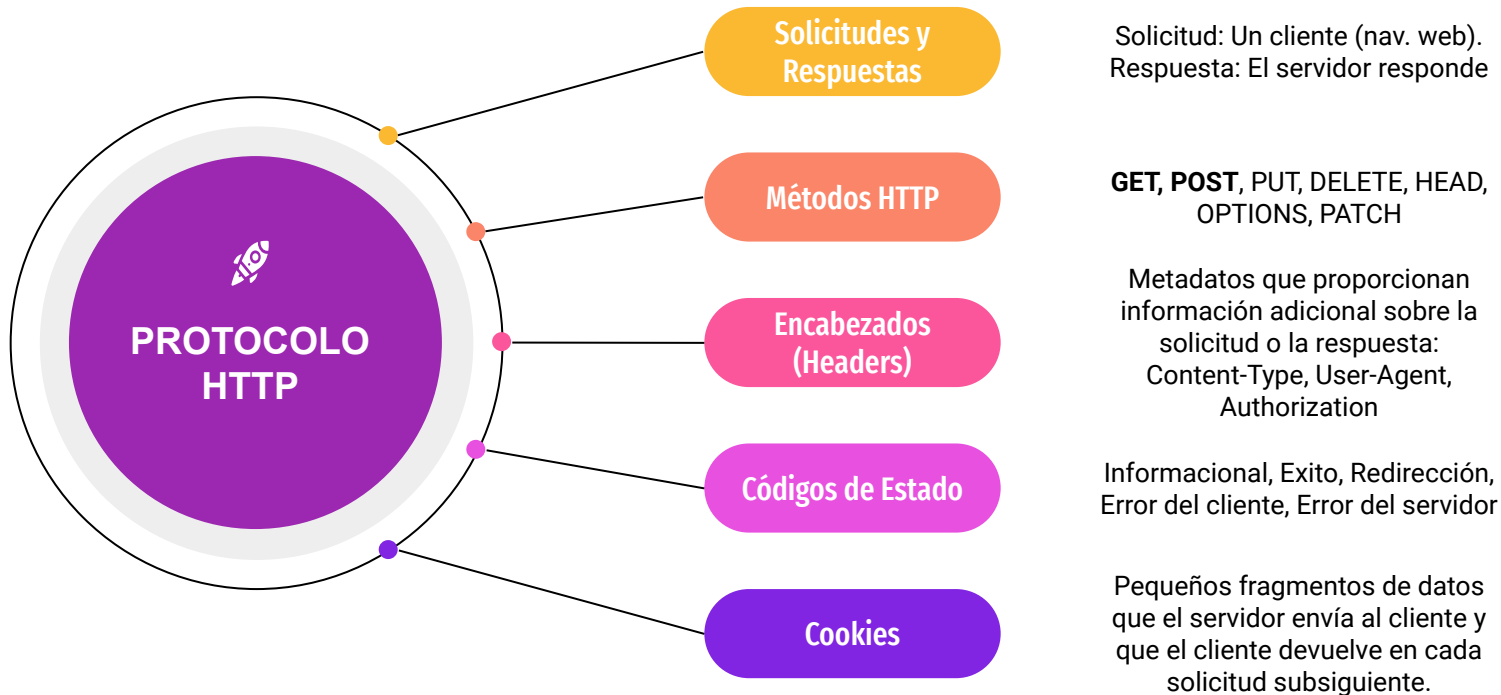
## Publicación de una web

### UNIDAD 3

# ¿QUE VIMOS HASTA AHORA? ¿HACIA DÓNDE VAMOS?



# 1) CONCEPTOS DE PROTOCOLO HTTP



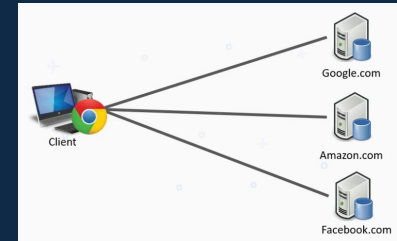
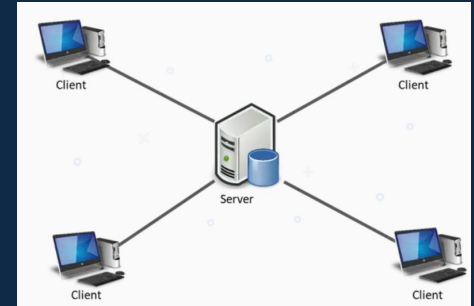
# CLIENTE - SERVIDOR: PARTE 1

**Este modelo permite la comunicación y el intercambio de datos entre múltiples dispositivos y aplicaciones**

**CLIENTE:** es un programa o dispositivo que solicita un servicio o recurso a un servidor.

## Funciones:

- Inicia la comunicación enviando una solicitud al servidor
- Procesa la respuesta recibida del servidor.
- Los clientes son aplicaciones de usuario final, como navegadores web, aplicaciones móviles, o software de escritorio. Ejemplos: Navegador web, aplicaciones de correo electrónico, aplicaciones de mensajería instantánea.



