

## Clase 01

# Diseño y Programación Web

## Materia: Maquetado y Desarrollo Web

**Docentes contenidistas:** GARCIA, Mabel y PEREZ, Jorge

**Revisión:** Coordinación

# Contenido

¿En qué consiste esta materia? .....	02
Sitios Web y Servidores .....	03
HTML y CSS .....	08
Estructura de directorio .....	10
Herramientas a utilizar .....	12
Etiquetas y atributos .....	13
Anatomía de un archivo HTML .....	16
Para ampliar la información .....	21

# CLASE 1



¡Te damos la bienvenida a  
Maquetado y Desarrollo Web!

**En esta clase vamos a ver los siguientes temas:**

- Sitios Web y Servidores.
- Definiciones HTML y CSS.
- Estructura de directorio.
- Herramientas a utilizar.
- Etiquetas y atributos.
- Anatomía de un archivo HTML.

*Ahora sí... ¡a empezar!*

## ¿En qué consiste esta materia?

*Durante esta cursada daremos los primeros pasos en la confección de una página web.*

*Aprenderemos conceptos, herramientas y técnicas indispensables para el maquetado de un sitio.*



*Nos focalizamos principalmente en dos lenguajes, **HTML** y **CSS**, sus características, reglas y potencial.*

*A medida que obtengamos conocimientos, los volcaremos en nuestra propia web, que servirá como entrega final de la materia.*

# Sitios Web y Servidores

## ¿Qué es un Sitio Web?

Un sitio web, tal vez mejor conocido como “una página de internet”, es una colección de contenidos.

Puede constar de una o varias páginas e incluir texto, imágenes, videos, audio y otros tipos de medios, usualmente organizados en un directorio.

Tal vez la característica más importante de un sitio web es su accesibilidad.

Un sitio web reside en un servidor web, el cual puede ser accedido desde cualquier parte del mundo siempre y cuando se cuente con una conexión a internet.

U.S. INTERNATIONAL CANADA ESPAÑOL 中文

SUBSCRIBE FOR \$0.25/WEEK LOG IN

Sunday, February 12, 2023  
Today's Paper

39°C 39° 28°  
S&P 500 +0.22% ↑

World U.S. Politics N.Y. Business Opinion Science Health Sports Arts Books Style Food Travel Magazine Cooking The Athletic Wirecutter Games

**As Trump Steps Up Attacks, DeSantis Grapples With How to Fight Back**

Gov. Ron DeSantis, who has become the leading G.O.P. alternative to former President Trump, is carefully avoiding conflict as Mr. Trump lobbs insults at him.

But if Mr. DeSantis runs for president, he must decide how to parry taunts and smears without offending Mr. Trump's supporters.

6 MIN READ

**Bernie Sanders has a new Senate role. It could be his final act in Washington.**

7 MIN READ

**Childbirth Is Deadlier for Black Families Even When They're Rich, Study Says**

An analysis of two million births in California showed that the best medical care for mothers and babies is not equally accessible to everyone.

**Infant deaths per 100,000 for mothers who are ...**

**Black** 437 Rate for richest Black mothers

**White** 350 Rate for poorest white mothers

Family income rank 20th 40th 60th 80th ← Poorer Richer →

**This Soup Can Be Many Things, but It's Always Delicious**

The warmth and richness of menudo is a fortifying ritual to finish up the week — or start a new one.

4 MIN READ

**Her foot looked normal. What was causing the excruciating pain?**

6 MIN READ

**Fiction is taking on the dating app revolution with new and bold results.**

6 MIN READ

**Opinion**

THOMAS L. FRIEDMAN

**In 46 Words, Biden Sends a Clear Message to Israel**

6 MIN READ

Sitio web del diario New York Times.

# ¿Qué es un Servidor?

Desde un punto de vista estructural, un servidor web no es particularmente diferente a una PC hogareña.

Es una computadora que cuenta con un sistema de carpetas y archivos para almacenar contenido.

Las principales diferencias entre un servidor y la PC que posiblemente estas utilizando para leer este archivo son las siguientes:

- **Un servidor se mantiene encendido y activo las 24 hs. del día.**  
Esto es indispensable para poder acceder a su contenido en cualquier momento.
- **Un servidor cuenta con una dirección IP que nos permite ubicarlo desde cualquier parte del mundo.**  
Esta dirección de IP normalmente se encuentra registrada para ser sinónima con una dirección web para facilitar el acceso de los usuarios.
- **Un servidor rara vez aloja una sola página.**  
Suelen ser compartidos y alojar cientos o miles de páginas web. Si bien los servidores privados existen son más raros y considerablemente más caros de mantener o contratar.

