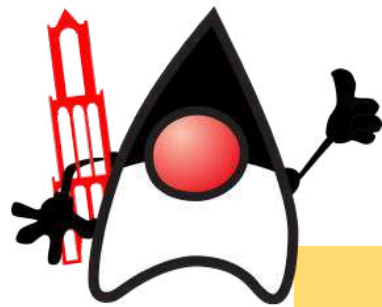
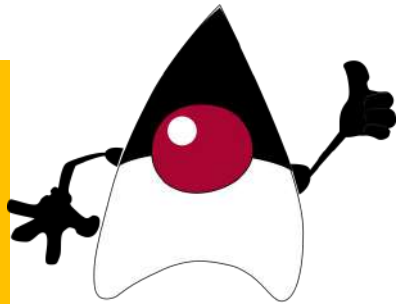




# Curso FullStack Python

Codo a Codo 4.0



# Duración del Curso



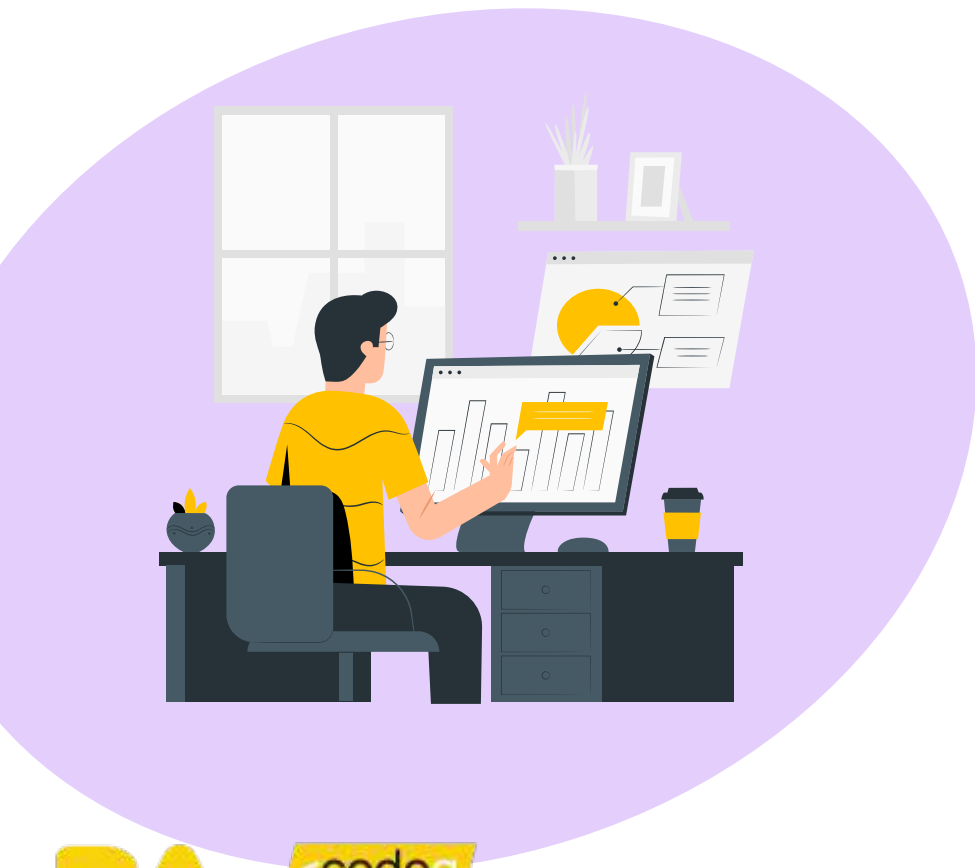
- 20 semanas
- Comienza en Marzo y finaliza a fines de Julio

## Documentación a Presentar

- Documento Identidad (Frente)
- Certificados de Estudios (Frente)

**Todos los documentos escaneados deberán ser enviados a su profesor.**








# 01

## OBJETIVOS Y LINEAMIENTOS DEL CURSO



- 
- 
- 
- **Brindar las herramientas necesarias para que el alumno pueda insertarse laboralmente en el mundo tecnológico.**
  - Este curso está dirigido a cualquier persona que ya posea conocimientos de programación y desee incursionar en el mundo de la programación web.
  - El curso tiene una duración de 20 semanas. Siendo 3 horas semanales de clase en vivo y 7 horas de dedicación fuera del horario de clase por parte de los alumnos (recomendado).



# 02

**ENCUESTA** AL  
CURSO



- **Link:** <https://forms.gle/kju8ETP5TfMmToz38>



# 03

**Desarrollador**  
FullStack



`<codo  
codo/>`



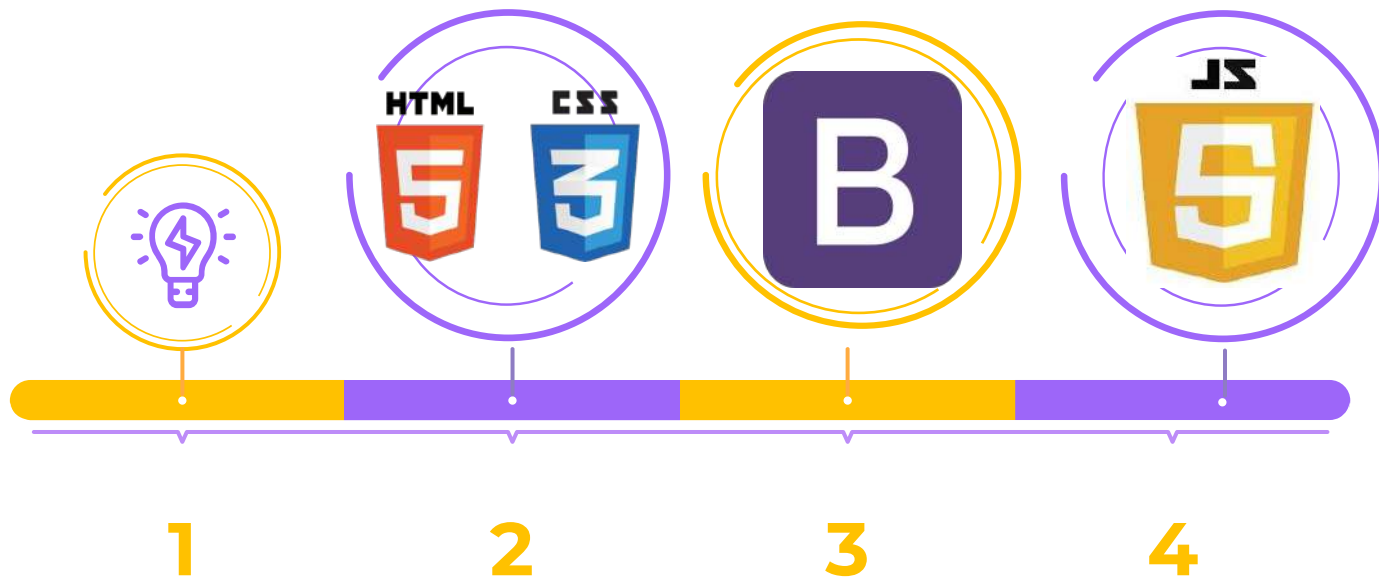
- **Programador con perfil técnico muy completo.**
- Encargado de manejar cada uno de los aspectos relacionados con la creación y el mantenimiento de una aplicación web.
- Es un **programador multiusos** y dentro del desarrollo del proyecto es responsable del montaje de los servidores, hasta el diseño.
- Es **fundamental que tenga conocimientos en desarrollo Front-End y Back-End.**
- Como desarrollador Front-end debería dominar, al menos, las siguientes tecnologías: HTML, CSS, JavaScript, Bootstrap.
- Como desarrollador Back-end debería tener conocimientos en: algún lenguaje de programación (Python, Java, PHP, etc.), algún framework y base de datos.



# Lenguajes y Frameworks en Desarrollo Web



# LO QUE VIENE





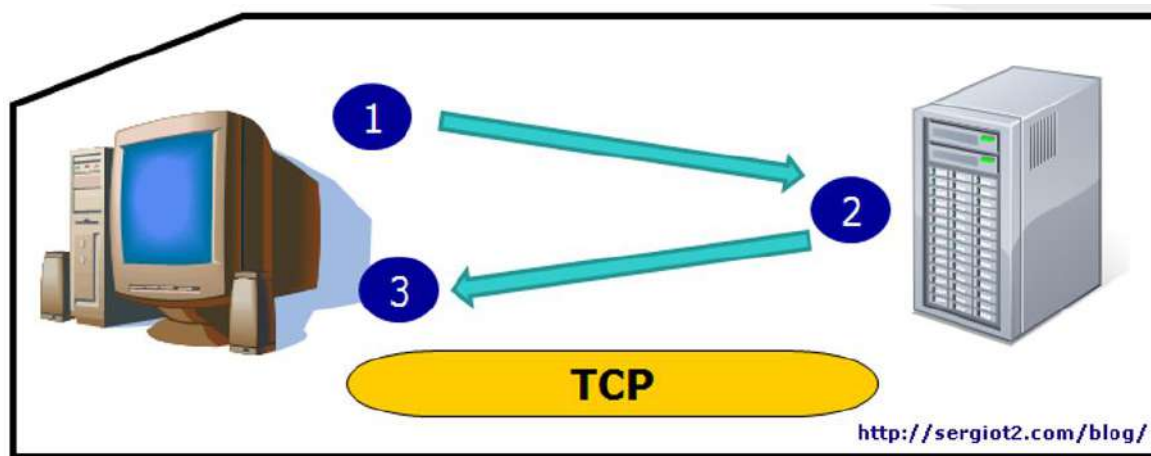
# 04

## **BASES** DEL DESARROLLO WEB (introducción)

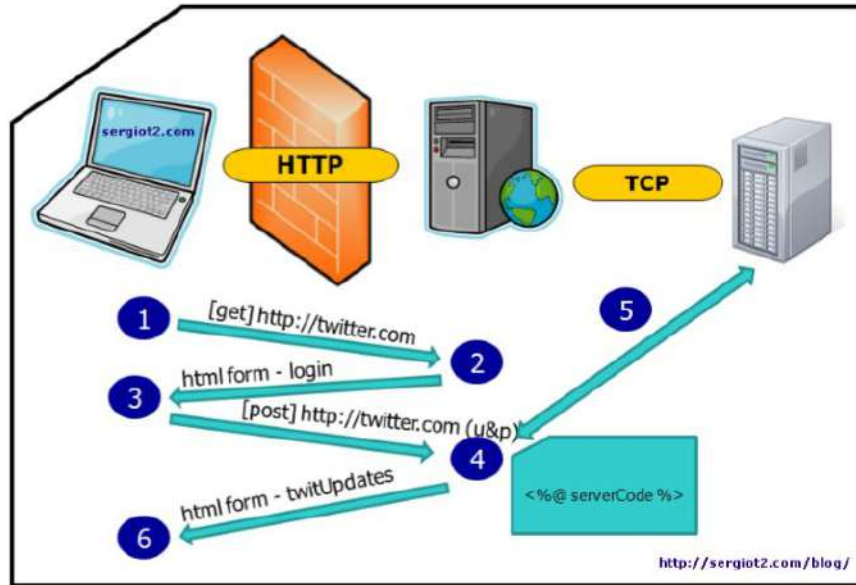


# Aplicación Escritorio

1. El usuario carga la aplicación.
2. La aplicación se conecta a la base de datos para pedir información o para modificarla.
3. La aplicación muestra al usuario la información solicitada.