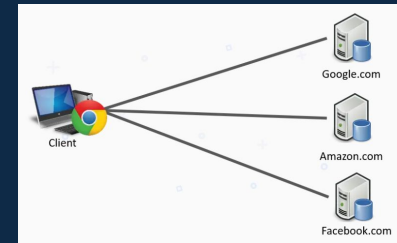
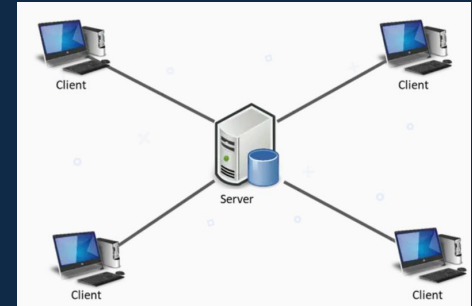
# CLIENTE - SERVIDOR: PARTE 2

**Este modelo permite la comunicación y el intercambio de datos entre múltiples dispositivos y aplicaciones**

**SERVIDOR:** es un programa o dispositivo que proporciona servicios o recursos a los clientes

## Funciones:

- Escucha y acepta solicitudes de clientes.
- Procesa las solicitudes y genera respuestas.
- Puede manejar múltiples solicitudes simultáneas.
- Almacena y gestiona datos, como bases de datos, archivos, y servicios web.
- Ejemplos: Servidores web (Apache, Nginx), servidores de bases de datos (MySQL, PostgreSQL), servidores de correo (SMTP, IMAP), servidores FTP.





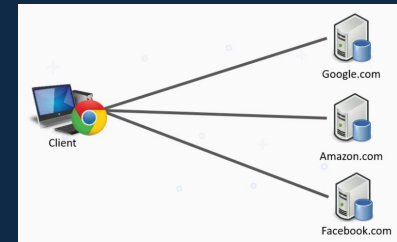
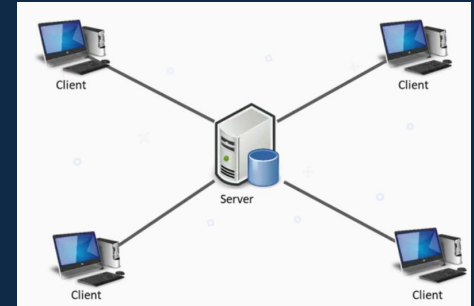
# CLIENTE - SERVIDOR: PARTE 3

## CONEXIÓN

**Establecimiento de Conexión:** El cliente inicia una conexión con el servidor. El servidor acepta la conexión y espera solicitudes.

**Solicitud y Respuesta:** El cliente envía una solicitud al servidor. El servidor procesa la solicitud y genera una respuesta. El servidor envía la respuesta al cliente. El cliente procesa la respuesta y la muestra al usuario.

**Cierre de Conexión:** Después de que el cliente recibe la respuesta, la conexión se cierra. El cliente puede iniciar una nueva conexión para solicitudes adicionales.



# CLIENTE - SERVIDOR: PARTE 4

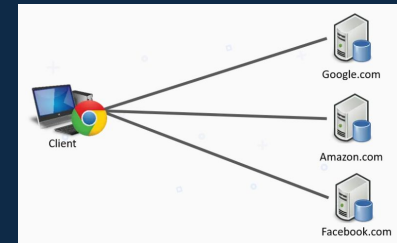
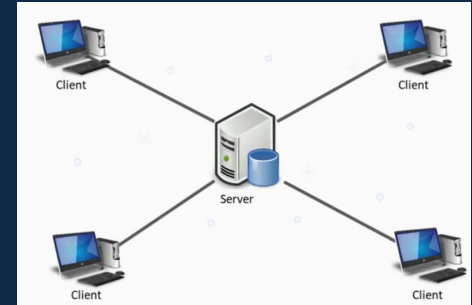
## VENTAJAS Y DESVENTAJAS

### VENTAJAS

- Escalabilidad: Los servidores pueden manejar múltiples clientes simultáneamente, mejorando el rendimiento y la disponibilidad.
- Centralización de Recursos: Los servidores pueden almacenar y gestionar recursos compartidos, como bases de datos y archivos.
- Separación de Responsabilidades: Los clientes pueden delegarse en la gestión de interfaces de usuario y la presentación de datos.

