

TECNICATURA  
UNIVERSITARIA  
EN PROGRAMACIÓN  
UTN-FRC



**UTN**   
Facultad Regional Córdoba

# TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN

## **PROGRAMACIÓN III**

### Unidad Temática 1: Programación Web

Material de Estudio

2<sup>do</sup> Año - 3<sup>er</sup> Cuatrimestre

2020



V.0.1

## Índice

Conceptualización de las estructuras involucradas en la transferencia de datos entre ordenadores.	3
Conceptos de HTTP, IP y Protocolos	5
Presentación de los lenguajes involucrados en el desarrollo Web tradicional.	9
Introducción al HTML; sus controles principales, declaraciones y manipulación.	11
Ejemplo simple de una página html.....	11
Etiquetas HTML.....	12
Estructura de página HTML.....	12
Entornos de desarrollo .....	12
Etiquetas HTML.....	13
Títulos .....	13
Párrafos .....	14
Links .....	15
Imágenes.....	16
Botones.....	17
Listas .....	18
Elementos sin contenido.....	19
Filete horizontal .....	19
Salto de línea.....	20
Atributos HTML.....	20
Formato de texto.....	21
Elementos <b> y <strong>.....	21
Elementos <i> y <em>.....	22
Elementos <small> y <mark> .....	23
Elementos <del> y <ins> .....	23
Elementos <sub> y <sup> .....	24
Comentarios HTML.....	24
Más sobre Links.....	24
Más sobre imágenes .....	25
Tablas.....	25

Elementos de bloque y de línea.....	29
Elementos de Bloque.....	29
Elementos en Línea.....	30
El elemento <div> .....	30
El elemento <span>.....	31
Block, Inline, Inline-block .....	31
Layout (diseño) .....	32
Formularios.....	32
Introducción a CSS; sus estructuras principales para la manipulación visual del código HTML.	<b>37</b>
Sintaxis CSS .....	39
El atributo style.....	39
Color de fondo de la página.....	40
Color de texto .....	40
Fuentes.....	40
Tamaño de fuente.....	40
Alineación de texto .....	41
Colores .....	41
Color de fondo .....	41
Color de texto .....	42
Color de borde.....	42
Image floating .....	43
Imagen de fondo.....	44
CSS padding.....	44
CSS margin .....	45
Box Model.....	45
Atributo Id .....	45
Atributo class .....	46
Ejemplo de estilo aplicado a clases .....	46
Múltiples clases .....	47
Conclusión Unidad I .....	48
Bibliografía .....	49

## Conceptualización de las estructuras involucradas en la transferencia de datos entre ordenadores.

Para entender la **programación web** es necesario primero conocer o tener alguna noción sobre las estructuras involucradas en la **transferencia de datos entre computadoras**.

Esta transferencia de datos, es decir, la comunicación entre dos o más computadoras se lleva a cabo a través de lo que se llama comúnmente **red de computadoras**. La más conocida red es **Internet** o, como también se la denomina, la red de redes.

Vamos a ver conceptos de red de computadoras, de Internet y cómo funcionan los elementos que las constituyen, a modo introductorio para entender cómo funcionan los sistemas para la Web.

Para comenzar vamos a definir que “(...) *una red de computadoras, también llamada red de ordenadores, red de comunicación de datos o red informática, es un conjunto de equipos nodos y software conectados entre sí por medio de dispositivos físicos o inalámbricos que envían y reciben impulsos eléctricos, ondas electromagnéticas o cualquier otro medio para el transporte de datos con la finalidad de compartir información recursos y ofrecer servicios (...)*” (Tanenbaum, 2003)

La finalidad principal para la creación de una red de ordenadores es la de compartir recursos e **información** a grandes distancias, asegurar la confiabilidad y la disponibilidad de la información (características propias que definen lo que es información), aumentar la velocidad de transmisión de los datos y reducir los costos(CENS, 2018)-

Si bien no es un curso de redes de computadora es necesario conocer los elementos físicos en los que trabajan los sistemas que vamos a realizar. Con ello queremos decir que si bien podemos abstraernos de esta tecnología es muy útil tener alguna noción de lo que pasa por detrás de bambalinas.

La comunicación por medio de una red se lleva a cabo en dos diferentes categorías: una **capa** denominada **física** y otra **lógica**.