# Aplicación web



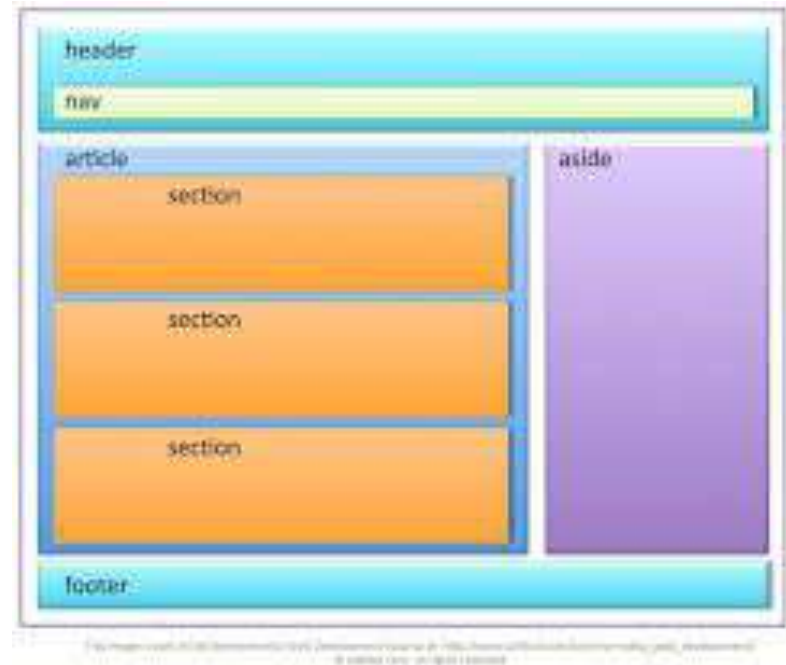
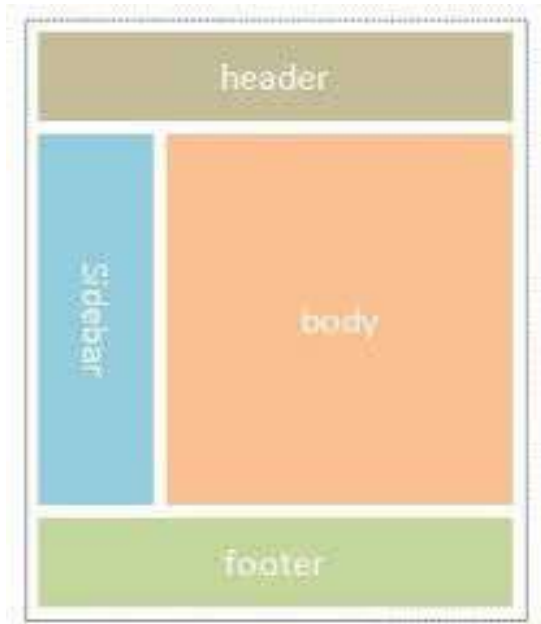
- El usuario ingresa la URL (dirección) en el navegador.
- El servidor web recibe la solicitud y envía la respuesta al navegador.

# Ventajas de una Aplicación web

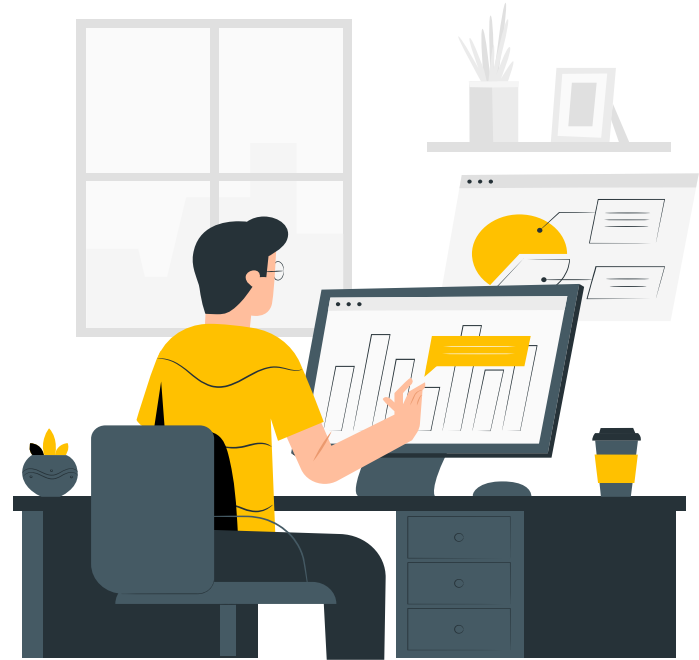
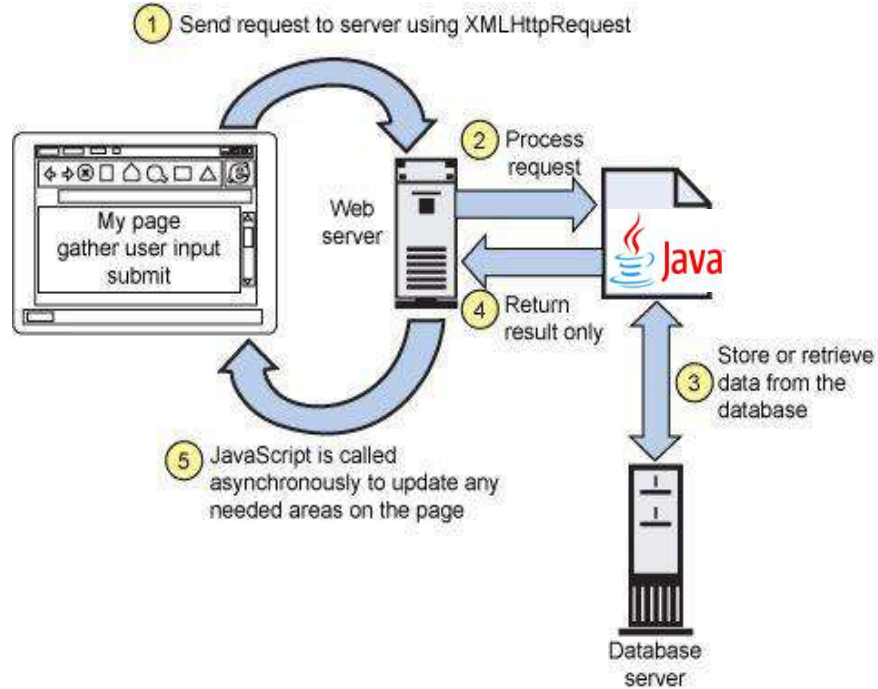
- Se puede ejecutar desde cualquier dispositivo que cuente con navegador y con conexión a internet. No es necesario instalarlas en el dispositivo, basta con un navegador web.
- Son fáciles de actualizar y mantener.
- Su funcionalidad es independiente al Sistema Operativo instalado en la PC del usuario.
- Las páginas web tienen estructuras similares que mejoran la experiencia del usuario al interactuar con una página.



# Ejemplo estructura de un Página Web



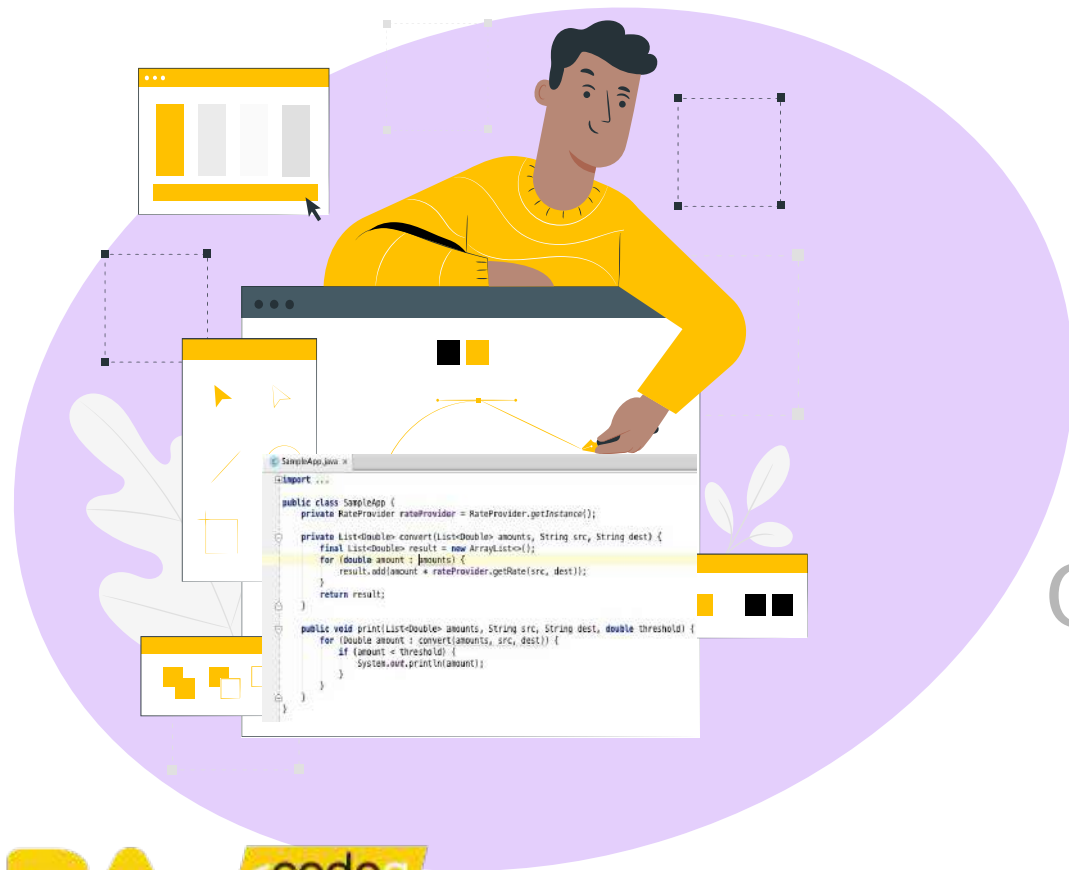
# Cliente/Servidor





# BACKEND FRONTEND





# 05

## HERRAMIENTAS QUE UTILIZAREMOS



# Herramientas que deberás instalar

- **Navegador:** Chrome, Firefox, Microsoft Edge (ex Internet Explorer).
- **Editor de Texto :** Notepad++ <https://notepad-plus-plus.org/>, SublimeText, o cualquier otro editor de texto que sea liviano y fácil de instalar.
- **IDE:** VSCode <https://code.visualstudio.com/>, Geany, Atom, Kompozer, etc.
- **Nota:** estas herramientas son sugeridas. Por supuesto podrás utilizar la que desees.



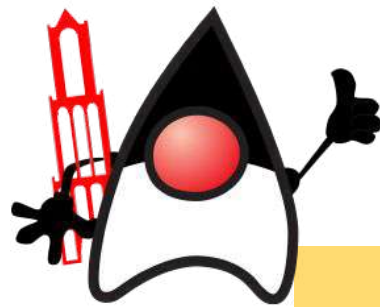
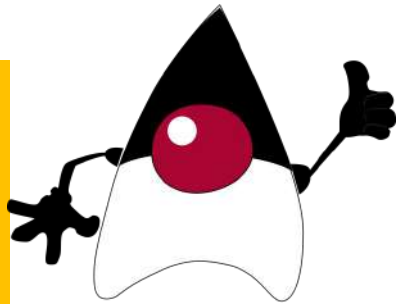
# MUCHAS GRACIAS!





# Curso FullStack Python

Codo a Codo 4.0



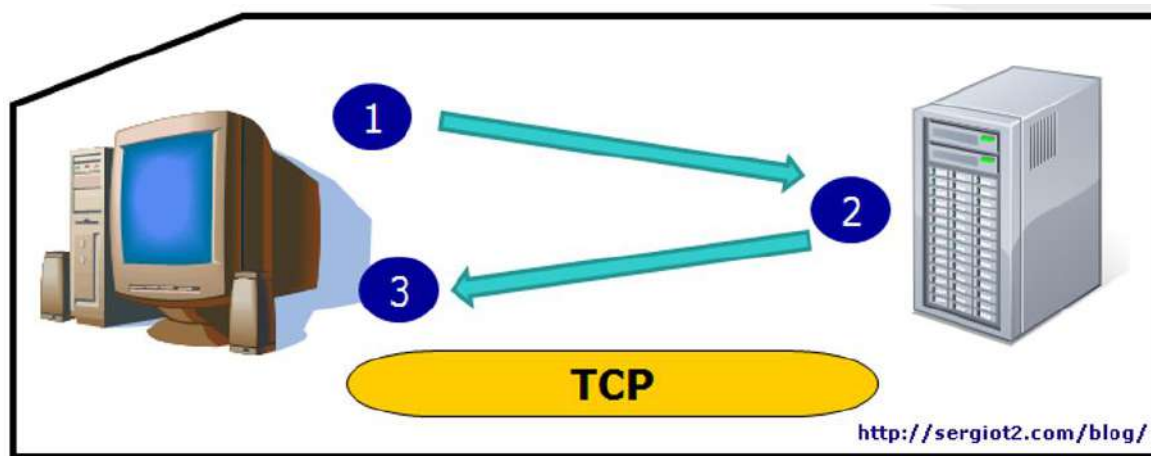


# BASES DEL DESARROLLO WEB



# Aplicación Escritorio

1. El usuario carga la aplicación.
2. La aplicación se conecta a la base de datos para pedir información o para modificarla.
3. La aplicación muestra al usuario la información solicitada.