Debido a estas características, un servidor suele ser un servicio que se contrata a un proveedor. Aunque es enteramente posible crear nuestro propio servidor privado si así lo deseamos.

## ¿Cómo accedemos a los contenidos alojados en un servidor?

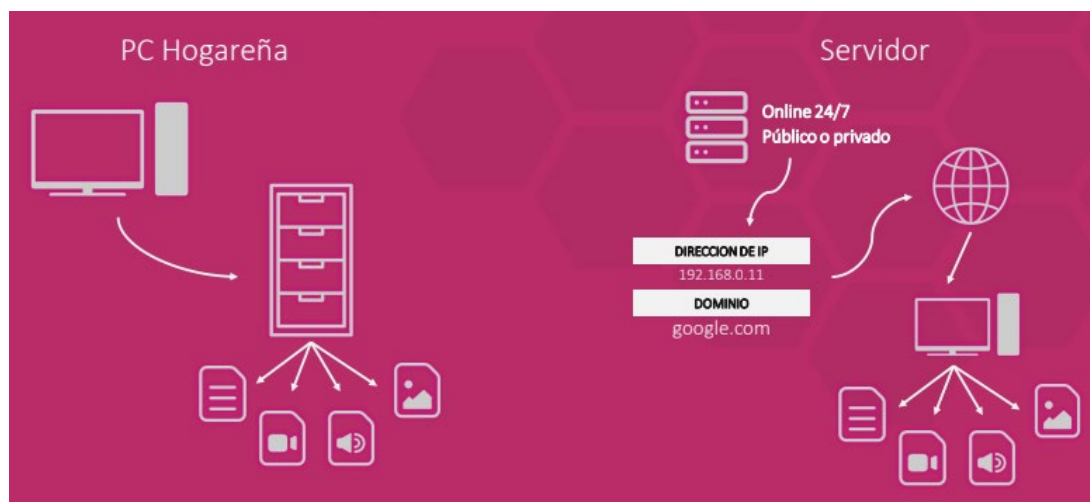
Para poder acceder a los contenidos de un servidor determinado, se hace uso de lo que se llama un cliente.

Un cliente es una pieza de software (programa informático) que se encargará de negociar con el cliente para obtener y mostrar los archivos guardados en el mismo.

Los clientes más comunes son los Navegadores Web o 'Web Browsers', como **Google Chrome**, **Mozilla Firefox** o **Microsoft Edge**.

Al ingresar una dirección web (o una dirección IP) en la barra de navegación de un Browser, éste enviará una solicitud al servidor correspondiente, que la procesa y de aceptarla, devolverá los archivos o el directorio solicitado.

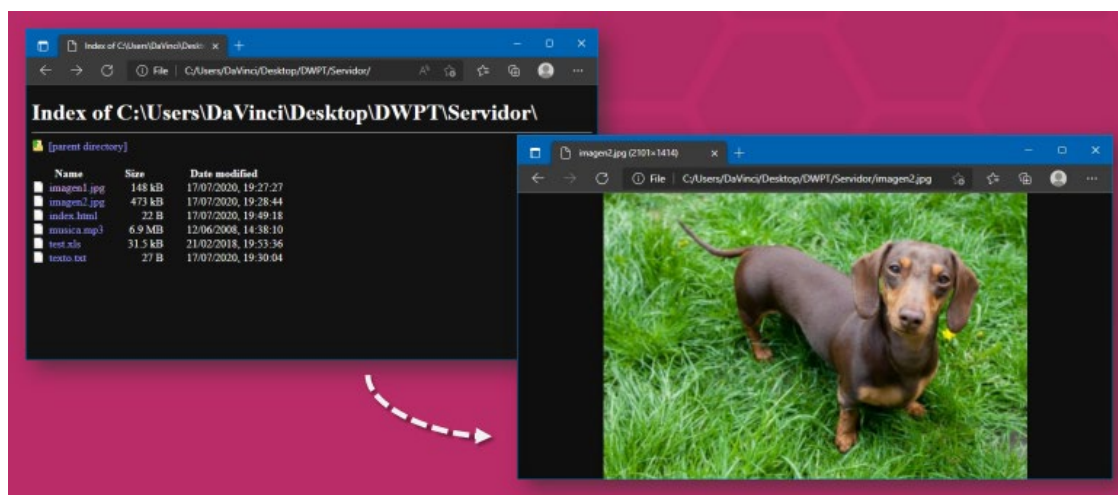
Esta interacción entre los dos sistemas es lo que se llama modelo **Cliente-Servidor**.