### DESVENTAJAS

- Latencia: Demora
- Seguridad: Los servidores deben implementar medidas de seguridad para proteger contra accesos no autorizados y ataques.
- Sobrecarga de Redes: Los servidores pueden volverse sobrecargados si no están bien dimensionados. Posible caída.



# DISEÑO DEL MODELO C-S



## Capas de Red

Las solicitudes y respuestas pueden pasarse a través de capas de red, como la capa de aplicación, de transporte, y de enlace.



## Seguridad

Se implementan medidas de seguridad, como autenticación y encriptación.



## Protocolo

Se utilizan protocolos específicos, como HTTP para web y SMTP para correo.

## Protocolo

**Conjunto formal de estándares y normas. Rigen tanto el formato como el control de la interacción entre distintos dispositivos en un sistema de comunicación. En internet (web) usamos el protocolo HTTP/HTTPS**

HTTP : "Protocolo de Transferencia de Hipertexto" //// SMTP: "Protocolo Simple de Transferencia de Correo"

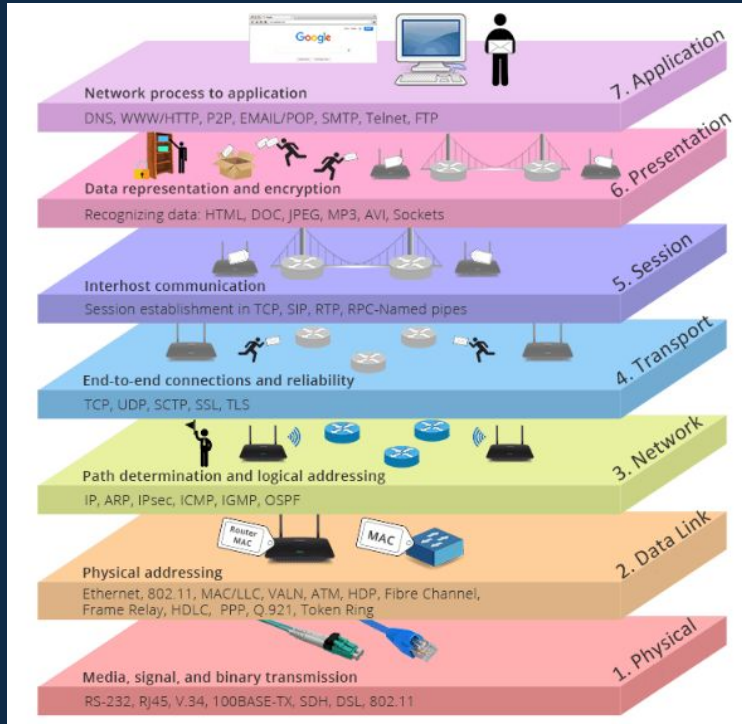
# MODELO OSI/TCP-IP

## RELACIÓN CON LOS PROTOCOLOS

Relación entre los Modelos OSI y TCP/IP					
Modelo OSI	Capa	Modelo TCP/IP	Capa	Funciones	Protocolos
7	Aplicación	Aplicación	1	Interfaz de usuario, manejo de datos y aplicaciones	HTTP, FTP, SMTP, DNS, Telnet
6	Presentación	Aplicación	1	Formato de datos, cifrado, compresión	SSL/TLS, MIME, JPEG, ASCII, EBCDIC
5	Sesión	Aplicación	1	Control de sesiones, sincronización, reanudación de sesiones	NetBIOS, RPC, PPTP
4	Transporte	Transporte	2	Transferencia de datos confiable, control de flujo, segmentación y reensamblaje	TCP, UDP
3	Red	Internet	3	Ruteo, direccionamiento lógico, fragmentación y reensamblaje	IP, ICMP, ARP, RARP
2	Enlace de Datos	Acceso a la Red	4	Control de acceso al medio, detección y corrección de errores	Ethernet, Wi-Fi (802.11), PPP, HDLC
1	Física	Acceso a la Red	4	Transmisión y recepción de bits brutos	Ethernet (cableado), Wi-Fi (radiofrecuencia), DSL, fibra óptica