# Aplicación web

1

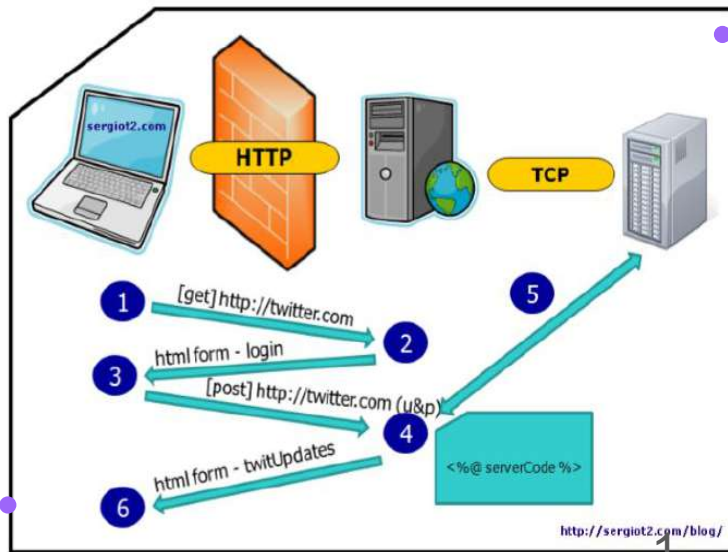
El usuario ingresa la URL (dirección) en el navegador.

2

El servidor web recibe la solicitud y envía la respuesta en html al navegador.

3

En el navegador se ingresa la información de usuario y contraseña



4

Dicha información se traduce a html.

5

Dicha información se traduce a html.

6

Si estos datos son válidos el usuario es redireccionado a la página que desea ingresar.

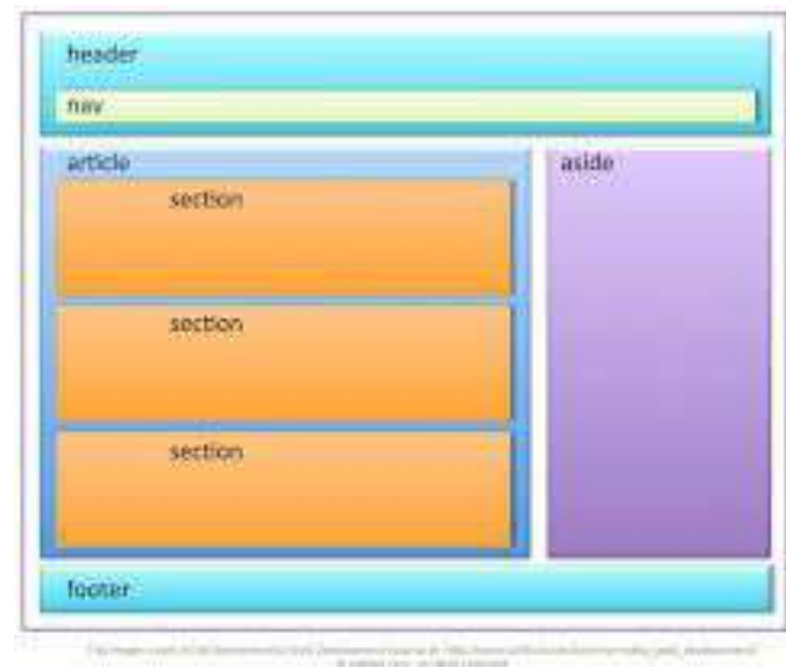
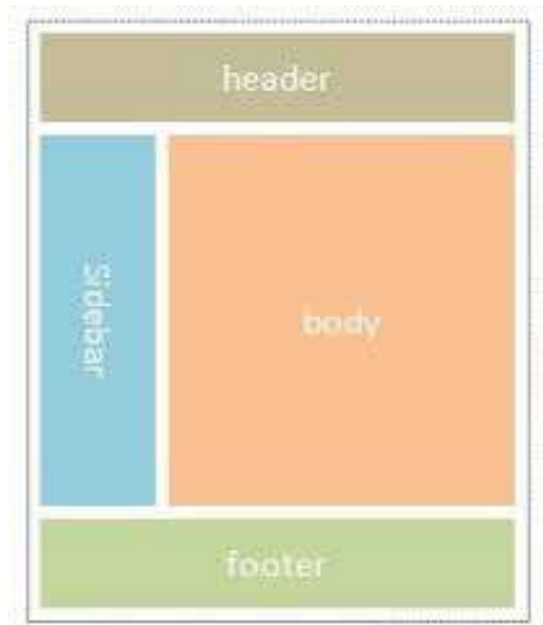


# Ventajas de una Aplicación web

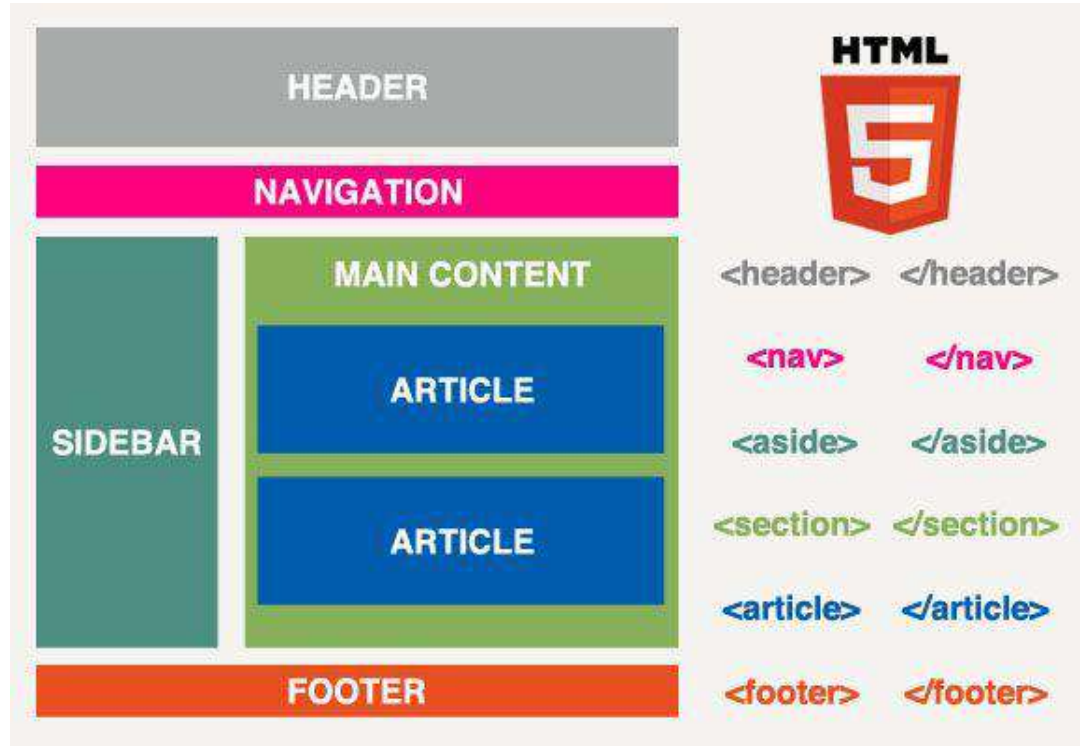
- Se puede ejecutar desde cualquier dispositivo que cuente con navegador y con conexión a internet. No es necesario instalarlas en el dispositivo, basta con un navegador web.
- Son fáciles de actualizar y mantener.
- Su funcionalidad es independiente al Sistema Operativo instalado en la PC del usuario.
- Las páginas web tienen estructuras similares que mejoran la experiencia del usuario al interactuar con una página.



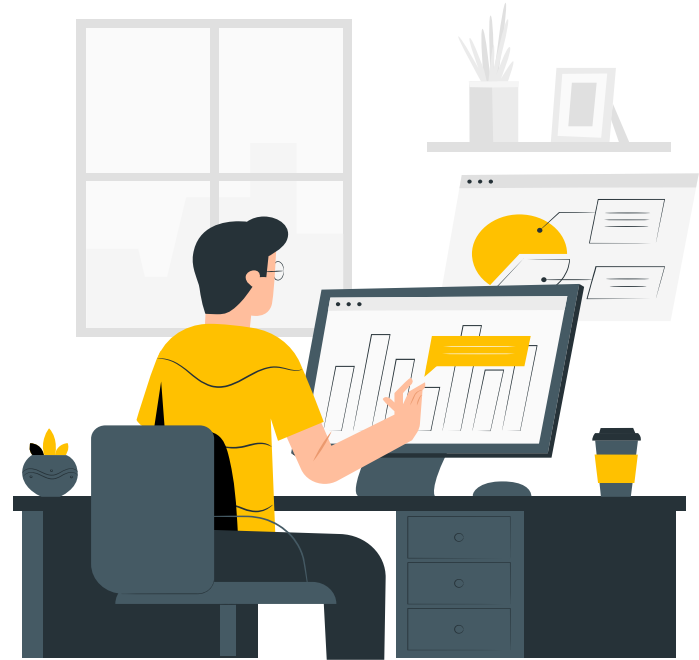
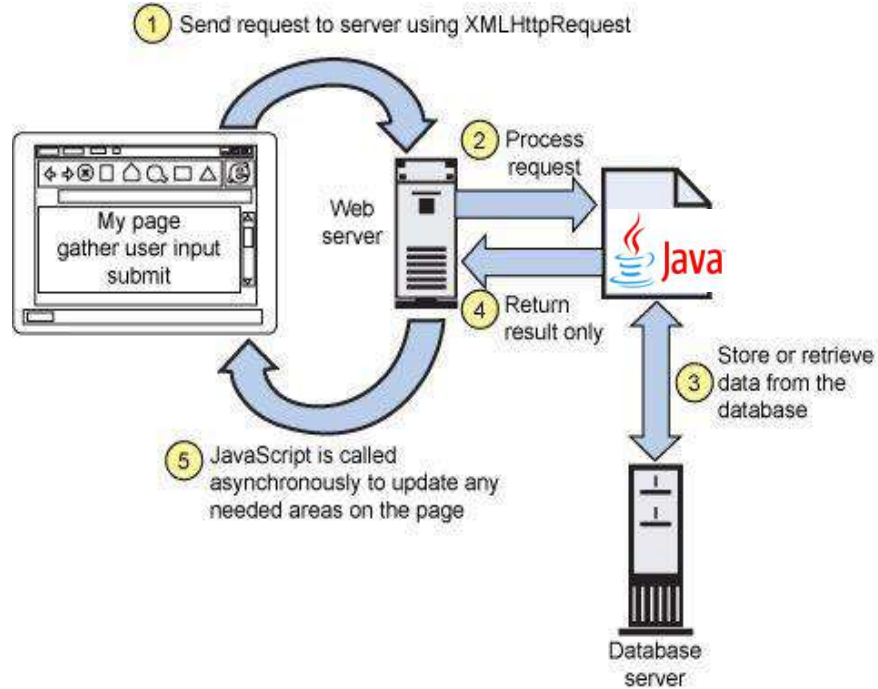
# Ejemplo estructura de un Página Web