La **capa física** incluye todos los elementos de los que hace uso un equipo para comunicarse con otro equipo dentro de la red, por ejemplo, tarjetas de red, los cables, las antenas, etc. esto es todo lo que diremos de la capa física si te interesa saber más puedes leer autores como Tanenbaum en donde habla de redes y el modelo OSI (modelo de interconexión de sistemas abiertos, protocolos de comunicación, etc.).

Con respecto a la **capa lógica** la comunicación se rige por normas muy rudimentarias que por sí mismas resultan de escasa utilidad. Sin embargo, haciendo uso de dichas normas es posible construir los denominados protocolos que son normas de comunicación más complejas (de alto nivel) capaces de proporcionar servicios útiles. Ya vamos a ver más sobre esto (Tacoa, 2016).

Los **protocolos** son un concepto muy similar al de los idiomas de las personas. Es decir si dos personas hablan el mismo idioma, y respetar ciertas reglas (tales como hablar y escucharse por turnos), es posible comunicarse y transmitir ideas e información. Ese es el modelo de un protocolo.

La diferenciación que se hace entre las capas físicas y lógicas hace que sea posible modificar la tecnología subyacente a una de las capas sin afectar a la otra. Esa es la idea de diseñar este modelo a través de capas. Por ejemplo, se podrían modificar como se ha hecho con la forma de transmisión de datos, con los avances tecnológicos, por ejemplo cambiando líneas telefónicas por fibra óptica, o sea cambiando la capa física sin tocar las capas lógicas superiores es decir sin tener que romper la red en su funcionamiento.

Entonces, para formar una red se requieren elementos de **hardware**, **software** y **protocolos**. los elementos físicos se clasifican en dos grupos: los dispositivos de usuario final ( llamados también Hosts) y dispositivos de red. Entre los dispositivos de usuario final podemos enumerar computadoras, impresoras, escáneres, y demás elementos que brindan servicios directamente al usuario. los segundos ( dispositivos de red) son todos aquellos que conectan entre sí a los dispositivos de usuario final posibilitando su intercomunicación (Farías, 2013).

Concluyendo vamos a decir que el fin de una red es interconectar los componentes Hardware de una red, y por tanto, principalmente, los ordenadores individuales también denominados Host, a los equipos que ponen los servicios en la red, los servidores Web, entre otros.

Ahora bien, hemos hablado de la tecnología de lo que es redes de computadoras pero no hemos nombrado a la Internet.

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 (Guillamón, 2009).

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 y utiliza Internet como medio de transmisión («Internet, n.». Oxford English Dictionary (Draft edición). Marzo de 2009).

Para hacer más notable esta distinción vamos a nombrar algunos otros servicios y protocolos en Internet aparte de la web por ejemplo el envío de correo electrónico (protocolo SMTP), transmisión de archivos (protocolo FTP), para nombrar algunos.

## Conceptos de HTTP, IP y Protocolos

Bueno ya hemos dejado en claro que significa la palabra protocolo. Por lo tanto, ahora podemos hablar de la familia de protocolos de Internet.

Es necesario tener nociones de estos protocolos ya que al programar aplicaciones para la web tenemos que tener presentes algunas reglas que los caracteriza.

Esta familia de protocolos es un conjunto de protocolos de red en los que se basa Internet y permite la transmisión de datos entre computadoras como hemos dicho.

Entre los principales podemos nombrar el conjunto de protocolos tcp/IP que hace referencia a los dos protocolos más importantes que componen la Internet, que fueron los primeros en definirse y qué son los más utilizados.

Por un lado **TCP** (protocolo de control de transmisión). Sin entrar en mucho detalle, tcp se usa para crear conexiones entre computadoras a través de las cuales pueden enviarse un flujo de datos. Por la forma en la que está implementado este protocolo los datos serán entregados en su destino sin errores y en el mismo orden en el que se transmitieron. Esto qué quiere decir, que es un protocolo orientado a conexión, ya que el cliente y el servidor deben anunciarse y aceptar la conexión antes de comenzar a transmitir los datos entre ellos. O sea, hay un intercambio de mensajes entre ellos para abrir una línea de conexión que permanece abierta durante toda la comunicación (por eso es orientado a conexión) (DSI, 2007).