# MODELO OSI - CAPAS

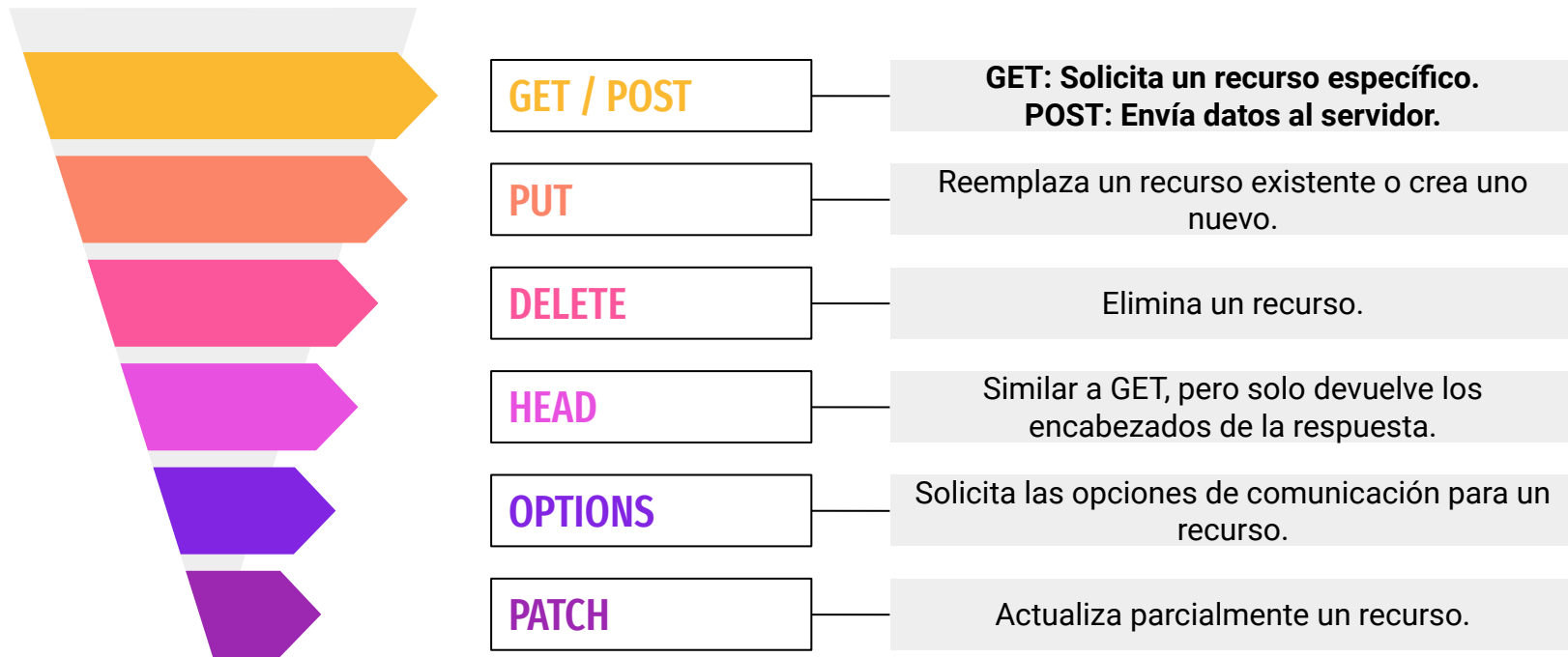
## RELACIÓN CON LOS PROTOCOLOS



### Ejemplo de Relación entre Protocolos y Capas

- **HTTP (Capa 7: Aplicación):** Protocolo de transferencia de hipertexto. Se utiliza para solicitar y recibir páginas web.
- **TCP (Capa 4: Transporte):** Protocolo de control de transmisión. Proporciona una conexión confiable y en orden.
- **IP (Capa 3: Internet):** Protocolo de internet. Se utiliza para ruteo y direccionamiento lógico.
- **Ethernet (Capa 2: Enlace de Datos):** Protocolo de enlace de datos. Se utiliza para la transmisión de datos a través de una red física.
- **DSL (Capa 1: Física):** Tecnología de línea digital suscriptor. Proporciona una conexión de datos a través de líneas telefónicas.

# MÉTODOS HTTP



**Métodos de petición para indicar la acción que se desea realizar para un recurso determinado**

# POST / GET

Los datos enviados se incluyen en el cuerpo de la solicitud HTTP. Los datos no son visibles en la barra de direcciones del navegador (S/LIMITE)

Se utiliza para solicitudes que modifican el estado del servidor, envían datos grandes o sensibles, o cuando no se desea que los datos sean visibles en la URL.