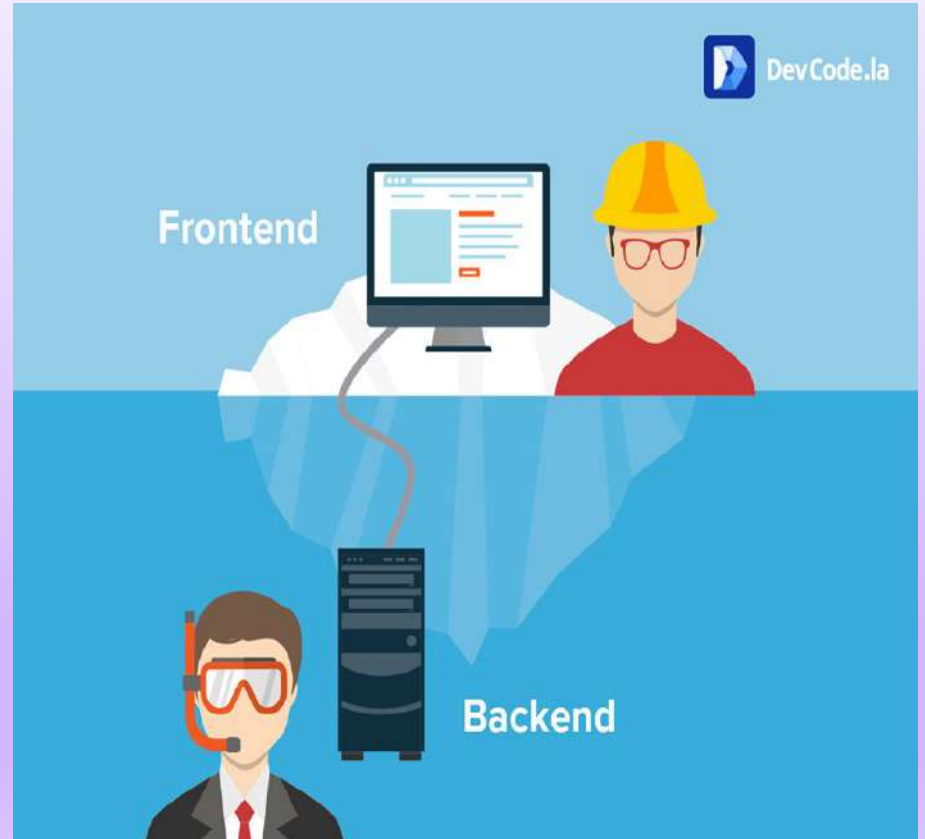
# Ejemplo estructura de un Página Web



# Cliente/Servidor

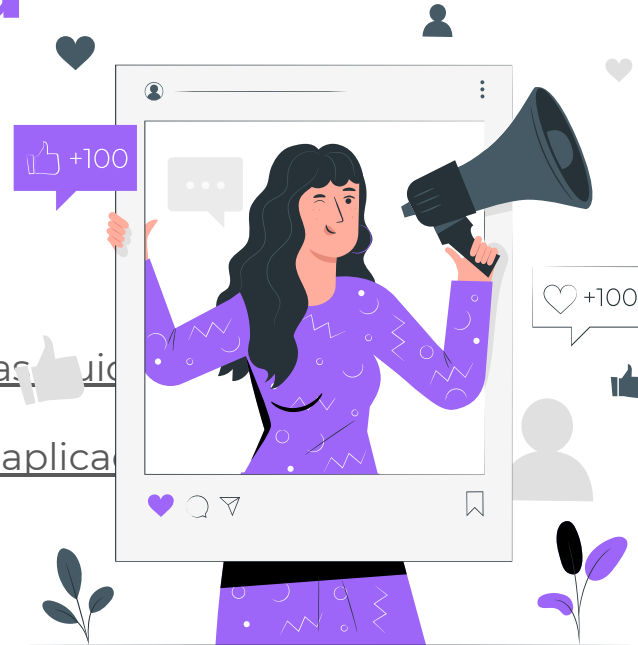


# BACKEND FRONTEND



# Ejemplos de Página Web

- Diferentes páginas web:
- <https://www.w3schools.com/>
- <https://instintobinario.com/category/hardware/>
- <https://www.knorr.com/ar/productos/sopas/sopas-quick.html>
- <https://blog.ida.cl/estrategia-digital/diferencias-aplicacion-web-sitio-web/>
- Lo que **no** hay que hacer:
- <http://arngren.net/>






# Cómo pensar la estructura sitio web (conceptual)



Tener **clara la idea de negocio y sus partes** (representadas en la web) conceptualmente. Se trata “simplemente” de hacer un ejercicio de análisis y reflexión.

## 1º. ¿Cuál es el objetivo que pretendo cumplir con este diseño web?

en ello está basado todo tu negocio. Si vas a montar una tienda de calzado en la calle, seguro que tienes clarísimo que tu objetivo es vender calzado. Sin embargo, la gente piensa que su objetivo es tener miles de visitas, o posicionarse en el número 1 de Google. Sí, eso está muy bien, pero PARA vender zapatos!





# Cómo pensar la estructura sitio web (conceptual)




2º. ¿Quiénes son tus clientes objetivo?

Y ¿quién eres tú para merecerlos?

Es **clave** saber **quién es tu audiencia** para dirigirte a ella. **Ahí radica la diferencia: recibir tráfico cualificado, interesado realmente en lo que ofreces.** De lo contrario, podrás conseguir miles de visitas de paso, pero muy pocos clientes interesados.

Es importante reflexionar sobre **qué tienes tú o tu producto o tu servicio que te haga diferente** del resto: ¿por qué habría yo de elegirte a ti y no al de al lado?

Identificar los **beneficios** que obtendrá la persona que te compre, contrate o lo que sea.








# Cómo pensar la estructura sitio web (conceptual)



3°. ¿Cuáles son, ahora que tienes claro lo anterior, tus palabras clave?


Las **palabras clave son el alimento del que se nutre** tu blog y tu web. **Tienes que tenerlas claras.** Algo que haga referencia, a tu cliente ideal, a tu especialidad o lo que te hace diferente, o a algún beneficio que posea tu producto. Esto último es fruto de identificar beneficios y nicho de mercado.





# Cómo pensar un Proyecto Web



1. Tipos de estructura de un sitio web
  2. Número de páginas en la estructura de un sitio web
  3. Niveles de estructura de un sitio web
  4. Cómo definir la estructura de un sitio web casi perfecto (conceptual)
  5. Pasos para estructurar el sitio (técnico)
    1. Paso 1: Hacer el árbol de la estructura de un sitio web
    2. Paso 2: Hacer estructura de un sitio web amigable para SEO
    3. Paso 3: Enlaces internos para mejorar la estructura de un sitio web
  6. Anexo: Yo he venido a hablar de mi libro, oiga
- 

## Internet - Acerca de...

Internet (el Internet o, también, la Internet) es un conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, lo cual garantiza que las redes físicas heterogéneas que la componen constituyan una red lógica única de alcance mundial. Sus orígenes se remontan a 1969, cuando se estableció la primera conexión de computadoras, conocida como ARPANET, entre tres universidades en California (Estados Unidos). Uno de los servicios que más éxito ha tenido en Internet ha sido la World Wide Web (WWW o la Web), hasta tal punto que es habitual la confusión entre ambos términos.

La WWW es un conjunto de protocolos que permite, de forma sencilla, la consulta remota de archivos de hipertexto. Esta fue un desarrollo posterior (1990) y utiliza Internet como medio de transmisión. Existen, por tanto, muchos otros servicios y protocolos en Internet, aparte de la Web: el envío de correo electrónico (SMTP), la transmisión de archivos (FTP y P2P), las conversaciones en línea (IRC), la mensajería instantánea y presencia, la transmisión de contenido y comunicación multimedia — telefonía (VoIP), televisión (IPTV)—, los boletines electrónicos (NNTP), el acceso remoto a otros dispositivos (SSH y Telnet) o los juegos en línea.

El uso de Internet creció rápidamente en el hemisferio occidental desde la mitad de la década de 1990 y desde la década de 2000 en el resto del mundo.

En los 20 años desde 1995, el uso de Internet se ha multiplicado por 100, cubriendo en 2015 a la tercera parte de la población mundial. La mayoría de las industrias de comunicación, incluyendo telefonía, radio, televisión, correo postal y periódicos tradicionales están siendo transformadas o redefinidas por Internet, y permitió el nacimiento de nuevos servicios como correo electrónico (e-mail), telefonía por Internet, televisión por Internet, música digital, y vídeo digital.

Las industrias de publicación de periódicos, libros y otros medios impresos se están adaptando a la tecnología de los sitios web, o están siendo reconvertidos en blogs, web feeds o agregadores de noticias. Internet también ha permitido o acelerado nuevas formas de interacción personal por medio de mensajería instantánea, foros de Internet, y redes sociales.

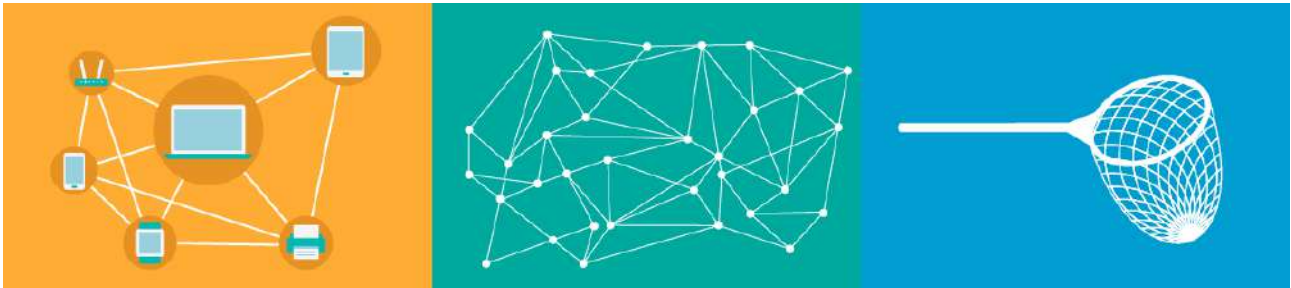
El comercio electrónico ha crecido exponencialmente tanto por grandes cadenas como pequeñas y medianas empresas o emprendedores, ya que ahora pueden vender por Internet productos o servicios hacia todo el mundo.

Los servicios empresariales y financieros en Internet afectan las cadenas de suministro en todas las industrias.

## ¿Cómo funciona Internet?

Muchos ven a Internet como una “nube” de tecnología o alguna suerte de repositorio de aplicaciones y datos a los que se puede acceder, trabajar o simplemente consultar. Sin embargo, el uso cotidiano hace que muy pocas veces nos preguntemos qué es realmente.

Internet se define como una gran “red de redes”, es decir, una red conectada a otra de manera continua y simultánea, pero para entenderlo mejor veamos primero qué es una red:



Cada uno de estos gráficos representa una especie de red: de computadoras, de puntos o de pesca. Lo que define a estas redes como tales es que existe “interconexión” entre sus componentes, aún cuando no se encuentren la totalidad de los puntos conectados entre sí. Desde cualquier punto de la red podremos llegar a otro, con más o menos vueltas, pero todos los destinos son alcanzables. Internet es justamente eso: una gran red donde todos los dispositivos están conectados entre sí.

## ¿Cómo se comunican las redes de Internet?

Para que una interconexión sea realmente global, todas las redes que se conectan a la gran red de redes deben hacerlo a través de un mismo protocolo o “lenguaje en común”. Es decir, el protocolo de comunicación de Internet debe ser una implementación estándar que garantice la conexión desde cualquier origen hasta cualquier destino.

Durante la década del '70, un grupo de investigadores de Estados Unidos creó un protocolo de comunicación que denominaron TCP/IP, y conectaron entre sí a un conjunto de redes de computadoras a pedido del gobierno de ese país, sin saber que el proyecto resultaría tan exitoso y tendría alcance global. De esta forma, TCP/IP se convirtió en el modelo de comunicación de las redes que componen Internet.