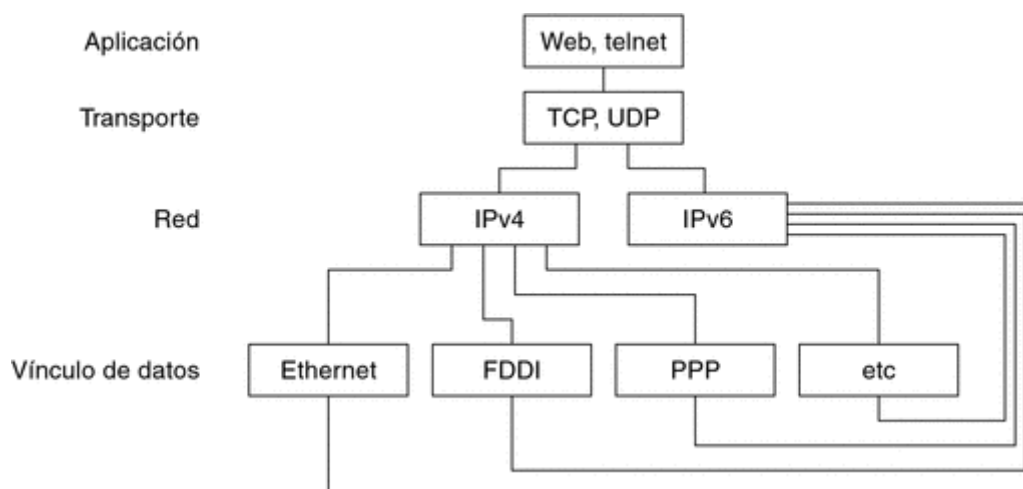


Figure 1: recuperado de Oracle (10/3/2020) Parte II Administración de TCP/IP.

Por otro lado el protocolo **IP** (Internet Protocol) es un protocolo cuya función principal es el uso direccional en origen o destino de comunicación para transmitir datos mediante un protocolo no orientado a conexión que transfiere paquetes conmutados a través de distintas redes físicas previamente enlazadas según la norma OSI.

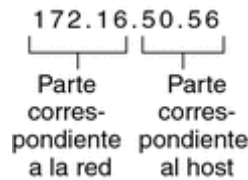


Figure 2: Recuperado de Oracle (10/3/2020) Parte II Administración de TCP/IP

Algo importante del diseño del protocolo IP es que se realizó suponiendo que la entrega de los paquetes de datos sería no confiable por eso se tratara de realizar del mejor modo posible mediante técnicas de enrutamiento sin garantías de alcanzar el destino final pero tratando de buscar la mejor ruta entre las conocidas por la máquina que está usando IP.

Para entender mejor esta distinción entre dos protocolos uno tcp y otro IP habría que analizar el modelo de capas OSI que no vamos a ver. Lo que sí vamos a decir es que hay una jerarquía entre capas y el protocolo IP pertenece a una capa denominada de red que está por encima de una capa denominada de transporte en dónde se encuentra tcp.

Entonces, en conclusión, se utiliza la combinación de estos dos protocolos para la comunicación en Internet, en dónde TCP aporta la fiabilidad entre la comunicación e IP la comunicación entre distintas computadoras ya que las cabeceras de IP (cabecera por ser una parte el protocolo, que aquí no tiene importancia) contienen las direcciones de las máquinas de origen y destino, llamadas direcciones IP. Estas direcciones serán usadas por los enrutadores (routers) para decidir el tramo de red por el que se enviarán en los paquetes.

Para entender mejor el funcionamiento de la Internet vamos a decir que dentro de la red de redes que es Internet debe existir un mecanismo para conectar dos computadoras. Este mecanismo lo provee el protocolo de Internet qué hace que un paquete de una computadora llegué a la otra de manera segura a través del protocolo tcp y que llegue a destino a través de las direcciones IP.

Para terminar, una dirección IP es un número que identifica de manera lógica y jerárquica una interfaz de un dispositivo dentro de una red que utiliza el protocolo de internet.

Todas las computadoras en internet tienen una dirección de IP. A modo informativo vamos a decir qué existe otro sistema que se denomina sistema de

nombres de dominio o DNS que asocia nombres comunes a direcciones IP por ejemplo la dirección [www.lavoz.com.ar](http://www.lavoz.com.ar) tiene asociado un número de IP correspondiente pero este mecanismo existe para que sea más fácil llegar a esa página web sin tener que recordar el número de IP.