Los datos enviados se incluyen en la URL de la solicitud.

Los datos son visibles en la barra de direcciones del navegador (2048)

Cuando se busca información y los parámetros de búsqueda no cambian el estado del servidor.  
Datos pequeños y no sensibles

## 2) CONCEPTOS DE SERVIDOR HTTP

### Puertos

El puerto predeterminado para HTTP es el 80, y para HTTPS es el 443.



### Logs y Monitoreo

Logs: Registros de las solicitudes y respuestas.  
Depuración y análisis.  
Monitoreo: Supervisión del rendimiento del servidor.

### Host Virtual

Permite que un solo servidor físico aloje múltiples sitios web utilizando diferentes nombres de dominio



### Proxy

Servidor intermedio que reenvía solicitudes a otros servidores

### Seguridad

Autenticación: Verifica la identidad del cliente.  
Autorización: Controla el acceso a recursos.  
Encriptación: Encriptar las comunicaciones.



### 3) REGISTRO DE DOMINIOS

Una dirección única en Internet (por ejemplo, example.com).

**Dominio**



La parte final del dominio (por ejemplo, .com, .org, .net).

**TLD (Top-Level Domain)**



**Registrar**

Entidad que gestiona el registro de dominios (por ejemplo, GoDaddy, Namecheap)



**DNS (Sist. de Nombres de Dominio)**

Sistema que traduce nombres de dominio en direcciones IP.

# URL: Parte 1

**URL (Uniform Resource Locator):** es una dirección que especifica la ubicación de un recurso en la web y el método para acceder a él.

## Componentes de una URL:

- Scheme: Especifica el protocolo de red utilizado para acceder al recurso
- Subdominio: Parte del nombre de dominio que se utiliza para identificar diferentes secciones de un sitio web.
- Dominio: El nombre principal del host.
- TLD (Top-Level Domain): La parte final del nombre de dominio
- Puerto: El número del puerto en el que el servidor está escuchando (opcional o default).
- Ruta (Path): Especifica la ubicación del recurso en el servidor.
- Consulta: Una cadena de pares clave-valor que se utiliza para pasar datos al servidor (?).
- Fragmento (Fragment): Una sección específica de la página web (#)

# URL: Parte 2

URL (Uniform Resource Locator): es una dirección que especifica la ubicación de un recurso en la web y el método para acceder a él.

<https://www.example.com:8080/path/to/resource?query=parameter#fragment>

Scheme: `http`, `https`, `ftp`, `mailto`, `file`

Subdominio: `www`, `mail`, `blog`

Dominio: `example`

TLD: `.com`, `.org`, `.net`, `.edu`

Puerto: `8080`

Ruta: `/index.html`

Consulta: `?key1=value1&key2=value2`

Fragmento: `#section1`

# HOST – NOMBRE DE DOMINIO

**Host:** es un dispositivo o sistema que se conecta a una red. El host, es el servidor que almacena y sirve los recursos de un sitio web.

**Ejemplo: IP Address: 192.168.1.1**

**Hostname:** [www.example.com](http://www.example.com)

**Nombre de Dominio:** es una etiqueta de texto que identifica a un host o a un conjunto de hosts en Internet. Es una forma más legible y memorable de referirse a una dirección IP. Los nombres de dominio son gestionados por el Sistema de Nombres de Dominio (DNS).

**Ejemplo: Dominio de Nivel Superior (TLD): .com, .org, .net**

**Dominio de Segundo Nivel (SLD): example.com**

**Subdominio: www.example.com**

# DNS: Parte 1

**Domain Name System:** Es un sistema distribuido que traduce nombres de dominio en direcciones IP. Así, es más fácil escribir un nombre de dominio que una dirección IP numérica.

## Funcionamiento del DNS:

Cuando un cliente necesita acceder a un recurso en un servidor, envía una solicitud de resolución de nombres al servidor DNS. El servidor DNS busca la dirección IP correspondiente al nombre de dominio y la devuelve al cliente.