### Interconexión

Probablemente la computadora que utilizamos a diario en la oficina esté conectada a una red de computadoras. En nuestra casa, la computadora portátil, tablet y celulares los vinculamos a un dispositivo inalámbrico que luego se conecta a la red de fibra óptica de la empresa que nos brinda servicio de Internet y de televisión por cable o telefonía. Éste se comunica a la red de otro Proveedor de Servicios de Internet o ISP (por sus siglas en inglés) más grande, que llega a un número mayor de hogares y empresas, y así sucesivamente. A su vez, estos proveedores de acceso internacional se interconectan con

otros más grandes, denominados Carriers, a través de fibras ópticas transcontinentales y satélites, entre otros.

Con el correr de los años, esta gran red de redes ha pasado a ocupar un lugar relevante en múltiples niveles ya que es transversal a cuestiones sociales, políticas y económicas, y ya no excluyente de ámbitos tecnológicos. El impacto de Internet es tal que ha modificado paradigmas y continúa estableciendo nuevos desafíos en torno a la educación, la industria, la seguridad, los derechos humanos, las políticas públicas y las nuevas tecnologías, entre otros.

De lo anterior se hace evidente que Internet no es algo dado y que es producto de un proceso histórico e interdisciplinario que continúa en constante movimiento y en donde la participación de diversas partes interesadas ha sido determinante para su desarrollo. Es por esto que resulta fundamental comprender que la participación de diversos sectores, y más aún de nuevas generaciones, en la Gobernanza de Internet, es indispensable en las definiciones del futuro de Internet, contemplando intereses y derechos de todos los sectores de la comunidad.

## Protocolo TCP/IP

### ¿Qué es un protocolo TCP/IP y cómo funciona?

Como sucede con las personas, es importante que los equipos tengan un modo común de comunicarse entre ellos. Para la mayoría de los equipos actuales, este modo es TCP/IP. TCP/IP suele venir integrado en los equipos y está automatizado en buena medida, aunque puede ser útil comprender el modelo TCP/IP, en especial si va a configurar un equipo para conectarlo a otro sistema. Este artículo explica cómo funciona TCP/IP.

### ¿Qué significa TCP/IP?

TCP/IP son las siglas de Transmission Control Protocol/Internet Protocol (Protocolo de control de transmisión/Protocolo de Internet). TCP/IP es un conjunto de reglas estandarizadas que permiten a los equipos comunicarse en una red como Internet.

### ¿Qué hace exactamente TCP/IP? ¿Y cómo funciona?

Por sí mismo, un equipo puede realizar determinadas tareas. Pero su potencia se multiplica cuando es capaz de comunicarse con otros. Muchas de las cosas para las que utilizamos los equipos (enviar mensajes de correo electrónico, ver Netflix u obtener indicaciones para llegar a un sitio) dependen de la comunicación entre ellos. Pueden ser equipos de distintas marcas o incluso encontrarse en zonas del mundo diferentes. Y las personas y programas que los utilizan pueden hablar distintos lenguajes humanos e informáticos.

Una interacción determinada puede darse entre dos sistemas informáticos o involucrar cientos de sistemas. Sin embargo, como sucede al pasar una carta o un paquete de mano en mano, cada transacción se produce entre solo dos equipos cada vez.

Para que esto suceda, los dos equipos deben saber, por adelantado, cómo se espera que se comuniquen.

- ¿Cómo inician la conversación?
- ¿A quién le toca comunicarse?
- ¿Cómo sabe un equipo si su mensaje se ha transmitido correctamente?
- ¿Cómo terminan la conversación?

Los equipos lo resuelven mediante protocolos. Un protocolo es un conjunto de reglas convenido. En términos humanos, utilizamos protocolos sociales para saber cómo comportarnos y comunicarnos con otras personas. Las tecnologías tienen su propia forma de establecer reglas de comunicación, como el telégrafo cuando empleaba el código Morse o una radio CB en la que se utilizan códigos como "10-4".

Con los equipos sucede lo mismo, aunque las reglas son más estrictas. Cuando todos los equipos emplean el mismo protocolo, es posible transferir información. Cuando no es así, cunde el caos.

La comunicación era más complicada cuando la gente comenzaba a intercambiar información entre equipos. Cada fabricante tenía un sistema de comunicación propio entre sus máquinas, pero dichos sistemas no permitían la comunicación con los equipos de los

demás fabricantes. Pronto quedó claro que era necesario un estándar convenido que permitiera a los equipos de todos los fabricantes comunicarse entre ellos. Ese estándar es TCP/IP.

### **¿En qué se diferencian TCP e IP?**

TCP e IP son dos protocolos distintos para redes informáticas.

IP es la parte que obtiene la dirección a la que se envían los datos. TCP se encarga de la entrega de los datos una vez hallada dicha dirección IP.

Es posible separarlos, pero lo cierto es que no tiene mucho sentido diferenciar entre TCP e IP. Como se usan juntos tan habitualmente, “TCP/IP” y “modelo TCP/IP” son ya terminología reconocida.

*Mírelo de esta forma:* La dirección IP es como el número de teléfono que se asigna a su smartphone. TCP es toda la tecnología que hace que el teléfono emita un timbre al recibir una llamada y que le permite hablar con alguien al otro lado de la línea. Son cosas diferentes, pero tampoco tienen sentido la una sin la otra.

### **¿Qué hace exactamente TCP/IP? ¿Y cómo funciona?**

TCP/IP fue desarrollado por el Departamento de Defensa de EE. UU. para especificar el modo en que los equipos transfieren datos de un dispositivo a otro. TCP/IP incide mucho en la precisión y dispone de varios pasos para garantizar la correcta transmisión de los datos entre ambos equipos.

Este es uno de los mecanismos que emplea para ello. Si el sistema enviara un mensaje entero en una pieza y se encontrara cualquier problema, sería necesario enviar de nuevo el mensaje completo. Lo que hace TCP/IP es descomponer cada mensaje en paquetes que se vuelven a ensamblar en el otro extremo. De hecho, cada paquete podría tomar una ruta distinta hasta el equipo de destino si la ruta deja de estar disponible o está muy congestionada.

Además, TCP/IP divide las distintas tareas de comunicación en capas. Cada capa tiene una función distinta. Los datos pasan por cuatro capas independientes antes de recibirse en el otro extremo (como se explica en la sección siguiente). A continuación, TCP/IP recorre estas capas en orden inverso para reensamblar los datos y presentárselos al destinatario.

El propósito de las capas es crear un sistema estandarizado, sin que los distintos fabricantes de hardware y software tengan que gestionar la comunicación por su cuenta. Es como conducir un coche: todos los fabricantes convienen en la posición de los pedales, así que no tenemos que tener eso en cuenta al cambiar de coche. También significa que es posible actualizar determinadas capas, por ejemplo, para mejorar el rendimiento o la seguridad, sin tener que actualizar todo el sistema.

### **Las cuatro capas del modelo TCP/IP**

TCP/IP es un protocolo de enlace de datos que se utiliza en Internet. Su modelo se divide en cuatro capas diferenciadas. Cuando se emplean juntas, es posible referirse a ellas como un paquete de protocolos.

#### *Capa de enlace de datos*

La capa de enlace de datos (también denominada capa de enlace, capa de interfaz de red o capa física) es la que maneja las partes físicas del envío y recepción de datos mediante el cable Ethernet, la red inalámbrica, la tarjeta de interfaz de red, el controlador del dispositivo en el equipo, etcétera.

#### *Capa de Internet*

La capa de Internet (también denominada capa de red) controla el movimiento de los paquetes alrededor de la red.

#### *Capa de transporte*

La capa de transporte es la que proporciona una conexión de datos fiable entre dos dispositivos. Divide los datos en paquetes, hace acuse de recibo de los paquetes que recibe del otro dispositivo y se asegura de que el otro dispositivo haga acuse de recibo de los paquetes que recibe a su vez.

#### *Capa de aplicaciones*

La capa de aplicaciones es el grupo de aplicaciones que requiere comunicación de red. Es con lo que el usuario suele interactuar, como el correo electrónico y la mensajería. Como la capa inferior gestiona los detalles de la comunicación, las aplicaciones no tienen que preocuparse por ello.

### **¿Mis paquetes de datos se mantienen privados?**

No. Cuando los paquetes se transmiten entre equipos, son vulnerables y otros pueden verlos. Esa es una de las razones por las que se aconseja evitar las redes Wi-Fi públicas para enviar datos que deban mantenerse privados, así como utilizar cifrado.

Si esto es algo que le preocupa (por ejemplo, si va a enviar información de identificación personal o datos financieros), puede cifrar los datos empleando una red privada virtual (VPN) o trabajando con páginas web con seguridad https.

### **¿TCP/IP funciona con toda clase de direcciones IP?**

Hay varios tipos de direcciones IP. No obstante, todas ellas utilizan TCP/IP.

Las diferencias entre los tipos de direcciones IP son transparentes para el usuario esporádico, y el hecho de que no necesite saber mucho al respecto es una de las ventajas de TCP/IP. Normalmente, estos asuntos los administra quien haya configurado el sistema operativo del equipo o el dispositivo móvil. En cualquier caso, a modo de aclaración:

- Las direcciones IP estáticas no cambian en ningún momento. Son como la dirección fija de su domicilio, un dato inalterable.



- Las direcciones IP dinámicas cambian, o al menos están diseñadas para cambiar. Cuando un sistema informático utiliza una dirección IP dinámica, anuncia “¡aquí es donde puedes encontrarme!” a la red local.

Tal vez haya oído hablar de ciudades en las que la población crece tan rápido que se han tenido que crear nuevos códigos de área para que los recién llegados puedan tener número de teléfono. Con el número siempre creciente de dispositivos conectados, TCP/IP ha tenido un problema similar. Básicamente, Internet se estaba quedando sin direcciones IP. Por eso se desarrolló una nueva versión de dirección IP denominada IPv6, una alternativa a las direcciones IPv4 existentes.

De hecho, TCP/IP se incluye como estándar. TCP/IP es el paquete de protocolos más utilizado en la web. Millones de personas lo emplean cada día, aunque no sean conscientes de ello.

En la inmensa mayoría de los equipos, TCP/IP se integra como estándar. No tiene que hacer nada para configurarlo de manera manual. En ocasiones, tal vez tenga que decirle a una aplicación cuál es su dirección TCP/IP.

En algunas circunstancias puede querer ocultar su dirección IP, normalmente por motivos de seguridad.

### **¿Cómo puedo saber cuál es mi dirección TCP/IP?**

Cada dispositivo tiene su propia dirección TCP/IP. Por lo general, el dispositivo puede comunicarse automáticamente, pero a veces es necesario proporcionar su dirección TCP/IP de forma manual. El modo de encontrar su dirección IP depende del sistema operativo.

Podés consultar tu IP pública desde una página web como <https://www.cual-es-mi-ip.net/>

Conceptualmente, el número obtenido se puede asemejar a la “dirección” de una habitación de hotel, en la que la gerencia del mismo sabe que puede encontrarlo o el número de teléfono de tu celular.



# Clase 2

Introd. a HTML





# ¿Qué es HTML?

Es un lenguaje utilizado para definir la estructura y semántica de una página web.

HTML significa **lenguaje de marcado de hipertexto**, marcado se refiere a que debemos utilizar determinadas etiquetas para indicarle al navegador lo que queremos mostrar y hipertexto porque esas mismas etiquetas contienen metadatos.

Llamamos HTML semántico a un documento que usa correctamente las etiquetas para que la estructura resultante, quitando la capa de diseño, tenga sentido por sí sola.



# ¿Cómo funciona?

- 1 - El navegador (cliente) le pide información al servidor.
- 2 - El servidor devuelve la información al cliente en un archivo HTML.
- 3 - El navegador en el cliente lee el archivo de arriba hacia abajo y de izquierda a derecha para interpretar la información.
- 4 - Tiene en cuenta las etiquetas que tiene el documento y las va renderizando en pantalla (lo que se ve en el navegador).



# Estructura



```
<!DOCTYPE html>
```

```
<html lang= "es">
```

```
  <head>
```

```
    <meta charset="utf-8"/>
```

```
    <title>Título de la Página</title>
```

```
  </head>
```

```
    <body>
```

```
      <!-- Acá va tu código -->
```

```
    </body>
```

```
</html>
```



# Etiquetas



Una etiqueta describe la información contenida entre la etiqueta de apertura y la de cierre.