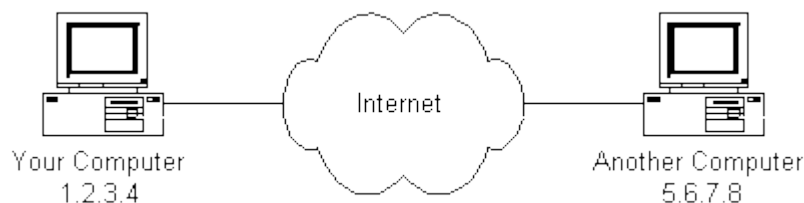


Figure 3: . Recuperado de: Oracle (10/3/2020) Parte II Administración

Bien, con esto concluimos una introducción de lo que es la comunicación entre computadoras en internet. Si bien parece algo complejo es de mucha importancia conocer estos mecanismos con los que trabajamos.

Por ejemplo si queremos comunicarnos con una base de datos en Internet y nos pide un número de IP ya sabemos de qué se trata. También sabemos que podemos comunicarnos a cualquier computadora conectada en red a través de estos mecanismos y que nos abstraemos de cómo se hace esto. Es decir no nos tenemos que preocupar de manejar errores o interferencias en la comunicación.

Por otro lado, antes de continuar con el desarrollo web tradicional tenemos que aprender un nuevo protocolo que es imprescindible para la comunicación en la web.

Ya vimos que la estructura de red se maneja en capas. También mencionamos qué hay una capa de red en dónde está el protocolo IP, una capa Superior de transporte en dónde está el protocolo tcp y ahora vemos una nueva capa que es la de aplicación en dónde se usa el protocolo http.

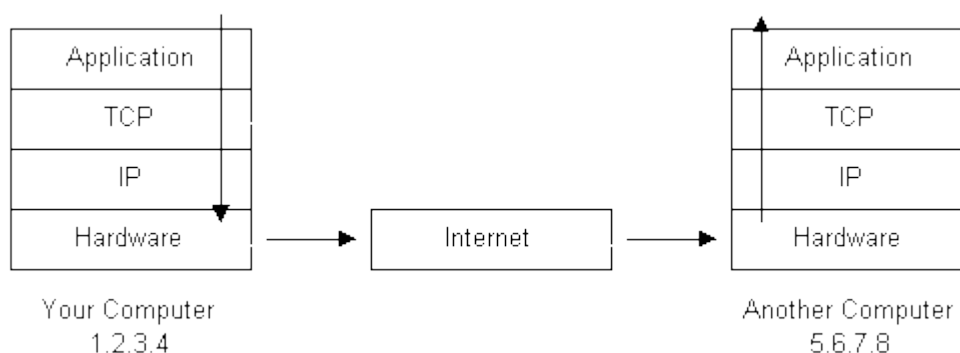


Figure 4: Recuperado de Oracle (10/3/2020) Parte II Administración de TCP/IP.

El protocolo de transferencia de hipertexto (en inglés, Hypertext Transfer Protocol, abreviado **HTTP**) Es el protocolo de comunicación que permite las transferencias de información en la web.



Es muy importante saber qué http es un protocolo sin estado, es decir, no guarda ninguna información sobre conexiones anteriores. Luego veremos las implicancias de esto.

Una descripción importante del protocolo es que es orientado a transacciones y sigue el esquema de petición/respuesta entre un cliente y un servidor. El cliente realiza una petición enviando un mensaje con cierto formato al servidor. El servidor le envía un mensaje de respuesta. Para hacerlo más concreto un cliente podría ser un navegador web y un servidor podría ser una aplicación en un servidor web corriendo en Internet.

Para que entendamos la forma de cómo se programa una aplicación web necesitamos en alguna medida entender los mensajes http.

Los mensajes http son en texto plano lo que los hace más legible y fácil de depurar. Estos tienen la siguiente estructura:

Primero, hay una línea inicial en donde se diferencian todos modos dependiendo de si son peticiones y respuestas. Para las peticiones la línea comienza con una acción requerida por el servidor, a esto se le denomina método de petición (ya veremos cuáles son a continuación) seguido de la url del recurso y la versión http que soporte al cliente. Lo importante el **método** de petición y la **URL** (Uniform Resource Locator o localizador de recursos uniforme..