*PC Hogareña vs. Servidor Web*

Los navegadores web están diseñados especialmente para mostrar sitios web de manera visual y por lo tanto tienen la capacidad de procesar varios tipos de contenido audiovisual, como pueden ser imágenes, videos o audio, así como también, mostrar los directorios que los contienen.

En caso de que se intente acceder a un formato de archivo que no se pueda visualizar, el browser lo resolverá ofreciendo al usuario la posibilidad de descargar el archivo.



*Un browser listando un directorio de archivos y una imagen alojada en un servidor web.*

Ahora que ya podemos acceder a los archivos guardados en un servidor, solo nos queda organizarlos, darles estructura y mejorar su presentación.

Nuestra página web tomará la forma de un directorio que incluirá todo el contenido que deseemos incluir, pero rara vez encontraremos un sitio web que sea únicamente una lista de archivos.

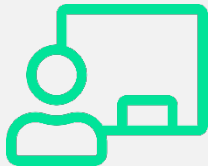
Para poder presentar nuestro contenido de una manera coherente y ordenada utilizaremos dos lenguajes: HTML y CSS.



# HTML y CSS

- **HTML** (HyperText Markup Language):  
Es un lenguaje de marcado que se utiliza para crear páginas web y estructurar su contenido.  
**HTML** permite definir la estructura de una página, incluyendo encabezados, párrafos, imágenes, enlaces y otras estructuras básicas.
- **CSS** (Cascading Style Sheets):  
Es un lenguaje de estilo que se utiliza para definir la apariencia visual de una página web. Con **CSS**, podemos definir colores, tipografías, tamaños de letra, posiciones y muchos otros aspectos visuales de una página web.  
**CSS** permite separar la presentación de la estructura de una página, lo que hace que sea más fácil de mantener y modificar a medida que los requisitos cambian.

*En resumen, podemos decir que  
HTML se utiliza para definir la estructura  
y CSS para definir el diseño.*



Si nuestra página web fuera una casa que tuviéramos que construir desde cero, deberíamos focalizarnos primero en la estructura antes de pensar en la decoración.

Por lo tanto, vamos a dejar **CSS** para más adelante en la cursada y adentrarnos en los detalles de qué es y cómo funciona **HTML**.

# HTML

Como ya vimos más arriba, **HTML** es una sigla que significa:

**HyperText Markup Language.**

Vamos a definirla parte por parte para poder entender mejor a qué hace referencia este nombre.

- **Lenguaje (Language):**  
**HTML** es un lenguaje, es decir que como todo lenguaje tiene reglas que debemos seguir para que sea considerado correcto.
- **Markup (Marcado):**  
**HTML** es un lenguaje de marcado. Lo que significa esto es que cada pieza de texto que incluyamos al utilizarlo deberá ser marcada con una función.  
De esta manera podemos diferenciar una pieza de texto cuya función es ser un encabezado, de una que es un párrafo o una lista.
- **HyperText (Hipertexto):**  
Hipertexto hace referencia a piezas especiales de texto que tienen la habilidad de vincular al usuario con otra pieza de contenido, ya sea dentro del mismo archivo, o un archivo distinto.  
Hipertexto no es otra cosa que los links que usamos cotidianamente al navegar por internet.

**HTML** es un lenguaje de texto plano, es decir texto sin formato.

Por lo tanto, para escribir **HTML**, debemos utilizar programas como Block de Notas, que se manejan únicamente en texto sin formato y no programas como Word que guardan metadatos de formato junto con cada archivo. Más adelante veremos opciones de software a utilizar.

