

# Internet

---

## Conceptos básicos y terminología

- **Paquete:** Pequeña unidad de datos que se transmite a través de Internet.
- **Enrutador:** Dispositivo que dirige paquetes de datos entre diferentes redes.
- **Dirección IP:** Identificador único asignado a cada dispositivo en una red, utilizado para enrutar datos al destino correcto.
- **Nombre de dominio:** Nombre legible por humanos que se utiliza para identificar un sitio web, como google.com.
- **DNS:** (Domain Name Server) es responsable de traducir los nombres de dominio en direcciones IP.
- **HTTP:** (Hypertext Transfer Protocol) el Protocolo de transferencia de hipertexto se utiliza para transferir datos entre un cliente (como un navegador web) y un servidor (como un sitio web).
- **HTTPS:** Versión cifrada de HTTP que se utiliza para proporcionar una comunicación segura entre un cliente y un servidor.
- **SSL/TLS:** protocolos Secure Sockets Layer y Transport Layer Security se utilizan para proporcionar una comunicación segura a través de Internet.

## ¿Qué es internet?

Internet es una red global compuesta por miles de millones de dispositivos interconectados. Estos dispositivos pueden ser computadoras, teléfonos inteligentes, servidores, enrutadores y muchos otros. Cada uno de estos dispositivos tiene una dirección única llamada dirección IP, que se utiliza para identificarlos en la red.

## ¿Cómo funciona internet?

El funcionamiento de Internet se basa en el intercambio de información a través de una serie de protocolos y estándares. Uno de los protocolos fundamentales es el Protocolo de Internet (IP), que permite que los datos se dividan en pequeños paquetes y se envíen de una computadora a otra a través de la red.

Para que los dispositivos se comuniquen entre sí, utilizan enrutadores. Los enrutadores son dispositivos especializados que dirigen los paquetes de datos a través de la red, eligiendo la mejor ruta posible. De esta manera, los datos pueden viajar de un dispositivo a otro, incluso si están ubicados en diferentes partes del mundo.

## ¿Qué son y para qué se usan los protocolos?

Los protocolos son conjuntos de reglas y estándares que permiten que los dispositivos se comuniquen y compartan información entre sí. Establecen cómo se deben enviar, recibir y procesar los datos para lograr una comunicación efectiva y confiable.

Existen numerosos protocolos utilizados en diferentes ámbitos de la tecnología, pero aquí te presentaré algunos de los principales protocolos que se utilizan en Internet:

- **IP** (Protocolo de Internet): El Protocolo de Internet es la base de la comunicación en Internet. Permite que los dispositivos se identifiquen y se comuniquen entre sí a través de direcciones IP únicas. IPv4 e

IPv6 son las versiones más comunes de este protocolo.

- **SMTP** (Protocolo de Transferencia de Correo Simple): SMTP es el protocolo estándar para el envío de correos electrónicos. Permite que los servidores de correo electrónico se comuniquen entre sí y entreguen mensajes de correo electrónico a sus destinatarios.
- **FTP** (Protocolo de Transferencia de Archivos): FTP es un protocolo utilizado para la transferencia de archivos en redes. Permite que los archivos sean subidos o descargados entre un cliente y un servidor FTP.
- **SFTP** (Protocolo de Transferencia de Archivos Seguro): SFTP es una versión segura del protocolo FTP. Utiliza SSH (Secure Shell) para establecer conexiones seguras y encriptadas, protegiendo así la transferencia de archivos.
- **HTTP** (Protocolo de Transferencia de Hipertexto): HTTP es el protocolo utilizado para la transferencia de datos en la World Wide Web. Define cómo se solicitan y se entregan los recursos web, como páginas HTML, imágenes y archivos multimedia. HTTP utiliza los métodos GET y POST para enviar solicitudes y recibir respuestas.
- **HTTPS** (Protocolo de Transferencia de Hipertexto Seguro): HTTPS es la versión segura de HTTP. Utiliza SSL/TLS para cifrar la comunicación entre el navegador y el servidor, brindando seguridad y confidencialidad en la transferencia de datos.

## IPs y nombres de dominio

Las direcciones IP son identificadores únicos asignados a cada dispositivo conectado a una red. Existen dos versiones principales de direcciones IP: IPv4 e IPv6.