Para las **respuestas** la línea comienza con la versión de http seguido por un código de respuesta y con una frase asociada a dicho retorno.

También los mensajes tienen una cabecera que son metadatos con información varia y el cuerpo de mensaje que es opcional. Típicamente este cuerpo tiene los datos que se intercambian entre el cliente y el servidor.

Los métodos de **petición** (o también llamados verbos) son varios. Cada método indica la acción que desea que se efectúe sobre el recurso identificado lo que este recurso representa depende de la aplicación del servidor.

Los métodos que vamos usar son los de **get** y **post**. Cabe destacar que hay convenciones en donde se utilizan muchos más métodos de los que vamos usar nosotros. Para más información podes leer lo que es Rest.

El método **get** solicita una representación del recurso especificado. las solicitudes que usan get sólo deben recuperar datos y no deben tener ningún otro efecto.

El método **post** envía los datos para que sean procesados por el recurso identificado. Los datos enviados se incluirán en el cuerpo de la petición. esto puede resultar en la creación de un nuevo recurso o de las actualizaciones de los recursos existentes.

Para finalizar con la explicación de http vamos a nombrar algunos códigos de respuesta. los que empiezan con 2 por ejemplo el 200 representa una respuesta correcta es decir que indica que la petición ha sido procesada correctamente. Otro ejemplo es la respuesta 404 Qué significa errores causados por el cliente por ejemplo que el recurso no se encuentre. finalmente las respuestas que comienzan con 5 por ejemplo el 500 son errores causados por el servidor. Esto indica que ha habido un error en el procesamiento de la petición a causa de un fallo en el servidor.

Todo lo que hablamos de http lo vamos a usar al momento de programar una aplicación web. Vamos a ver que al presionar por ejemplo un link se hace una petición get al servidor para buscar otra página lo que resultará en una respuesta 200 si se encuentra la página o en un 404 si no se encuentra. También veremos que al llenar un formulario y enviarlo al servidor lo haremos a través de una petición post.

Estos últimos ejemplos son para que veas la utilidad de conocer este protocolo qué es el que más vamos con mayor detalle en las últimas unidades.

Los otros protocolos el de tcp/IP son necesarios para configuraciones de conexiones pero a nivel aplicación no los utilizaremos. Eso no significa que no haya que entenderlos pues nos facilitarán la solución de problemas, por ejemplo, al conectarnos a una base de datos desde la aplicación web del servidor.

## Presentación de los lenguajes involucrados en el desarrollo Web tradicional.

Ya hemos discutido la infraestructura subyacente a las conexiones entre computadoras, hemos visto lo que significa Internet y los protocolos que se usan para las comunicaciones.

También mencionamos que existe un modelo en el que se basan las comunicaciones, hablamos de emisores y receptores. en nuestra jerga los llamaremos clientes y servidores como hemos mencionado anteriormente.

Dentro de lo que es la web, podemos identificar como cliente principalmente al navegador web sin importar en que dispositivo se encuentre; puede ser un dispositivo móvil o una computadora. Dejemos de lado el tema del servidor por el momento, lo retomaremos en unidades próximas.

Los **navegadores web** reconocen tres lenguajes y nada más que tres. Estos son **html**, **css** y **javascript**.

Si bien uno puede confundirse en el desarrollo web como si los navegadores entendieran otros lenguajes, donde se nombran por ejemplo php, Java, python, mvc.net, entre otros. En realidad, lo que hacen estos lenguajes es responder a las peticiones devolviendo respuestas en los tres lenguajes que acabamos de nombrar,

o sea, html, css y javascript. Es decir no envían al navegador web código php, ni Java, ni python. **Es muy importante identificar esta distinción.**

Por eso es imperativo que primero veamos los lenguajes que entiende el navegador web para luego especializarnos en lenguajes que se ejecutan en el servidor (en nuestro caso será asp.Net MVC)

Existe una librería o framework que posibilita la utilización de javascript del lado del servidor llamada Node.js pero nos vamos a enfocar nada más en el lado del cliente.

En el desarrollo web tradicional podemos decir que el servidor le entrega al cliente, el navegador, lo que se denomina una página web. Una página web está compuesta de contenido tanto de texto como imágenes cómo tipografía cómo videos, etc..

