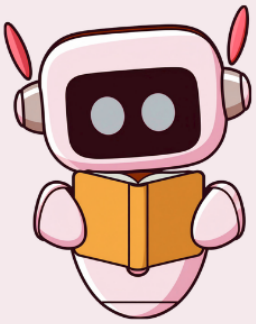
Especialización en
Programador Web

Introducción

Resumen introductorio de conceptos

Prof. Katherine Molina

Esta es una guía complementaria que reúne los conceptos básicos que necesitás entender antes de empezar a programar. Es un resumen de todo lo que venimos trabajando hasta ahora. No hace falta memorizar todo, usala como material de consulta a lo largo de la cursada.



Índice

Parte 1 — ¿Qué es Internet y cómo funciona?

1. ¿Qué es Internet?

Internet es una red mundial de computadoras conectadas entre sí que se comunican y comparten información. La palabra viene de *inter-connected networks* (redes interconectadas), no es una sola red, sino millones de redes más pequeñas unidas entre sí.

Un detalle importante: **Internet no es lo mismo que la Web**. Internet es la infraestructura física: los cables, antenas y servidores; la Web es solo **uno** de los servicios que funcionan sobre ella (otros son el correo electrónico, los videojuegos en línea, WhatsApp, etc.).

2. ¿Qué es el protocolo TCP/IP?

Un **protocolo** es un conjunto de reglas que dos computadoras acuerdan para entenderse, igual que dos personas necesitan hablar el mismo idioma.

TCP/IP es el "idioma" principal de Internet. En realidad, son dos protocolos que trabajan juntos:

- **IP** (Internet Protocol): se encarga de las **direcciones**, o sea de saber a dónde tiene que ir la información.
- **TCP** (Transmission Control Protocol): se encarga de **cortar la información en pedacitos** (llamados "paquetes"), enviarlos y volver a armarlos en el orden correcto al llegar, asegurándose de que no se pierda nada.

3. ¿Qué es la dirección IP?

Una **dirección IP** es el número único que identifica a cada dispositivo conectado a Internet. Sirve para que la información sepa a qué computadora exacta tiene que llegar.

> **Nota:** es como el número de documento (DNI) o la dirección de tu casa, pero para los dispositivos. Sin ella, los datos no sabrían a dónde ir.

Existen dos versiones:

- **IPv4:** cuatro números separados por puntos (ej. 142.250.78.14)
- **IPv6:** más larga y moderna, con letras y números (ej. 2a00:1450:4003:801::200e)

4. ¿Qué es la WWW?

WWW significa **World Wide Web** ("red informática mundial"). Es el conjunto de **páginas web** que podemos ver con un navegador, conectadas entre sí mediante **enlaces** (links).

> **Recordar:** Internet es la infraestructura, los cables y conexiones (el "todo"); **la Web** es uno de los servicios que viajan por esa infraestructura (una "parte")

La Web la inventó **Tim Berners-Lee** en 1989. Esas tres "w" que a veces aparecen al principio de una dirección (www.google.com) son justamente la herencia de ese nombre.

5. ¿Qué es el DNS?

DNS significa “Domain Name System” (Sistema de Nombres de Dominio). Es el sistema que **traduce los nombres que escribimos** (como google.com) **a la dirección IP** correspondiente (como 142.250.78.14).

¿Por qué existe? Porque las personas recordamos nombres, pero las computadoras se ubican con números. El DNS hace de traductor entre ambos.

> **Analogía.** Pensá en transferir plata. Cada cuenta tiene un CBU: un número larguísimo de 22 dígitos, imposible de recordar y muy fácil de equivocar al dictarlo (igual que una dirección IP). Por eso se inventó el alias: un nombre corto y fácil de recordar. Cuando alguien te transfiere usando el alias, el sistema lo traduce al CBU real por detrás. El DNS hace exactamente lo mismo: vos escribís google.com (el alias) y él lo traduce a la dirección IP (el CBU) sin que te tengas que enterar.

6. ¿Qué es un dominio?

Un **dominio** es el **nombre** que le ponemos a un sitio web para no tener que recordar su dirección IP. Es la dirección “amigable” que escribís en el navegador. Ejemplos: google.com, wikipedia.org.