- **IPv4:** Es el sistema de direccionamiento más común. Consiste en una serie de cuatro números separados por puntos, como 192.168.0.1. Cada número puede variar entre 0 y 255, lo que da un total de aproximadamente 4.3 mil millones de direcciones posibles. Sin embargo, debido al crecimiento de Internet, las direcciones IPv4 **están agotándose**.
- **IPv6:** Es la versión más reciente del protocolo IP y fue desarrollada para solucionar la escasez de direcciones IPv4. Utiliza una notación hexadecimal y tiene una capacidad mucho mayor para asignar direcciones IP. Un ejemplo de una dirección IPv6 es 2001:0db8:85a3:0000:0000:8a2e:0370:7334.

Los **nombres de dominio** son identificadores alfanuméricos más fáciles de recordar que se utilizan para representar direcciones IP. Proporcionan una forma más amigable para que los usuarios accedan a sitios web y otros servicios en Internet. Por ejemplo, el nombre de dominio "google.com" se traduce en una dirección IP específica que indica dónde se encuentra almacenado el sitio web de Google.

- **Sistema de Nombres de Dominio (DNS):** El DNS es un sistema distribuido que asocia nombres de dominio con direcciones IP. Actúa como una especie de guía telefónica de Internet, traduciendo los nombres de dominio legibles para los humanos en direcciones IP que las computadoras puedan entender. Cuando escribimos un nombre de dominio en el navegador, el sistema DNS se encarga de encontrar la dirección IP correspondiente para que la página web pueda cargarse correctamente.

### ¿DNS y nombres de dominio es lo mismo?

Son conceptos relacionados pero distintos en el contexto de la infraestructura de Internet. El **DNS** es el sistema global y distribuido que **realiza la traducción de nombres de dominio a direcciones IP**.

Proporciona la infraestructura necesaria para que los nombres de dominio sean utilizados y accedidos en Internet. Por otro lado, el **nombre de dominio** es simplemente el **identificador alfanumérico que se utiliza**

**para representar una dirección IP** y permitir un acceso más fácil y amigable a los sitios web y servicios en línea.

## HTTP y HTTPS

HTTP (Protocolo de Transferencia de Hipertexto) es un protocolo utilizado para la transferencia de datos en la web. HTTPS es una versión más segura de HTTP que utiliza cifrado SSL/TLS para proteger la comunicación entre el navegador y el servidor.

## Protección en la comunicación: SSL/TLS

SSL (Capa de sockets seguros) y TLS (Seguridad de la Capa de Transporte) son protocolos de seguridad que encriptan los datos transmitidos entre el navegador y el servidor. Ayudan a proteger la integridad y confidencialidad de la información transmitida.

## VPN

Una VPN (Red Privada Virtual) permite establecer una conexión segura y encriptada a través de Internet, proporcionando privacidad y seguridad al navegar y acceder a recursos en línea. Puede ser útil en situaciones de uso diario, como proteger tus datos en redes Wi-Fi públicas, así como en casos específicos, como acceder a contenido bloqueado o establecer conexiones seguras para el teletrabajo.

## Arquitecturas

### Cliente-Servidor

La arquitectura cliente-servidor es un modelo en el que un cliente (como un navegador) realiza solicitudes a un servidor y el servidor responde a esas solicitudes proporcionando los recursos solicitados.

En la arquitectura cliente-servidor, hay dos componentes principales:

- **Cliente:** El cliente es una aplicación o dispositivo que solicita y consume servicios o recursos proporcionados por un servidor. Puede ser una aplicación web, una aplicación móvil, un navegador o cualquier otro software que interactúe con el servidor.
- **Servidor:** El servidor es una máquina o un software que provee servicios o recursos solicitados por los clientes. Puede ser un servidor web, un servidor de bases de datos, un servidor de correo electrónico, entre otros. El servidor responde a las solicitudes de los clientes y proporciona los datos o servicios solicitados.

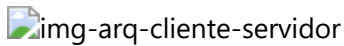
La comunicación entre el cliente y el servidor sigue un modelo de solicitud y respuesta:

1. El cliente envía una solicitud al servidor para acceder a un recurso o solicitar un servicio específico. La solicitud puede incluir información como parámetros, datos de entrada o instrucciones.
2. El servidor recibe la solicitud y la procesa. Puede realizar operaciones, acceder a bases de datos u otros recursos para cumplir con la solicitud del cliente.
3. Una vez que el servidor ha procesado la solicitud, envía una respuesta de vuelta al cliente. La respuesta puede contener los datos solicitados, el resultado de una operación o un mensaje de estado.