Como dijimos anteriormente el navegador entiende html css y javascript solamente.

Html es un lenguaje de maquetado, ya lo veremos con más detalle en la sección siguiente, qué se encarga del contenido de la página web. A grandes rasgos define la estructura y el contenido de lo que se lee o se vé.

Por otro lado la función principal de las hojas de estilo css es la de agrupar el formato de cómo se verá la página web.

Por último javascript es un lenguaje de programación con el cual se puede modificar los elementos de una página web de forma dinámica.

Una acotación importante es la separación entre estilo y contenido que se logró con la introducción de las hojas de estilo. En un principio el estilo se especificaba directamente sobre el documento html (que por compatibilidad aún se puede pero se mezclaría responsabilidades ajenas a las propias de html).

Esto tiene que ver con las raíces de html, html es un lenguaje que deriva de otro lenguaje xml en donde se intentó marcar el contenido de un texto de tal forma de identificar entidades semánticas, por ejemplo cuál es el título del documento, cuál es el autor, etc. Html se creó con el mismo concepto en los inicios de la web en donde la Internet consistía en el intercambio de documentos sin muchos agregados.

Es por eso a medida que pasó el tiempo que el estándar html se vio obligado a incorporar marcas de estilo. qué pasaron hacer un lenguaje propio y a desprenderse de html formando las hojas de estilo tratando de respetar la concepción de html.

Veamos a continuación y en detalle qué es html y qué son las hojas de estilo.

## Introducción al HTML; sus controles principales, declaraciones y manipulación.

**Html** significa Hyper text markup Language o lenguaje de marcado de hyper texto. Se denomina de esta forma porque en su concepción se basaba en la vinculación entre documentos de texto en dónde la estructura se define con marcas que no son visibles en el navegador pero si dividen el texto semánticamente (esa es la parte de marcado). Hyper text viene de la palabra hyper link, que hace que se puedan vincular documentos distintos y “navegar” de uno al otro. Esto fué revolucionario en su época.

En resumen, html describe la estructura de página web usando un lenguaje de maquetado a base de marcas de texto. Las veremos a continuación.

Todos los elementos de html constituyen una página web, algunos son visibles y otros no.

Estos elementos son representados mediante etiquetas en el código fuente de la página y no lo que muestra el navegador en la pantalla

Las etiquetas html representan partes del contenido como lo son los títulos, párrafos, tablas, etcétera.

Cómo se dijo anteriormente, los navegadores no muestran las etiquetas html sino que las utilizan para renderizar el contenido.

### Ejemplo simple de una página html

```
<!DOCTYPE html>

<html>

<head>

<title>Título de página</title>

</head>

<body>

<h1>Mi primer titular</h1>

<p>Mi primer párrafo.</p>

</body>

</html>
```

Ya veremos el significado de cada línea de código.

## Etiquetas HTML

Las etiquetas, también llamadas tags (en inglés), son palabras clave rodeadas por los signos menor-mayor

**<etiqueta>el contenido va aquí...</etiqueta>**

Generalmente las etiquetas vienen de a pares, como <p> y </p>

El primer tag del par se llama **tag de apertura** y el segundo, **tag de cierre**

El tag de cierre se escribe igual que el de apertura pero con la barra de división "</...>"