El dominio (como el alias) es **único**: no pueden existir dos sitios con exactamente el mismo dominio en el mundo. Por eso hay que comprarlo o registrarlo.

7. Partes de un dominio

Tomemos como ejemplo www.tudominio.com.ar y veamos sus partes (se leen de derecha a izquierda según su jerarquía):

Parte	Ejemplo	Qué es
Dominio de nivel superior (TLD)	com	Indica el tipo o rubro del sitio (comercial, educativo, etc.).
Código de país	ar	Indica el país (Argentina). Es opcional.
Nombre (segundo nivel)	tudominio	El nombre propio que elige el dueño. Es la parte personalizable.
Subdominio	www	Una subdivisión dentro del sitio. Podría ser blog, tienda, mail, etc.

Algunos TLD frecuentes:

- .com → comercial | .org → organizaciones | .net → redes/tecnología
- .edu → educación | .gob / .gov → gobierno | .ar, .es, .mx → códigos de país

8. Tips para elegir el dominio correcto

Elegir un buen dominio es importante porque es la “primera impresión” del sitio. Algunos consejos:

- **Corto y fácil de recordar.** Cuanto más simple, mejor.
- **Fácil de escribir y pronunciar.** Evitá palabras que se escriban distinto a como suenan.
- **Relacionado con el contenido o la marca.** Que dé una pista de qué vas a encontrar.
- **Evitá números y guiones.**
- **Elegí bien el TLD.** .com es el más reconocido; usá .com.ar si tu público es de Argentina.
- **Verificá que esté disponible** y que no infrinja marcas registradas.
- **Pensá a futuro:** que sirva, aunque el proyecto crezca o cambie un poco.

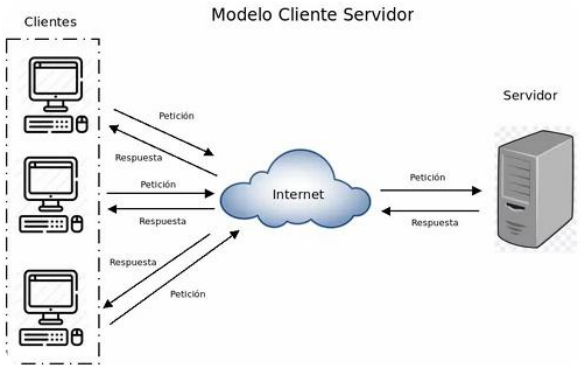
Parte 2 — La comunicación: el modelo cliente-servidor

9. El modelo cliente-servidor

La mayor parte de Internet funciona con el **modelo cliente-servidor**. Es la base para entender cómo “conversan” dos computadoras en la web.

- **Cliente:** es quien **pide** algo. Normalmente es tu dispositivo con el navegador (la computadora, el celular).
- **Servidor:** es una computadora (potente y siempre encendida) que **guarda** los sitios web y **responde** a los pedidos enviando lo que se le solicita.

El cliente **pide** y el servidor **sirve** (de ahí el nombre). Esa conversación de ida y vuelta es justamente lo que vamos a ver con HTTP, el Request y el Response.



10. Servidor y hosting

Profundicemos en el “otro lado” de la conversación:

- **Servidor:** es la computadora (o programa) que **almacena los archivos** de un sitio web y los **entrega** cuando un cliente los pide. Está encendido las 24 horas para que el sitio esté siempre disponible.
- **Hosting (alojamiento):** es el **servicio de alquilar un espacio en un servidor** para guardar ahí tu sitio web y que esté disponible en Internet. Lo brindan empresas especializadas.

> **Analogía.** el **hosting es como alquilar un local o un terreno**, y tu sitio web es el negocio que construís adentro. El **dominio** es el **cartel con la dirección** que le ponés a ese local para que la gente lo encuentre.

> **Para publicar un sitio en Internet necesitás dos cosas:** un **dominio** (el nombre) y un **hosting** (el espacio donde vive). Van siempre de la mano.

11. El protocolo HTTP

HTTP significa “HyperText Transfer Protocol” (Protocolo de Transferencia de Hipertexto). Es el conjunto de reglas que usan el **navegador** y el **servidor** para comunicarse y enviarse páginas web.

Cada vez que entrás a un sitio, tu navegador “habla” HTTP con el servidor: le **pide** la página y el servidor le **responde** con ella.