**<TipoDeEtiqueta>** *Contenido* **</CierreTipoDeEtiqueta>**

**<!DOCTYPE html>** indica que la versión corresponde a HTML5.

**<html lang= “es”>** es la etiqueta principal que engloba al resto de las etiquetas, el atributo lang define el tipo de lenguaje.

**<head>** es la cabeza del documento que contiene los metadatos de la página web.

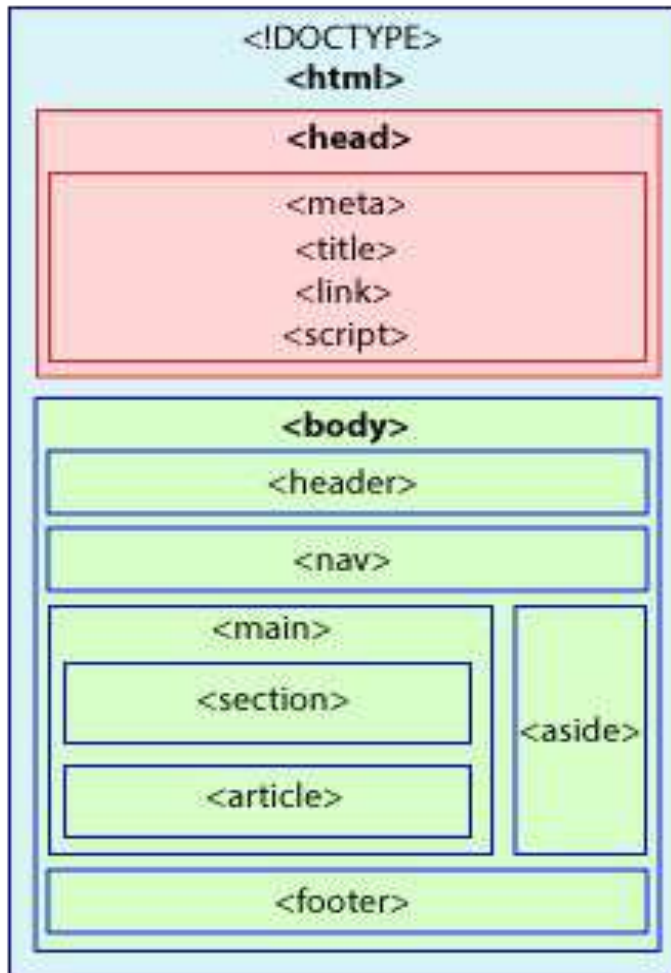
**<body>** es el cuerpo del documento donde va a estar todo el contenido que vamos a mostrar.

**<meta charset=“utf-8”/>** indica al navegador qué tipo de caracteres contiene la página, con el atributo charset vamos a indicar el conjunto de caracteres que vamos a usar y con el valor “utf-8” abarcamos a la mayoría de los sistemas escritura.



# Etiquetas

## Estructura página web



## Etiquetas semánticas

**<header>** encabezado de un documento o sección.

**<nav>** define un conjunto de enlaces de navegación.

**<main>** representa al contenido principal dentro del body.

**<section>** define secciones de un documento.

**<article>** especifica contenido independiente ej: un mensaje en un foro, comentarios, etc.

**<aside>** se suele usar para colocar información adicional ej: publicidad.

**<footer>** pie de página, suele contener información de contacto, mapa del sitio.



# Etiquetas



**<Etiqueta**      **atributo =**      **“valor”**      **>**      *Contenido*      **</Etiqueta>**



html. body,  
img, title,  
head ..



charset, alt,  
src, id, class,  
href, target



utf-8, src,  
nombre de la  
clase/id, link

## Ejemplo:

**<a href = “https://www.google.com” target = “\_blank”**      *Google*      **</a>**





# Etiquetas Básicas

**<h1>, <h2>,**

**<h3>....<h6>:** Encabezados, numerados del 1 al 6 por orden de relevancia. Es importante respetar ese orden para que el navegador entienda la estructura de la página.

**<p>:** Representa un párrafo.



# Etiquetas de Texto

**<b>:**           **Texto en negrita o bold**

**<mark>:**       **Texto marcado**

**<ins>:**       Texto insertado

**<small>:**      Texto más pequeño

**<i>:**           *Texto en Itálica o cursiva*

**<del>:**       ~~Texto tachado~~

**<sup>:**        Texto en superíndice

**<sub>:**        Texto del subíndice



## Más Etiquetas Básicas

**<!-- comentario -->** se utiliza para añadir comentarios dentro del código que el usuario no podrá ver.

**<a>** Define un hipervínculo, con el atributo href le indicamos el link

`<a href="https://google.com">ir a google</a>`

**<img />** Define una imagen y con el atributo src le indicamos al navegador dónde buscarla.

``



# SEO

- ❑ El posicionamiento en buscadores, optimización en motores de búsqueda o SEO (del inglés search engine optimization), es un conjunto de acciones orientadas a mejorar el posicionamiento de un sitio web en la lista de resultados de Google, Bing, u otros buscadores de internet.
- ❑ El SEO trabaja aspectos técnicos como la optimización de la estructura y los metadatos de una web, pero también se aplica a nivel de contenidos, con el objetivo de volverlos más útiles y relevantes para los usuarios.



# Actividad en clase

Investigar los siguientes sitios web. Identificar header, footer, nav, headings, párrafos, imágenes, link, iconos, etc.:

- <https://www.knorr.com>
- <https://www.eladerezo.com>

## **Ingresando al mundo HTML**

Esta sección HTML te enseña a trabajar con el lenguaje de marcación que sirve para construir las páginas web. HTML es el primer paso que debería completar cualquier persona que quiera dedicarse al desarrollo de web en general. Es además un conocimiento recomendado para cualquier persona que trabaje en el medio Internet.

Este es un manual con bastante detalle, que empieza en el conocimiento más básico y recorre cada uno de los elementos que se pueden usar para construir todo tipo de contenido en una web, repasando las etiquetas disponibles en el lenguaje. Las distintas secciones del manual de HTML te llevarán desde la creación de contenido básico como párrafos o listas, hasta el trabajo con formularios, tablas y otros elementos más complejos.

Entonces, esta sección HTML tiene como objetivo que aprendas a trabajar con todo el conjunto de etiquetas que existen en este lenguaje, pero además a entender cuáles son los mecanismos habituales para construir páginas web, adaptadas a las costumbres más actuales.

### **¿Qué podés esperar de esta sección?**

A lo largo de la sección vamos a descubrir el principal lenguaje utilizado para la creación de páginas web: **HyperText Markup Language**, más conocido mediante sus siglas HTML.

Puede que en un principio, el hecho de hablar de un lenguaje informático te ponga un poco nervioso, sin embargo, el HTML no deja de ser más que una forma sencilla de especificar el contenido de las páginas, indicando el texto y otros elementos como imágenes, tablas, listas, etc. Al final es de suma importancia el lenguaje porque es el medio con el cual se suministra el contenido a los navegadores y por tanto, si queremos comenzar a aprender a crear páginas web, necesariamente debemos comenzar por aquí.

Esta sección será práctica y teórica. Iremos conociendo el lenguaje a través de numerosos ejemplos que te sugerimos realizar por ti mismo para asimilar mejor los conocimientos.

Verás también como HTML dentro del contexto de la creación de una página web, se apoya en sobre editores, programas para subir archivos al servidor, etc.

Es necesario leer este texto para cualquier persona, que con conocimientos de informática, desean hacer proyectos front-end en cualquier lenguaje.

Con ganas, y realizando los ejercicios, al final tendrás las habilidades y conocimientos como para realizar una página web por tu cuenta lista para publicar. Además podrás:

- Identificar qué se debe hacer con HTML y qué no.
- Capacidad para crear y publicar tu propio sitio web con un mínimo de calidad.
- Conocimientos de todo tipo sobre las tecnologías y herramientas empleadas en el ámbito de la red.

**Las primeras cosas que debes saber sobre HTML: historia, objetivos y demás conocimientos...**

HTML es el lenguaje con el que se escribe el contenido de las páginas web.

Las páginas web pueden ser vistas por el usuario mediante un tipo de aplicación llamada cliente web o más comúnmente "navegador". Podemos decir por lo tanto que el HTML es el lenguaje usado para especificar el contenido que los navegadores deben representar a la hora de mostrar una página web.

Este lenguaje nos permite unir textos, imágenes, enlaces... y combinarlos a nuestro gusto. La ventaja del HTML a la hora de representar el contenido en un navegador, con respecto a otros formatos físicos como libros o revistas, es justamente la posibilidad de colocar referencias a otras páginas, por medio de los enlaces hipertexto.

### **Un poco de historia.**

El lenguaje HTML se creó en 1991. Tiene una historia realmente corta pero para su poca vida ha sufrido importantes cambios. Su padre es Tim Berners-Lee que lo diseñó con objetivos divulgativos. Inicialmente no se pensó que la web llegaría a ser un área de ocio con carácter multimedia, de modo que, nació sin dar respuesta a todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro. Sin embargo, pese a esta deficiente planificación, si que se han ido incorporando modificaciones con el tiempo, agregando nuevas características para cubrir las nuevas necesidades.

Como hemos dicho, los programas que leen y presentan las páginas web a los usuarios se llaman navegadores. Éstos son los responsables de interpretar el HTML y "pintar" una página tal como ellos entiendan que deben hacer. Sin embargo, esas reglas de representación no son subjetivas de cada fabricante del navegador, sino que existe una organización llamada W3C que se encarga de definir el estándar que todos deben seguir a la hora de escribir e interpretar el HTML. Estos estándares del HTML se conocen como "Especificaciones", las cuales han ido apareciendo en el tiempo. El HTML5 es el último estándar en la actualidad.

Históricamente los navegadores, además de la propia comunidad de usuarios, han sido los mayores impulsores de los cambios ocurridos en el lenguaje. Una vez detectada la necesidad es W3C el que crea el estándar y marca una dirección que todos deberían seguir.

### **Los navegadores y sus problemas**

El conflicto generado por los navegadores es debido a su diversidad. Existen multitud de navegadores o clientes web presentes en el mercado los cuales muchas veces no son capaces de interpretar un mismo código de una manera unificada. Esto obliga al desarrollador a, una vez creada su página, comprobar que esta puede ser leída satisfactoriamente por todos los navegadores, o al menos, los más utilizados. Cuando surgen problemas de interpretación, queda de parte del desarrollador resolver el problema tirando de técnicas o conocimientos que él disponga.

Afortunadamente, en la actualidad las diferencias de interpretación de los navegadores con respecto a un mismo código HTML son mínimas, pero en el pasado los desarrolladores tenían que emplear mucho tiempo en remar contracorriente para solucionarlos. Sin embargo, todavía quedan muchos usuarios que navegan con sistemas

anticuados, ya sea por falta de interés para actualizar su navegador, conocimientos, o por disponer ordenadores muy antiguos.

Pero no todo ha sido malo por parte de los navegadores. Ellos también son los responsables de introducir nuevas etiquetas en el uso común del día a día, que se han ido incorporando al estándar HTML en sucesivas versiones. Aunque antes de estandarizarse esas etiquetas era común que cada navegador crease su etiqueta propietaria para resolver la misma necesidad, lo que obligaba a los desarrolladores a repetir código o incluso a hacer versiones de páginas diferentes para navegadores. Con todo esto no queremos asustar a nadie y volvemos a repetir que las diferencias en la actualidad son mínimas, pero sí deseamos que quede clara la necesidad de la estandarización creada por el W3C, responsable de marcar una pauta que actualmente cumplen todos los navegadores modernos de manera bastante fiel.

## **Los lenguajes de la web**

HTML no está solo como único lenguaje para crear la web, aunque en un principio sí que era así. Su evolución tan anárquica ha supuesto toda una serie de inconvenientes y deficiencias que han debido ser superados con la introducción de otras tecnologías accesorias capaces de organizar, optimizar y automatizar el funcionamiento de la web. Ejemplos que pueden sonarnos son CSS o JavaScript. Veremos más adelante en qué consisten estas tecnologías.