La arquitectura cliente-servidor se utiliza en una amplia variedad de aplicaciones y sistemas, desde aplicaciones web y servicios en la nube hasta bases de datos y servicios de correo electrónico. Algunas

características clave de esta arquitectura incluyen:

- **Escalabilidad:** La arquitectura cliente-servidor permite que múltiples clientes se conecten y utilicen los servicios proporcionados por un servidor. Esto permite que el sistema maneje una gran cantidad de solicitudes simultáneamente y se pueda escalar según las necesidades.
- **Centralización:** Los servidores actúan como puntos centrales para la administración y el suministro de recursos o servicios. Esto facilita el control y la gestión de los datos y servicios en un solo lugar.
- **Separación de responsabilidades:** En esta arquitectura, el cliente y el servidor tienen roles y responsabilidades claras. El cliente se encarga de la presentación de datos y la interacción con el usuario, mientras que el servidor se encarga del procesamiento y la gestión de los recursos.



## Otras arquitecturas

Además de la arquitectura cliente-servidor, existen otras arquitecturas como la arquitectura de tres capas (presentación, lógica de negocio y acceso a datos) y la arquitectura de microservicios.

- **Arquitectura de tres capas:** Esta arquitectura se divide en tres capas: la capa de presentación, la capa de lógica de negocio y la capa de datos. Cada capa tiene una responsabilidad específica. La capa de presentación se encarga de la interfaz de usuario, la capa de lógica de negocio maneja la lógica y el procesamiento de la aplicación, y la capa de datos se encarga del almacenamiento y acceso a los datos.
- **Arquitectura de microservicios:** En esta arquitectura, las aplicaciones se dividen en componentes pequeños y autónomos llamados microservicios. Cada microservicio se enfoca en una funcionalidad específica y puede ser desarrollado, desplegado y escalado de forma independiente. Los microservicios se comunican entre sí a través de interfaces bien definidas, lo que permite la construcción de sistemas escalables y flexibles.
- **Arquitectura orientada a servicios (SOA):** Esta arquitectura se basa en la idea de que las funcionalidades de una aplicación se ofrecen como servicios que pueden ser utilizados por otros componentes o aplicaciones. Los servicios son unidades de software independientes que se comunican a través de estándares y protocolos. Esta arquitectura promueve la reutilización, la interoperabilidad y la modularidad de los componentes de software.
- **Arquitectura sin servidor (Serverless):** En esta arquitectura, el proveedor de servicios en la nube se encarga de la infraestructura subyacente, como los servidores y el escalado automático, permitiendo a los desarrolladores centrarse en la lógica de negocio. Las aplicaciones sin servidor se basan en funciones o servicios que se ejecutan en respuesta a eventos, eliminando la necesidad de administrar servidores de forma manual.

## Frontend vs Backend vs FullStack

El frontend se refiere a la parte de una aplicación o sitio web que los usuarios ven y con la que interactúan. El backend se encarga del procesamiento de la lógica y el almacenamiento de datos. Un desarrollador full stack tiene habilidades tanto en frontend como en backend.

## Tecnologías/lenguajes

En el desarrollo web, se utilizan diversas tecnologías, como HTML (lenguaje de marcado para estructurar el contenido de una página web), CSS (hojas de estilo para dar estilo y diseño a la página) y JavaScript (lenguaje de programación para la interactividad en el navegador).

Frontend	Backend	Fullstack
HTML	JavaScript	JavaScript
CSS	Python	Python
JavaScript	Java	Java
TypeScript	PHP	PHP
React.js	Ruby	Ruby
Angular	C#	C#
Vue.js	Node.js	Node.js
Sass	Go	Go
Less	Kotlin	Kotlin
	Swift	Swift

## Frameworks

Un framework es una estructura de desarrollo que proporciona una base sólida y predefinida para construir aplicaciones de software. Está compuesto por un conjunto de herramientas, bibliotecas y componentes que facilitan el proceso de desarrollo al ofrecer funcionalidades comunes, estándares de codificación y patrones de diseño.

- **Eficiencia:** Los frameworks ofrecen una estructura y conjunto de herramientas que permiten a los desarrolladores trabajar de manera más eficiente. Proporcionan soluciones probadas y estandarizadas para tareas comunes, lo que reduce el tiempo y esfuerzo necesarios para desarrollar una aplicación desde cero.
- **Productividad:** Al ofrecer componentes reutilizables y funcionalidades predefinidas, los frameworks permiten a los desarrolladores centrarse en la lógica de negocio de la aplicación en lugar de tener que preocuparse por aspectos técnicos más complejos. Esto agiliza el proceso de desarrollo y permite una mayor productividad.
- **Escalabilidad:** Los frameworks están diseñados para facilitar el manejo de aplicaciones de mayor escala. Proporcionan estructuras modulares y flexibles que permiten agregar nuevas funcionalidades y adaptarse a medida que la aplicación crece en tamaño y complejidad.
- **Mantenibilidad:** Los frameworks promueven buenas prácticas de desarrollo, como la separación de responsabilidades y la estructuración del código en módulos. Esto facilita el mantenimiento y la evolución de la aplicación a lo largo del tiempo, ya que se fomenta un código más limpio, organizado y fácil de entender.