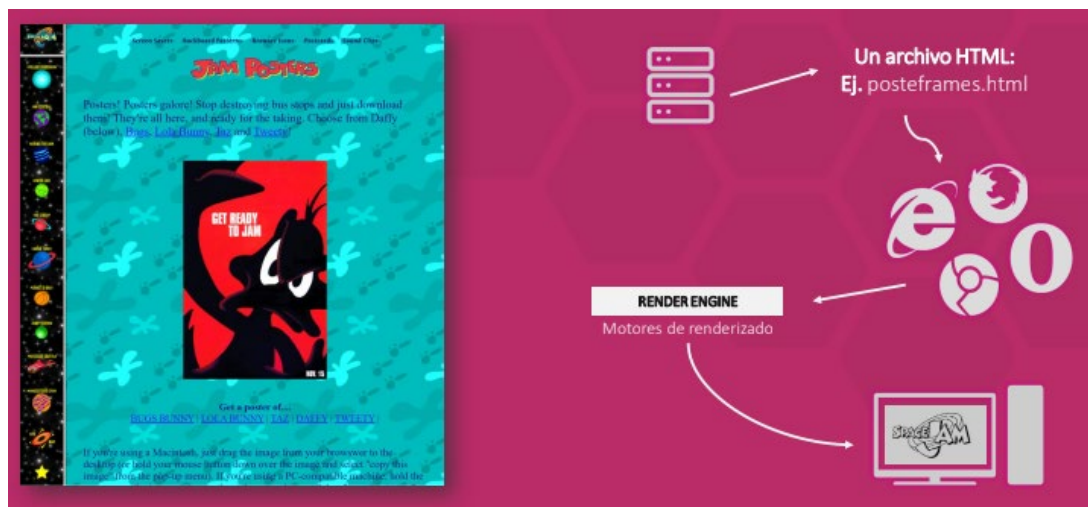
Una vez que tengamos nuestro archivo **HTML** listo, cualquier browser podrá abrirlo e interpretarlo mediante su motor de renderizado, permitiéndonos controlar la manera en la que se muestran los contenidos alojados en nuestro servidor al usuario.

# Estructura de directorio

Como mencionamos anteriormente, una página web usualmente toma la forma de un directorio alojado en un servidor.

Si bien vamos a utilizar **HTML** para la presentación de nuestro contenido, un archivo **HTML** en sí solo puede contener texto: todos los otros contenidos a utilizar ya sean imágenes, videos o música, deben estar guardados en el mismo directorio.

De esta manera, al acceder a un archivo **HTML**, éste llamará a todos los otros archivos necesarios en tiempo real para confeccionar el contenido deseado.



*Un browser accede a un archivo .html y muestra su contenido*

## index.html

Para evitar que el usuario tenga que memorizar los nombres de cada archivo **HTML** que compone un sitio web, siempre se utiliza una página de índice o página principal. Este archivo debe tener obligatoriamente el nombre index y será la página que se abre **automáticamente** al ingresar a ese directorio.

Tener un archivo **index.html** evita que el usuario pueda visualizar los contenidos del directorio directamente y nos otorga control absoluto de la experiencia del contenido.

## Estructura del directorio



Ahora que sabemos cómo se utilizan los archivos **HTML** para organizar el contenido de nuestra página web, es importante mencionar algunas reglas para mantener organizado el directorio de la misma. El hecho de que el usuario no lo pueda ver directamente, no es excusa para que sea un desorden.

Como regla principal, los únicos archivos que queremos sueltos en la raíz de nuestro directorio son archivos .html (incluyendo, pero no limitados a nuestro index.html).

Todos los demás archivos que iremos incorporando a medida que desarrollamos nuestro sitio, deben encontrarse adecuadamente organizados dentro de carpetas.

Name	Size	Date modified
css/		25/02/2023, 21:05:58
img/		25/02/2023, 21:05:59
argumento.html	10.2 kB	09/06/2022, 22:15:50
gradients.html	425 B	10/06/2022, 19:36:53
index.html	3.9 kB	17/11/2022, 20:09:48
reparto.html	3.4 kB	10/06/2022, 19:09:13

*El directorio de una página web, conteniendo varias secciones con sus respectivos archivos .html, un archivo index.html para obrar de página principal y carpetas para organizar el resto de los archivos*

Otro punto importante a tener en cuenta es que los nombres de todos estos archivos deben ser aptos para web, lo que significa que **únicamente** pueden contener letras de la a a la z, números del 0 al 9, guiones bajos (\_) y guiones medios (-). En ningún caso debemos incluir espacios, caracteres con acentos o símbolos.