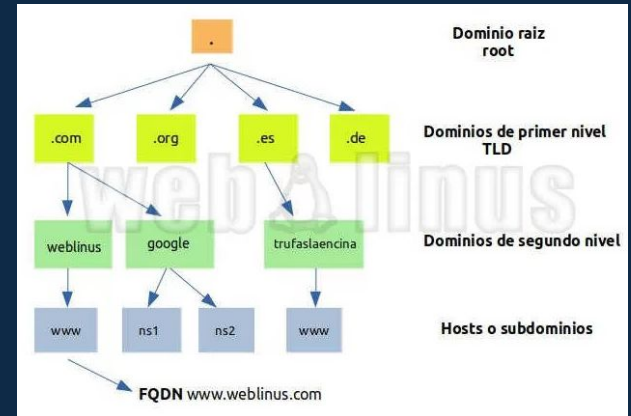
- **Caché:** Los servidores DNS y los clientes pueden almacenar en memoria caché las respuestas de resolución de nombres para mejorar el rendimiento.

- **Registros DNS:**

A: Mapea un nombre de dominio a una dirección IP v4.

CNAME: Alias de otro nombre de dominio.

Otros: MX, TXT, NS



# DNS: Parte 2

**Domain Name System:** Es un sistema distribuido que traduce nombres de dominio en direcciones IP. Así, es más fácil escribir un nombre de dominio que una dirección IP numérica.

## Ejemplo de Resolución DNS

**Cliente:** Un dispositivo necesita resolver el nombre de dominio www.example.com.

**Servidor DNS Recursivo:** El cliente consulta a un servidor DNS recursivo

**Servidor DNS Raíz:** El servidor recursivo consulta a un servidor DNS raíz para obtener la dirección del servidor TLD para .com.

**Servidor DNS TLD:** El servidor TLD para .com proporciona la dirección del servidor de autoridad para example.com.

**Servidor DNS de Autoridad:** El servidor de autoridad para example.com proporciona la dirección IP de www.example.com.

**Cliente:** El servidor recursivo devuelve la dirección IP a la solicitud del cliente, que luego puede usar para conectarse al servidor web.

# DNS: Parte 3

## Registración de DNS:

- **INTERNIC:** (Puede asignar nombres IP y resolver quejas/disputas) – Información pública  
<https://www.internic.net/>
- **IANA:** Autoridad de números asignados de internet – Responsable de coordinación de raíz DNS, direccionamiento IP y recursos de IP  
<https://www.iana.org/>
- **NICAR:** Dirección Nacional del registro de dominios de internet  
<https://nic.ar/>


# DNS: Parte 4

← → ↻

whatsmydns.net/#A/www.google.com

☆ 🔍 📺 🔊 🌐 📁 ⬇️ 👤 ⋮

🗖️ Outlook Mail FI Discord Portal marketing LdIn FI UBA YT Meet Drive HuggingChat Facebook Maestria c++ plc >>



whatsmydns.net  
Global DNS Propagation Checker

www.google.com

A ▾ ⚙️ 🔍 Search

Donate

🐦

f

🇺🇸 San Jose CA, United States  
Corporate West

🇺🇸 Dallas TX, United States  
Speakeasy

🇺🇸 Kansas City, United States  
WholeSale Internet

🇺🇸 Dothan AL, United States  
Comodo

🇺🇸 Atlanta GA, United States  
Speakeasy

🇨🇦 Quebec, Canada  
Cogeco Peer 1

🇲🇽 Mexico City, Mexico  
TotalPlay

A

AAAA

CNAME

MX

NS

PTR

SOA

SRV

TXT

CAA

142.250.189.228 ✓

✗

74.125.132.103  
74.125.132.104  
74.125.132.105  
74.125.132.106  
74.125.132.147  
74.125.132.99

108.177.122.103  
108.177.122.104  
108.177.122.105  
108.177.122.106  
108.177.122.147  
108.177.122.99

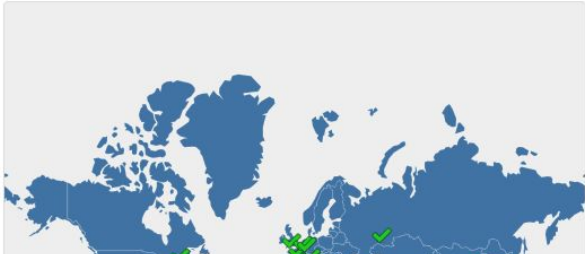
142.251.41.4 ✓

142.250.190.36 ✓

192.178.56.68 ✓

DNS Propagation Checker

whatsmydns.net lets you instantly perform a DNS lookup to check a domain name's current IP address and DNS record information against multiple nameservers located in different parts of the world.





## 4) PUBLICACIÓN DE UNA WEB: PARTE 1



# PUBLICACIÓN DE UNA WEB: PARTE 2