HTTPS es la versión **segura** de HTTP (la “S” es de *Secure*). Cifra la información para que nadie en el medio pueda leerla. Es la que se usa hoy en casi todos los sitios, y por eso ves el candado en el navegador. Fundamental cuando se manejan contraseñas o datos de tarjetas.

12. La petición (Request)

El **Request** (petición o solicitud) es el **mensaje que el navegador le envía al servidor** para pedirle algo: una página, una imagen, un video, etc. Una petición incluye, entre otras cosas:

- **El método:** qué quiere hacer (por ejemplo GET para *pedir/leer* información, POST para *enviar* información como un formulario).
- **La URL:** qué recurso quiere (qué página o archivo).
- **Encabezados (headers):** datos extra, como qué navegador se está usando o el idioma preferido.

13. La respuesta (Response)

El **Response** (respuesta) es el **mensaje que el servidor le devuelve al navegador** después de recibir la petición. Trae lo que se pidió (o un aviso de error). Una respuesta incluye:

- **El contenido:** el código HTML, la imagen, etc.
- **Un código de estado** que indica qué pasó. Los más conocidos: 200 (todo bien), 404 (no encontrado / la página no existe), 500 (error del servidor).

> Cuando escribís google.com → el **DNS** lo traduce a una **IP** → el navegador manda un **Request HTTP** a esa IP → el servidor devuelve un **Response** con la página → ¡la ves en pantalla! Todo esto ocurre en menos de un segundo.

Parte 3 — Conceptos clave de la Web

14. Navegador vs. Buscador

Es una de las confusiones más comunes. **No son lo mismo.**

- **Navegador (browser):** es el **programa** que instalás para entrar a Internet y ver páginas web. Ejemplos: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari.
- **Buscador (search engine):** es un **sitio web** que usás “dentro” del navegador para **encontrar** otras páginas. Ejemplos: Google, Bing, DuckDuckGo.

15. Página web vs. Sitio web

- **Página web:** es **un solo documento**, una pantalla individual (por ejemplo, la página de “Contacto”).
- **Sitio web:** es el **conjunto de páginas** relacionadas que están bajo un mismo dominio y forman un todo.

Ejemplo: wikipedia.org es un **sitio web**, y el artículo sobre “Argentina” es una **página** dentro de ese sitio.

16. Páginas estáticas vs. dinámicas

- **Página estática:** muestra **siempre el mismo contenido** para todos los visitantes. Solo cambia si el programador edita el código a mano. Son rápidas y simples (ej.: la página institucional de una panadería con su dirección y horarios).
- **Página dinámica:** el contenido **cambia** según el usuario, la fecha, la base de datos, etc. Se genera “al momento” (ej.: tu muro de Facebook, un carrito de compras, tu home de Netflix).

17. Tipos de páginas web según su funcionalidad

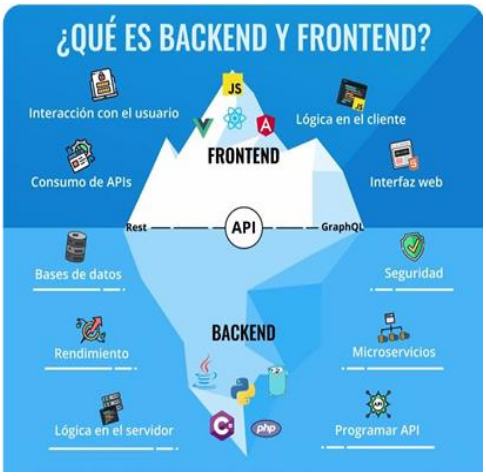
Según para qué se usan, podemos clasificarlas en categorías como:

- **Institucionales / corporativas:** presentan una empresa o marca (quiénes son, qué hacen).
- **Blogs:** publican artículos o notas ordenados por fecha.
- **Tiendas online (e-commerce):** venden productos o servicios (ej. Mercado Libre).
- **Portales de noticias:** informan y se actualizan constantemente.
- **Redes sociales:** conectan personas (Instagram, X, Facebook).
- **Educativas / e-learning:** ofrecen cursos y contenidos (ej. la plataforma de tu facultad).
- **Buscadores:** ayudan a encontrar otros sitios (Google).
- **Landing pages:** una sola página enfocada en un objetivo (vender un producto, captar un contacto).
- **Aplicaciones web (web apps):** funcionan como un programa dentro del navegador (Gmail, Google Docs, Trello).