Por otro lado, si bien no es obligatorio, es muy recomendable no utilizar letras mayúsculas en ningún caso.

# Herramientas a utilizar

## Editores de texto

Tanto **HTML** como **CSS** son lenguajes de texto plano. Para poder crear archivos de este tipo requerimos editores que se manejen con este lenguaje.

Algunos ejemplos de software a utilizar son:

- **Bloc de notas** (disponible en toda instalación de Windows)
- **Notepad++**
- **Sublime Text**
- **Visual Studio Code** (el que verán en los ejemplos de la cursada)

## Navegadores Web

Utilizaremos los navegadores web para visualizar nuestras páginas. Debido a que la interpretación de esas páginas puede variar de navegador en navegador, es recomendable tener varios instalados en simultáneo.

Alguno ejemplos:

- Google Chrome
- Mozilla Firefox
- Safari
- Microsoft Edge
- Opera

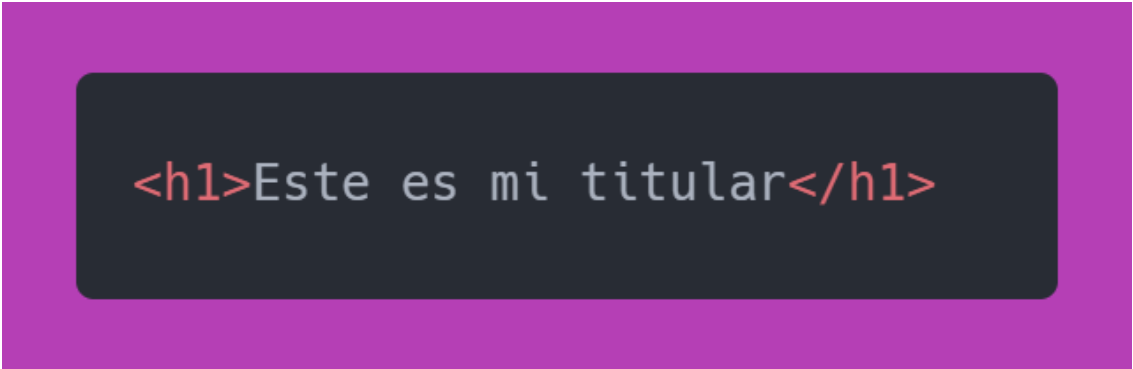
# Etiquetas y atributos

## Anatomía de una etiqueta

**HTML** utiliza etiquetas para asignar funciones a cada pieza de texto que la compone.

Cada etiqueta tiene un nombre único, que indica la función a designar.

En el siguiente ejemplo, utilizaremos la etiqueta **<h1>**, que designa una pieza de texto como un encabezado.

El diagrama muestra un recuadro rectangular con un fondo morado. Dentro de este recuadro hay un rectángulo más pequeño con un fondo gris oscuro. En el centro del rectángulo gris oscuro se encuentra el código HTML `<h1>Este es mi titular</h1>`. El código está escrito en un color rojo claro, con los caracteres de apertura y cierre de la etiqueta (<h1> y </h1>) resaltados en un rojo más intenso que el texto central.

Una etiqueta cuenta con dos partes, ambas envueltas entre símbolos **<>**:

- Una etiqueta de apertura, que consta únicamente del nombre.
- Una etiqueta de cierre, que incluye una barra invertida (/) antecedita al nombre.

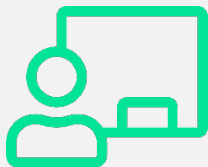
El contenido a marcar siempre se posiciona en el medio.

Los nombres de cada etiqueta son importantes y **siempre van en minúscula**.

Existen etiquetas que no es necesario cerrar, veremos ejemplos más adelante

Las etiquetas **se pueden anidar**, es decir poner una dentro de la otra.

```
<div>
<p>Primer párrafo de texto</p>
<p>Segundo párrafo de texto</p>
</div>
```



En este ejemplo, vemos dos etiquetas **<p>** (párrafo) anidadas dentro de una etiqueta **<div>** (división).

Esta es la base para construir todo archivo html, por eso es importante:

**Ser prolijo y siempre cerrar nuestras etiquetas.**

# Atributos

Una etiqueta puede tener atributos; éstos, se usan para definir parámetros que la afectan.