Lo que es importante para el desarrollador es conocer el enfoque de cada lenguaje, para saber cuál es la manera correcta de utilizarlo y cómo se complementan los unos a los otros. No es necesario que se sea experto en todos ellos, pero sí saber qué cosas se deberían hacer con cada uno, para no cometer errores que deriven en una mala interpretación por parte de los navegadores. Así mismo tenemos que pensar que no todas las personas van a acceder a una web a través de un ordenador, sino también de un teléfono o de navegadores especializados en donde la accesibilidad debe ser diferente. Es por ello que es importante escribir correctamente los lenguajes, respetando los estándares y así cada navegador podrá hacer su mejor papel para representar la página lo más correctamente posible.

Deseamos incidir mucho en este detalle, la correcta utilización del HTML: escribir el contenido, para que nuestro trabajo sea lo más adecuado y de elevada calidad.

## **Los editores de HTML**

Además del navegador necesario para ver los resultados de nuestro trabajo, necesitamos evidentemente otra herramienta capaz de crear la página en sí.

Un archivo HTML (una página) no es más que un texto plano (sin forma estética) al que le colocamos extensión ".html". Es por ello que para programar en HTML necesitamos un editor de texto.

Es recomendable usar un editor de textos sencillo, de texto plano. Queremos remarcar que nunca se debe usar el tipo de editor de textos que se usan para escribir documentos, cartas, trabajos para el colegio, como Wordpad o Microsoft Word, pues colocan su propio código especial al guardar los documentos y HTML es únicamente texto plano, con lo que podremos tener problemas.



Aunque para aprender es recomendable que se use un programa que te permita escribir el código plano, podés comenzar tus primeros pasos con algún editor que genera el código mientras escribís en un procesador de texto al estilo Word. Te recomendaría el uso de Notepad++ o Kompozer, un sistema gratuito y que puede ayudarte a iniciar en el código HTML, posteriormente, creo que se puede continuar con VSCode o Geany, como para iniciar las etapas de creación de páginas web profesionales.

El tipo de editores que recomendamos son aquellos específicos para la edición de código, los cuales están pensados para facilitar los procesos de la programación y de la escritura de código plano como el del lenguaje HTML. Existen infinidad de editores de código interesantes, que nos aportan más o menos facilidades y que nos permiten aumentar nuestra productividad. No obstante, es aconsejable en un principio utilizar una herramienta lo más sencilla posible para poder prestar la máxima atención a nuestro código y familiarizarnos lo antes posible con él. Siempre tendremos tiempo más adelante de pasarnos a editores más versátiles con la consiguiente ganancia de tiempo.

No es posible decir a nadie el editor que debe de usar, porque cada uno tendrá sus preferencias. Igualmente para quienes están comenzando nosotros recomendamos:

- Notepad++
- VSCode

Igualmente una persona que comience puede pensar que tienen muchas opciones, pero realmente podemos comenzar con lo básico, crear nuevos archivos, editar el código, guardarlos en nuestro disco duro y punto. Recomendamos estos editores porque están disponibles para todas las plataformas (Windows, Mac y Linux) y porque son gratuitos para cualquier uso.

Aunque comenzaremos a usar HTML para definir cómo se puede ver un contenido, por ejemplo: si el texto debe tener color rojo, o el tamaño de la letra, o si se debe alinear a la derecha, es de destacar que actualmente para especificar el aspecto que debe tener una web se usa un lenguaje complementario, llamado CSS.

En resumen, HTML sirve para decir qué contenido debe tener una página y CSS sirve para decir cómo se debe representar tal contenido, con qué estilo, color, imágenes etc. Es fácil saltarse esta regla, porque en HTML existen diversas etiquetas (y atributos, de los que ya hablaremos) que realmente están pensados para definir la presentación. Es una herencia de versiones pasadas del HTML y aunque comenzaremos de esa forma debemos recordar que usar HTML para definir cómo debe de representarse un elemento en la página no es técnicamente correcto.

## **En pocas palabras**

Como has visto, una página es un archivo donde está contenido el código HTML en forma de texto. Estos archivos tienen extensión .html o .htm (es indiferente cuál utilizar). De modo que cuando programemos en HTML lo haremos con un editor de textos y guardaremos nuestros trabajos con extensión .html, por ejemplo mipágina.html

*Consejo: utiliza siempre la misma extensión en tus archivos HTML. Eso evitará que te confundas al escribir los nombres de tus archivos unas veces con .htm y otras con .html. Cabe remarcar que hoy todo el mundo usa la extensión ".html" y no ".htm".*

## Iniciando con la sintaxis de HTML

Describiremos a continuación la sintaxis con la que se trabaja en el lenguaje HTML, así como la estructura que tendrá el documento básico HTML.

El HTML es un "lenguaje de marcado". Basa su sintaxis en un elemento base al que llamamos marca, tag o simplemente etiqueta. A través de las etiquetas vamos definiendo los elementos del documento, como enlaces, párrafos, imágenes, etc. Así pues, un documento HTML estará constituido por texto y un conjunto de etiquetas para definir la función que juega cada contenido dentro de la página. Todo eso servirá al navegador para saber cómo se tendrá que presentar el texto y otros elementos en la página.

Existen etiquetas para crear negritas, párrafos, imágenes, tablas, listas, enlaces, etc. Así pues, aprender HTML es básicamente aprenderse una serie de etiquetas, sus funciones, sus usos y saber un poco sobre cómo debe de construirse un documento básico. Es una tarea muy sencilla de afrontar, al alcance de cualquier persona, puesto que el lenguaje es muy entendible para los seres humanos.

## Anatomía de una etiqueta HTML

La etiqueta presenta frecuentemente dos partes, su apertura y cierre, y se encierran ambas partes entre símbolos "menor que" y "mayor que". Lo veremos a continuación.

### *Apertura*

El inicio de una etiqueta se produce de la siguiente manera:

`<etiqueta>`

### *Cierre*

El final de una etiqueta se produce de manera similar a su apertura, aunque agregando una barra:

`</etiqueta>`

Si en nuestro documento HTML escribimos una frase con el siguiente código:

`<b>Esto está en negrita</b>`

Veremos que las palabras "Esto está en negrita" aparecen en **negrita**. Ya que el texto que encerramos entre una `<b>` que abre y otra que cierra `</b>` genera que la página muestre ese texto en negrita:

`<b> acá ponés cualquier texto </b>`

Es así de simple.

Otro ejemplo rápido. La etiqueta `p` define un párrafo. Si en nuestro documento HTML escribimos:

`<p>Hola, estamos en el párrafo 1</p>`

`<p>Ahora hemos cambiado de párrafo</p>`

Como resultado obtendremos dos párrafos con esos textos. En HTML los párrafos están separados por un doble salto de línea.

## Partes de un documento HTML

Además de todo esto, un documento HTML ha de estar delimitado por la etiqueta HTML.

Dentro de este documento, podemos asimismo distinguir dos partes principales:

La cabecera, delimitada por la etiqueta HEAD, donde colocaremos etiquetas de índole informativo, como por ejemplo el título de nuestra página. El contenido de la cabecera no suele aparecer en el cuerpo de la página, pero sirve a los navegadores y otros sistemas para encontrar información útil para entender y procesar el documento.

El cuerpo, flanqueado por la etiqueta BODY, que será donde colocaremos nuestro texto e imágenes delimitados a su vez por otras etiquetas como las que hemos visto.

El resultado de un documento básico tiene la siguiente estructura:

```
<html>
<head>
<title>Mi documento básico</title>
</head>
<body>

<p>Este es el cuerpo de mi primera página HTML</p>
<p>Este segundo párrafo también forma parte del cuerpo</p>

</body>
</html>
```

*Nota: a este documento básico le faltan todavía algunas cosas importantes que no queremos que se te olviden. Sin embargo hablaremos de ellas más adelante.*

## Las mayúsculas o minúsculas son indiferentes al escribir etiquetas

En HTML las mayúsculas y minúsculas son indiferentes. Quiere decir que las etiquetas pueden ser escritas con cualquier tipo de combinación de mayúsculas y minúsculas. Resulta sin embargo aconsejable acostumbrarse a escribirlas en minúscula ya que otras tecnologías que pueden convivir con nuestro HTML (XML por ejemplo) no son tan permisivas y nunca viene mal hacernos a las buenas costumbres desde el principio, para evitar fallos triviales en un futuro.

## Salto de línea en HTML

Otra de las cosas importantes de conocer sobre la sintaxis básica del HTML es que los saltos de línea no importan a la hora de interpretar una página. Un salto de línea será simplemente interpretado como un separador de palabras, un espacio en blanco. Es por ello que para separar líneas necesitamos usar la etiqueta de párrafo comentada antes, o la etiqueta BR que significa un salto de línea simple.

Esto es una línea.

<br>

Esto es otra línea<br>Esto es otra línea

Ahora, aunque estoy escribiendo aparentemente en otra línea, no se verá el salto de línea porque el BR no muestra algo en pantalla, sino que directamente muestra el texto en el siguiente renglón. Del código anterior obtendremos una página que muestra:

Esto es una línea.

Esto es otra línea.

Esto es otra línea

*Nota: la etiqueta BR no tiene su correspondiente cierre. Es un detalle que quizás te haya llamado la atención. Volveremos sobre ello más adelante.*

Es un detalle que choca al principio de usar HTML, pero al que te acabas acostumbrando con rapidez. Iremos viendo ejemplos a lo largo del curso, por lo que no debes preocuparte por ahora, sólo seguir leyendo.

## Tu primera página HTML

Vamos a ver cómo se hace una página muy sencilla en HTML, que sirva de práctica a quienes recién comienzan con esta forma de codificar.

Solo hemos comenzado a ver solo un poco de HTML, pero ya podemos hacer un primer ejemplo para darnos una idea de que se trata. Con lo visto hasta ahora ya casi estamos listos para practicar, aunque todavía tenemos que adquirir algún conocimiento adicional.

No te olvides ahora de practicar, así que usa tu editor de código preferido. Por ejemplo, se puede comenzar con el editor online de <http://html5-editor.net/> que es una gran ayuda para crear por tu cuenta los ejercicios que vamos a ir realizando.

Recuerda que la práctica es la mejor vía para afianzar los conocimientos.

Anteriormente ya adelantamos la forma de un documento HTML básico, con sus etiquetas de cabecera y cuerpo. No obstante, si deseamos seguir las pautas técnicas más estrictas, aún tenemos que agregar alguna cosa adicional para que todo funcione de la mejor manera.

### Doctype

Reconozco que el "doctype" no es la etiqueta más intuitiva, pero debemos mencionarla ahora porque es el inicio de cualquier archivo HTML. Viene heredada del XML, que es un lenguaje precursor del HTML. En el pasado la etiqueta Doctype era bastante más compleja, pero afortunadamente con la llegada de HTML5 se simplificó para quedar simplemente como esto:

```
<!DOCTYPE html>
```

El Doctype o "Declaración del tipo de documento" es una instrucción especial que va al inicio de nuestro documento HTML y que permite al navegador entender qué versión de HTML estamos utilizando. Esta información determinará la manera en la que el navegador procesa el documento, un DOCTYPE distinto podría implicar hasta una visualización diferente del sitio web dentro del mismo navegador.

### Juego de caracteres

El juego de caracteres es otro asunto que puede parecer un poco complejo, pero que tenerlo claro desde el principio te ayudará a no pasar en el futuro por diversos problemas. Este juego de caracteres, o codificación, depende del sistema operativo que estás usando para crear tu archivo HTML. Mientras que unos sistemas como Linux o Mac usan por defecto un juego de caracteres llamado UTF-8, en Windows se usa de manera predeterminada otro juego de caracteres llamado ISO-8859-1. Parece una información un tanto técnica y fuera de necesidad para introducir ahora que estamos comenzando, pero insistimos que nos ahorrará frustraciones al dar los primeros pasos, pero sobre todo en un futuro.