## Estructura de página HTML

Esto es una visualización de la estructura de página

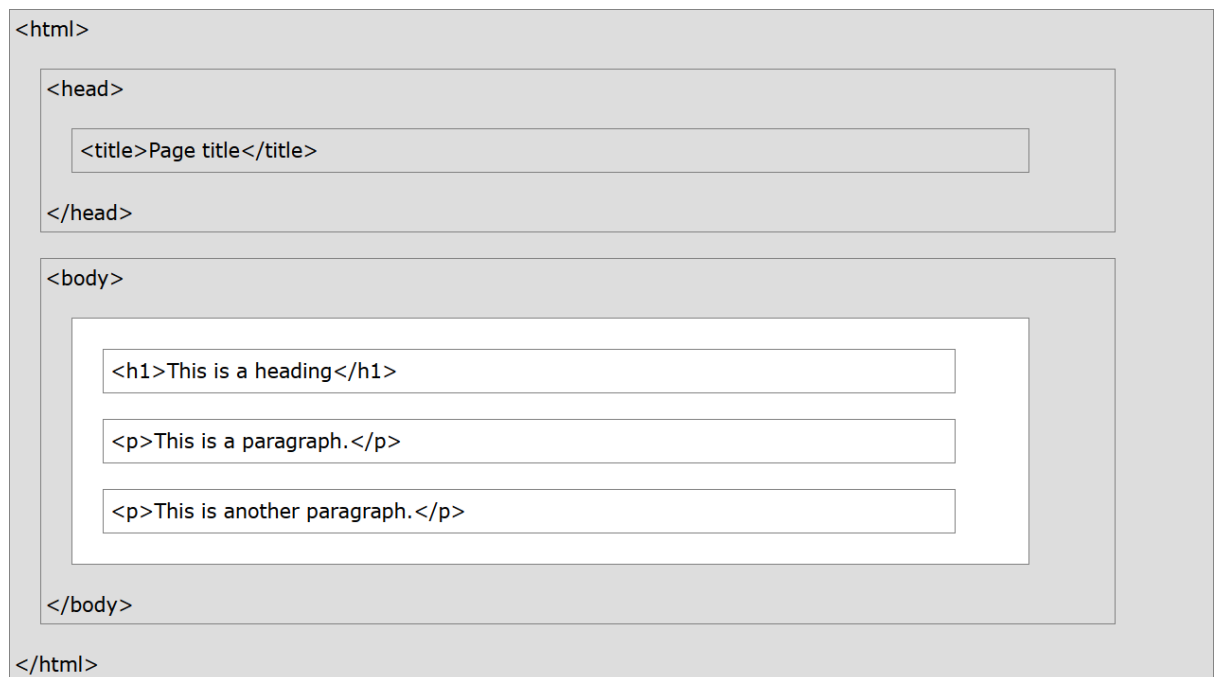


Figure 5: Recuperado de W3schools (10/3/2020)

Las cajas responden a la "anidación de bloques" que establecen las etiquetas. Es decir, todas las etiquetas están dentro de la etiqueta <html> y <title> está dentro del <head>.

## Entornos de desarrollo

Como hemos visto hasta ahora las páginas html son páginas de texto plano con marcas o tags. por lo tanto se puede utilizar cualquier editor de texto para crear dichos archivos. Es decir, el navegador recibe a la página web tal cual está escrita, sin la necesidad de ser compilada o traducida.

En particular, los entornos de desarrollo y algunos editores de texto avanzados reconocen este lenguaje y distinguen el contenido común en contraposición con las marcas ayudando al desarrollador en su tarea de escritura.

Es entonces valioso la utilización de estas herramientas. Algunas muy utilizadas son Atom, brackets, Sublime text, notepad++, etc. Estos ejemplos son editores de texto avanzados. También podemos nombrar algunos editores que ya pasan a ser entornos de desarrollo integrados (IDEs) cómo son visual Studio code, visual Studio, netbeans, entre otros.

## Etiquetas HTML

### Títulos

Los títulos están definidos con las etiquetas desde <h1> hasta la <h6>.

<h1> es el título más importante mientras que <h6> el de menor importancia.

**<h1>This is heading 1</h1>**

**<h2>This is heading 2</h2>**

**<h3>This is heading 3</h3>**

Ejemplo de títulos

**<!DOCTYPE html>**

**<html>**

**<body>**

**<h1>This is heading 1</h1>**

**<h2>This is heading 2</h2>**

**<h3>This is heading 3</h3>**

**<h4>This is heading 4</h4>**

**<h5>This is heading 5</h5>**

**<h6>This is heading 6</h6>**

**</body>**

**</html>**

# This is heading 1

## This is heading 2

### This is heading 3

#### This is heading 4

##### This is heading 5

###### This is heading 6

Figure 6: Recuperado de W3schools (10/3/2020). HTML Tutorial.