Los atributos siempre se escriben con un nombre clave, seguido de =, seguido del valor entre comillas simples (') o dobles (").

```
<a href="perrito.html">Esto es un link</a>
```

Una etiqueta puede tener **múltiples atributos**.

Los atributos **siempre van en la etiqueta de apertura** y **separados por espacios**.

## Void Elements - Elementos Vacíos

Algunas etiquetas no requieren envolver contenido y por lo tanto no necesitan una etiqueta de cierre.

El ejemplo más prevalente es la etiqueta **<img>** que se utiliza para definir el área que ocupará una imagen y su contenido se define únicamente con atributos.

```

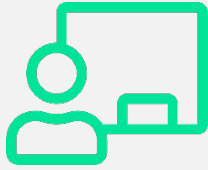
```

Las etiquetas como estas, que no requieren un cierre, se denominan **Void Elements** (Elementos vacíos).

Otros ejemplos incluyen las etiquetas **<hr>** (Línea horizontal) y **<br>** (Corte de línea).



# Anatomía de un archivo HTML



Ahora que sabemos cómo funcionan las etiquetas, veamos un ejemplo de un archivo html con sus componentes básicos.

Traten de identificar todas las etiquetas y atributos presentes.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Un Perrito</title>
</head>
<body>
  <h1>Encabezado</h1>
  <p>Una foto de un perrito</p>
  
</body>
</html>
```

## Declaración de documento - !DOCTYPE

```
<!DOCTYPE html>
```

Esta etiqueta es obligatoria, sin ella, el código no es válido.

Define qué versión de **HTML** estamos usando.

En nuestro caso usaremos **HTML5**, por lo que nuestra etiqueta es simplemente el nombre y el atributo **html** sin ningún valor.

## Etiqueta <html>

```
<html Lang="es">  
  
</html>
```

Esta es nuestra etiqueta principal, dentro, pondremos todo nuestro código HTML

Lleva el atributo **lang**, que define el lenguaje principal en el que presentaremos nuestro contenido.

Esto es importante para lectores de pantalla y accesibilidad general.

El valor es un código de lenguaje estandarizado (ISO 639-1).

Dos letras, ej. **en** para inglés, **es** para español.

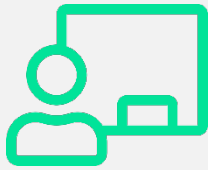
## Etiquetas <head> y <title>

```
<head>  
  <title>Un Perrito</title>  
</head>
```

Todo lo que incluyamos en esta etiqueta es Información para el cliente (Por ejemplo, el navegador donde veremos la página).

Nada de esto será visible al usuario, a menos que el cliente decida mostrarlo, como, por ejemplo, en el nombre de una pestaña de Chrome.

Es obligatorio incluir la etiqueta **<title>**, que define el título de la página y cómo se indexan en buscadores.



Más adelante, vamos a agregar más cosas acá.

## Etiqueta <body>

```
<body>
  <h1>Encabezado</h1>
  <p>Una foto de un perrito</p>
  
</body>
```

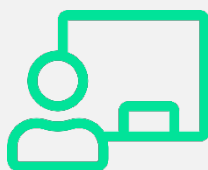
Todo lo que incluyamos en esta etiqueta es Información para el usuario, es decir, se traducirá a contenido visible en el navegador.

El 99% de lo que trabajemos en **HTML**, va a ser dentro de esta etiqueta.



Llegamos así al final de esta primera clase en la que vimos:

- Sitios Web y Servidores.
- Definiciones HTML y CSS.
- Estructura de directorio.
- Herramientas a utilizar.
- Etiquetas y atributos.
- Anatomía de un archivo HTML.



Recuerden visualizar el **desafío semanal** que encontrarán en el aula y conectarse a la clase en vivo de esta semana.

**¡Hasta la próxima clase!**

## Para ampliar la información:

Mozilla Developers Network, Introducción a HTML:

[https://developer.mozilla.org/es/docs/Learn/HTML/Introduction\\_to\\_HTML/Getting\\_started](https://developer.mozilla.org/es/docs/Learn/HTML/Introduction_to_HTML/Getting_started)

W3 Schools. Anatomía de un archivo HTML:

[https://www.w3schools.com/html/html\\_intro.asp](https://www.w3schools.com/html/html_intro.asp)