En HTML5 el juego de caracteres a usar es siempre UTF-8. Por lo que tendremos que tener especial atención si somos usuarios de Windows, para asegurarnos que usamos la codificación correcta. Por ejemplo, Brackets o Atom (editores de código), trabajan siempre en UTF-8, independientemente del sistema operativo. Si no estás usando uno de esos

editores, te recomendamos que los investigues y si quieres trabajar con tu propio editor infórmate sobre el juego de caracteres que produce y si existe alguna opción o configuración que te asegure usar siempre UTF-8.

Para definir qué juego de caracteres estamos usando en un documento HTML se tiene que escribir una etiqueta en la cabecera de la página, en el HEAD, llamada META. Realmente las etiquetas META las trataremos más adelante, porque sirven para varias cosas interesantes. Pero de momento nos aseguraremos que tenemos esta etiqueta en el head.

```
<meta charset="UTF-8">
```

Igualmente, no pretendemos hablar mucho de esta etiqueta por el momento, solo que te la tomes como un contrato a cumplir para tener un documento correcto.

A continuación tienes un documento básico con las etiquetas necesarias para comenzar con un código completo.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Mi primera página</title>
</head>
<body>
<p><b>Bienvenido,</b></p>
<p>Estás en la página <b>Comida para Todos los gustos</b>.</p>
<p>Aquí aprenderás recetas fáciles y deliciosas.</p>
</body>
</html>
```

Puedes copiar y pegarlo en tu editor de código. Ahora guarda ese archivo con extensión .html en tu compu. Para ello accedemos al menú Archivo y seleccionamos la opción Guardar como. En la ventana elegimos el directorio donde deseamos guardarlo y colocaremos su nombre, por ejemplo ejercicio1.html

*Consejo: Utiliza nombres en tus archivos que tengan algunas normas básicas para ahorrarte disgustos y líos.*

Nuestro consejo es que no utilices acentos ni espacios ni otros caracteres raros. También te ayudará escribir siempre las letras en minúsculas. Esto no quiere decir que debes hacer nombres de archivos cortos, es mejor hacerlos descriptivos para que te aclaren lo que hay dentro. Algún carácter como el guión medio "-" te puede ayudar a separar las palabras. Por ejemplo quienes-somos.html

Con el documento HTML creado, podemos ver el resultado obtenido a partir de un navegador.

Es conveniente, llegado a este punto, hacer hincapié en el hecho de que no todos los navegadores son idénticos a la hora de interpretar un documento. Desgraciadamente, los resultados de nuestro código pueden cambiar de uno a otro por lo que resulta aconsejable visualizar la página en varios clientes web. Generalmente se usan Chrome, Internet Explorer, Microsoft Edge y Firefox como referencias ya que son los más extendidos. Si usas Geany podrás probar el código haciendo click en el botón de "ejecutar" desde dentro de la aplicación. Lo que podrás ver en el video de explicación del ejercicio.

Actualmente Google Chrome acapara la mayoría de usuarios y Firefox e Internet Explorer/Edge están relegados a un segundo plano. Esto no quiere decir que lo debemos dejar totalmente de lado ya que incluso una minoría que puede proporcionarnos puede resultar muy importante para nosotros.

### **¿Cómo ver la página que hicimos?**

Una vez guardado el fichero con extensión .html, para abrir la página en el navegador, simplemente tienes que acceder a la carpeta donde has guardado el archivo y darle un doble clic. Se trata de una tarea sencilla que estamos seguros que podrás realizar. Si no lo consigues, fíjate que la mayoría de los navegadores tienen un menú. En el menú de "Archivo" de tu navegador preferido encontrarás una opción como "Abrir archivo", desde donde también podrás abrir una página realizada por ti con tu editor de código.

Una vez abierto el archivo podés ver tu primera página web. Algo sencillita pero por algo se empieza. Seguramente en poco tiempo seremos capaces de mejorar sensiblemente.

En la parte superior izquierda de la ventana del navegador puedes comprobar la presencia del texto delimitado por la etiqueta TITLE. Esta es una de las funciones de esta etiqueta, cuyo principal cometido es el de servir de referencia en los motores de búsqueda como Google.

Por otro lado, los elementos que colocamos entre la etiqueta BODY, y su cierre, se pueden ver en el espacio reservado para el cuerpo de la página.

Si hacés click con el botón derecho sobre la página y elegís "Ver código fuente de la página" o "Inspeccionar" en una ventana accesoria aparece el código de nuestro archivo HTML. Este recurso es de extrema importancia, ya que nos permite ver el tipo de técnicas empleadas por otros para la confección de sus páginas.

Una vez que hicimos esta práctica podemos ya profundizar en la descripción de algunas de las etiquetas más empleadas del HTML.

## Formato de párrafos en HTML

Veremos cómo utilizar párrafos y saltos de línea en páginas web. Incluiré también los encabezados como párrafos que sirven de título.

Anteriormente hemos presentado a título de ejemplo algunas etiquetas que permiten dar formato a nuestro texto.

Dar formato a un texto pasa por tareas tan evidentes como definir los párrafos, justificarlos, introducir viñetas, numeraciones o bien poner en negrita, itálica, etc.

Hemos visto que para definir los párrafos nos servimos de la etiqueta P que introduce un salto y deja una línea en blanco antes de continuar con el resto del documento.

Podemos también usar la etiqueta BR, de la cual no existe su cierre correspondiente (/BR), para realizar un simple salto de línea con lo que no dejamos una renglón en blanco sino que solo nos posicionamos en el renglón siguiente.

***Nota:*** existen otras etiquetas que no tienen su correspondiente cierre, como IMG para las imágenes, que veremos más adelante. Esto ocurre porque un salto de línea o una imagen no empiezan y acaban más adelante sino que sólo se colocan y ya tienen su propio funcionamiento.

Los párrafos delimitados por etiquetas P pueden ser fácilmente alineados a la izquierda, centro, derecha o justificados especificando dicha alineación en el interior de la etiqueta por medio de un atributo "align".

Un atributo, es entonces, un parámetro incluído en el interior de la etiqueta que ayuda a definir el funcionamiento de la etiqueta de una forma más personalizada. Veremos una gran cantidad de atributos muy útiles para todo tipo de etiquetas.

***Recordatorio:*** HTML se usa para definir el contenido. Por tanto, los atributos align actualmente ya no se usan en HTML, porque están definiendo la estética con la que un párrafo debe de representarse y no el contenido. Actualmente toda la definición estética se realiza bajo el código CSS, que sirve para definir el estilo, la forma.

Así, si deseamos introducir un texto alineado a la izquierda escribiríamos:

```
<p align="left">Texto alineado a la izquierda</p>
```

Para una justificación al centro:

```
<p align="center">Texto alineado al centro</p>
```

Para alinear a la derecha:

```
<p align="right">Texto alineado a la derecha</p>
```

En cada caso mencionado, el atributo align toma determinados valores que son escritos entre comillas. En algunas ocasiones necesitamos especificar algunos atributos para el



correcto funcionamiento de la etiqueta. En otros casos, el propio navegador toma un valor definido por defecto. Para el caso de align, el valor por defecto es left.

El atributo align no es exclusivo de la etiqueta P. Otras etiquetas muy comunes, que veremos más adelante, entre las cuales se introducen texto o imágenes, suelen hacer uso de este atributo de una forma habitual.

Imaginemos un texto relativamente largo donde todos los párrafos están alineados a la izquierda (por ejemplo). Una forma de simplificar nuestro código y de evitar introducir continuamente el atributo align sobre cada una de nuestras etiquetas es utilizando la etiqueta DIV.

Esta etiqueta, DIV, por sí sola no sirve para nada, salvo producir un salto de línea simple. Para que realmente se vea tiene que estar acompañada de algún estilo definido en el CSS o de atributos del HTML como align y lo que nos permite es alinear cualquier elemento (párrafo o imagen) de la manera que nosotros deseemos.

Así, el código:

```
<p align="left">Parrafo1</p>
<p align="left"> Parrafo3</p>
<p align="left"> Parrafo2</p>
```

es equivalente a:

```
<div align="left">
<p>Parrafo1</p>
<p>Parrafo2</p>
<p>Parrafo3</p>
</div>
```

*Nota: no es correcto utilizar align en una etiqueta DIV, por el mismo motivo que no es correcto utilizarlo en un párrafo.*

Como hemos visto, la etiqueta DIV marca divisiones en las que definimos un bloque de contenido, y a los que podríamos aplicar estilo de manera global, aunque lo correcto sería aplicar ese estilo del lado del CSS.

Ejemplo práctico:

Para practicar un poco lo que acabamos de ver vamos a proponer un ejercicio que podés resolver en tu PC. Simplemente queremos construir una página que tenga:

- 1 salto de línea triple
- 1 párrafo alineado a la izquierda
- 2 Párrafos centrados
- 3 Párrafos alineados a la derecha

El código fuente del ejercicio, con lo que sabemos hasta ahora, podría tener la siguiente forma:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>parrafos</title>
</head>
<body>

  <p align="center">
Ejemplo de formatear parrafos
</p>
  <p align="center">
Esto es el resultado:
</p>

  <div align="right">
<p>
Qué son los buscadores que no tienen porque mantener un índice y que tienen robots
que constantemente recorren Internet en busca de nuevas páginas para incluirlas en el
buscador.
</p>
<p>
Imaginemos un texto relativamente largo donde todos los párrafos están alineados a la
izquierda (por ejemplo).
</p>
<p>
Qué son los buscadores que no tienen porque mantener un índice y que tienen robots
que constantemente recorren Internet en busca de nuevas páginas para incluirlas en el
buscador. Estos buscadores suelen tener un formulario accesible desde la página inicial,
con el enlace correspondiente. No hay que navegar las categorías para acceder al
formulario.
</p>
</div>
<br>
<br>
<br>
<p>
Aquí finaliza la web.
</p>

</body>
</html>
```

## Encabezados

Existen otras etiquetas para definir párrafos especiales, que harán las veces de títulos. Son los encabezados o headings en inglés. Como decimos, son etiquetas que formatean

el texto como un titular, pero el hecho de que cambien el formato no es lo que nos tiene que preocupar, sino el significado en sí de la etiqueta. Es cierto que los navegadores asignan un tamaño mayor de letra y colocan el texto en negrita, pero lo importante es que sirven para definir la estructura del contenido de un documento HTML. Entre otras cosas, los motores de búsqueda sabrán interpretar mejor el contenido de una página en función de los titulares y subtítulos.

Hay varios tipos de encabezados, que se diferencian visualmente en el tamaño de la letra que utilizan. La etiqueta en concreto es la H1, para los encabezados más grandes, H2 para los de segundo nivel y así hasta H6 que es el encabezado más pequeño. Pero lo importante, insistimos, es la estructura que denotan. Una página tendrá generalmente un encabezado de nivel 1 y dentro varios de nivel 2. Luego, dentro de los H2 encontraremos si acaso H3, etc. Nunca debemos usar los encabezados porque nos formatean el texto de una manera dada, sino porque nuestro documento lo requiera según su estructura.

Los encabezados implican también una separación en párrafos, así que todo lo que escribamos dentro de H1 y su cierre (o cualquier otro encabezado) se colocará en un párrafo independiente.

Podemos ver cómo se presentan algunos encabezados a continuación.

```
<h1>Encabezado de nivel 1</h1>
```

Los encabezados, como otras etiquetas de HTML, soportan el atributo align. Vemos un ejemplo de encabezado de nivel 2 alineado al centro, aunque repetimos que este formato estético debería hacerse desde CSS.

```
<h2 align="center">Encabezado de nivel 2</h2>
```

Otro ejercicio interesante es construir una página web que contenga todos los encabezados posibles. Se puede ver a continuación.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>headings</title>
</head>
<body>
  <h1>Encabezado de nivel 1</h1>
  <h2>Encabezado de nivel 2</h2>
  <h3>Encabezado de nivel 3</h3>
  <h4>Encabezado de nivel 4</h4>
  <h5>Encabezado de nivel 5</h5>
  <h6>Encabezado de nivel 6</h6>
</body>
</html>
```