Parte 4 — ¿Cómo se construye un sitio por dentro?

18. Front-end y Back-end

Todo sitio web tiene **dos partes** que trabajan juntas:



- **Front-end (el "frente"):** es **todo lo que el usuario ve y toca**: los textos, los colores, los botones, las imágenes, los menús. Es la parte visual, la que corre en el **navegador** (lado del cliente). Se construye con **HTML** (estructura), **CSS** (estilo) y **JavaScript** (comportamiento).
- **Back-end (lo de "atrás"):** es **todo lo que el usuario NO ve**: la lógica, los cálculos, la seguridad y la conexión con la base de datos. Funciona en el **servidor**. Usa lenguajes como PHP, Python, JavaScript (Node.js), Java, etc.

> **Analogía.** en un restaurante, el **front-end es el salón**: las mesas, la decoración, la carta, el mozo (todo lo que el cliente ve). El **back-end es la cocina**: donde se preparan los platos y el cliente no entra, pero sin ella nada funcionaría.

19. Base de datos

Una **base de datos** es un sistema donde se **guarda y organiza la información** de un sitio de forma ordenada, para poder consultarla, modificarla o borrarla cuando haga falta.

¿Qué guarda? Por ejemplo: los usuarios y contraseñas, los productos de una tienda, los comentarios de un blog, los mensajes de un chat. Cuando una página es **dinámica**, casi siempre hay una base de datos detrás alimentándola con información actualizada. El **back-end** es el que se conecta con ella.

> **Analogía.** es como un **gran archivero o una biblioteca muy ordenada**. La información está clasificada en estantes y cajones (tablas) para poder encontrar rápido lo que buscás, en lugar de tener todo amontonado.

Ejemplos de sistemas de bases de datos: MySQL, PostgreSQL, MongoDB.

Parte 5 — Antes de programar: planificación y diseño

20. Arquitectura web

La **arquitectura web** es la forma en que se **organiza y estructura** un sitio: cómo se ordenan las páginas, cómo se conectan entre sí y cómo el usuario navega de una a otra. Una buena arquitectura hace que el usuario **encuentre lo que busca rápido y sin perderse**.

> **Analogía.** es como el **plano de una casa**. Antes de construir, definís dónde va cada ambiente y cómo se conectan los pasillos. En la web, definís qué páginas hay (Inicio, Productos, Contacto…) y cómo se llega de una a otra.

También incluye la parte técnica: cómo se relacionan el **cliente** (el navegador del usuario) y el **servidor** (donde vive el sitio).

21. Diseño UX / UI

Son dos conceptos que van de la mano pero **no son lo mismo**:

- **UX** (*User Experience* — Experiencia de Usuario): se ocupa de **cómo se siente** usar el sitio. ¿Es fácil? ¿Es claro? ¿El usuario logra lo que quiere sin frustrarse? Es la parte **funcional y de comodidad**.
- **UI** (*User Interface* — Interfaz de Usuario): se ocupa de **cómo se ve** el sitio. Los colores, los botones, las tipografías, los espacios. Es la parte **visual y estética**.

Un buen sitio necesita **las dos cosas**: que se vea bien y que sea fácil de usar.

22. Prototipado

El **prototipo** es una **maqueta o boceto** del sitio que se hace **antes de programar**, para visualizar cómo va a quedar y probar ideas sin gastar tiempo escribiendo código. Sirve para acordar con el cliente o el equipo cómo se verá y funcionará, y detectar problemas temprano (cuando todavía es barato cambiarlos). Existen distintos niveles:

- **Wireframe (baja fidelidad):** un boceto simple, en blanco y negro, con cuadrados y líneas que indican dónde va cada cosa.
- **Mockup (alta fidelidad):** una versión más realista, ya con colores, imágenes y tipografías, parecida al diseño final.
- **Prototipo interactivo:** una maqueta donde ya se puede “hacer clic” y navegar, para simular la experiencia real.

Herramientas típicas: Figma, Adobe XD, Balsamiq, o incluso papel y lápiz.