Frontend	Backend
React.js	Express.js
Angular	Django
Vue.js	Ruby on Rails
Ember.js	Flask

Frontend	Backend
Svelte	ASP.NET Core
Laravel	

## Hostings

Los hostings son servicios que permiten alojar y publicar sitios web en Internet. Proporcionan servidores donde se almacenan los archivos del sitio web y permiten que esté accesible para los usuarios en línea. Hay varios tipos de hosting disponibles, cada uno con características y niveles de control diferentes:

- **Hosting Compartido:** Es el tipo de hosting más común y económico. En un servidor compartido, múltiples sitios web comparten los recursos y configuraciones del servidor. Es una opción adecuada para sitios web pequeños o de bajo tráfico, ya que el rendimiento puede verse afectado si hay una alta carga en el servidor.
- **VPS (Servidor Privado Virtual):** Un VPS es un servidor virtualizado dentro de un servidor físico compartido. Proporciona mayor aislamiento y recursos dedicados en comparación con el hosting compartido. Permite un mayor control y personalización del entorno de alojamiento, pero requiere conocimientos técnicos para su configuración y administración.
- **Servidor Dedicado:** En este tipo de hosting, se alquila un servidor físico completo que se utiliza exclusivamente para un sitio web. Ofrece el máximo control y rendimiento, pero también conlleva una mayor responsabilidad de administración y un costo más alto.
- **Hosting en la Nube:** Esta opción aprovecha los recursos de varios servidores virtuales distribuidos en una red de servidores interconectados. Proporciona escalabilidad y flexibilidad, ya que los recursos pueden ajustarse según las necesidades del sitio web.

### Algunas empresas que ofrecen hosting

- De pago
  - Hostinger
  - Hostalia
  - Raiola Networks
  - Tecnocrática
- Gratuitos
  - Freehostia
  - 000webhost
  - Byethost

## Navegadores

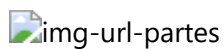
Los navegadores web son aplicaciones diseñadas para acceder y visualizar páginas web en Internet. Funcionan como intermediarios entre los usuarios y los servidores web que alojan los sitios web.

1. Google Chrome: Es el navegador más utilizado a nivel mundial, conocido por su velocidad y amplia compatibilidad con extensiones.
2. Mozilla Firefox: Es un navegador de código abierto que destaca por su enfoque en la privacidad y la personalización.

3. Safari: Es el navegador predeterminado en los dispositivos Apple, conocido por su rendimiento y optimización en el ecosistema de Apple.
4. Microsoft Edge: Es el navegador desarrollado por Microsoft y se basa en el proyecto de código abierto Chromium, al igual que Google Chrome.
5. Opera: Es un navegador con múltiples funciones, que ofrece características como un bloqueador de anuncios integrado y una VPN gratuita.

## URLs

Una URL (Uniform Resource Locator) es la dirección única que identifica una página web específica en Internet. Conocer cómo funciona una URL es útil porque permite comprender cómo acceder a los recursos en Internet, cómo enlazar y compartir páginas web, y cómo solucionar problemas relacionados con la navegación y la dirección de los sitios web. Además, entender las diferentes partes de una URL ayuda a interpretar los resultados de búsqueda, analizar el tráfico del sitio web y realizar optimizaciones de SEO.



- **Protocolo:** Especifica cómo se va a acceder al recurso. Los protocolos más comunes son HTTP (Hypertext Transfer Protocol) y HTTPS (HTTP Secure), que utiliza cifrado para mayor seguridad.
- **Dominio:** Es el nombre único que identifica un sitio web. Por ejemplo, en la URL "https://www.example.com", "example.com" es el dominio.
- **Subdominio:** Es una parte opcional de la URL que precede al dominio principal. Por ejemplo, "www" en "www.example.com".
- **Ruta:** Es la ubicación específica del recurso dentro del sitio web. Por ejemplo, "/blog/articulo" en "https://www.example.com/blog/articulo".
- **Parámetros:** Son información adicional que se puede incluir en la URL para personalizar la solicitud. Se indican después del símbolo de interrogación (?).
- **Fragmento:** Es una referencia a una sección específica de una página web. Se indica después del símbolo de almohadilla (#).