### Párrafos

Los párrafos en HTML están definidos por el tag <p>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

### Ejemplo de párrafos

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
</body>
```

```
</html>
```

This is a paragraph.

This is another paragraph.

Figure 7: W3schools (10/3/2020). HTML Tutorial.

### Links

Los links están definidos por el tag <a>

```
<a href="https://www.w3schools.com">This is a link</a>
```

### Ejemplo de link

```
<!DOCTYPE html>
<html>
<body>
<h2>HTML Links</h2>
<p>HTML links are defined with the a tag:</p>
<a href="https://www.w3schools.com">This is a link</a>
</body>
</html>
```

## HTML Links

HTML links are defined with the a tag:

[This is a link](#)

Figure 8: Recuperado de W3schools (10/3/2020). HTML



## Imágenes

Las imágenes están definidas por el tag <img>

```

```

Ejemplo de imagen

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>HTML Images</h2>
```

```
<p>HTML images are defined with the img tag:</p>
```

```

```

```
</body>
```

```
</html>
```

# HTML Images

HTML images are defined with the `img` tag:



Figure 9: Recuperado de W3schools (10/3/2020). HTML Tutorial.

## Botones

Los botones están definidos por el tag `<button>`

`<button>Click me</button>`

Ejemplo de botones

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>HTML Buttons</h2>
```

```
<p>HTML buttons are defined with the button tag:</p>
```

```
<button>Click me</button>
```

```
</body>
```

```
</html>
```

# HTML Buttons

HTML buttons are defined with the button tag:



Click me

Figure 10: Recuperado de W3schools (10/3/2020). HTML Tutorial.

## Listas

Las listas están definidas por <ul> (listas desordenadas con viñetas) o <ol> (listas ordenadas numeradas). Seguido por la etiqueta <li> (list items)

```
<ul>
```

```
<li>Coffee</li>
```

```
<li>Tea</li>
```

```
<li>Milk</li>
```

```
</ul>
```

Ejemplo de listas

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>An Unordered HTML List</h2>
```

```
<ul>
```

```
<li>Coffee</li>
```

```
<li>Tea</li>
```

```
<li>Milk</li>
```

```
</ul>
```

```
<h2>An Ordered HTML List</h2>
```

```
<ol>
```

```
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>
</body>
</html>
```

## An Unordered HTML List

- Coffee
- Tea
- Milk

## An Ordered HTML List

1. Coffee
2. Tea
3. Milk

Figure 11: Recuperado de W3schools (10/3/2020). HTML Tutorial.

### Elementos sin contenido

Existen algunos elementos de HTML que no tienen contenido y son llamados elementos vacíos.

Los elementos vacíos no tienen la etiqueta de cierre.

Vamos a ver a continuación un par de ellos.

### Filete horizontal

En artes gráficas y tipografía, sinónimo de línea o de adorno más o menos lineal que no sea demasiado grueso.

La traducción de hr sería línea horizontal (horizontal ruler).

La etiqueta <hr> define al filete horizontal. Es usado para separar contenido

```
<h1>This is heading 1</h1>
```

```
<p>This is some text.</p>
```

**<hr>**

**<h2>This is heading 2</h2>**

**<p>This is some other text.</p>**

**<hr>**

### Salto de línea

En HTML la etiqueta `<br>` define un salto de línea en el texto.

Utilice `<br>` si quiere una nueva línea sin tener que comenzar con un párrafo nuevo.

**<p>This is<br>a paragraph<br>with line breaks.</p>**

### Atributos HTML

- Los atributos proveen información adicional a los elementos HTML.
- Todos los elementos HTML pueden tener atributos.
- Los atributos proveen información adicional acerca de un elemento.
- Los atributos siempre están especificados en la etiqueta de apertura.
- Los atributos están definidos por el par nombre/valor como por ejemplo nombre = "valor"

### El atributo href

Los links están definidos por el tag `<a>`.

La dirección del link esta especificado por el atributo href

**<a href="https://www.w3schools.com">This is a link</a>**

### El atributo src

Las imágenes están definidos con el tag `<img>`

El nombre de archivo de la imagen de origen está especificado por el atributo src

****

### Los atributos width y height

Las imágenes en HTML tienen un conjunto de atributos para el tamaño, que especifican el ancho y la altura de la imagen.

```

```

El tamaño de la imagen está especificado en pixeles, por ejemplo width="500" significa 500 pixeles de ancho

### El atributo alt

El atributo alt especifica un texto alternativo para las imágenes cuando una imagen no puede ser mostrada, sea por la razón que sea.

Además sirve para ser leída por los lectores de pantalla para personas no videntes.

```

```

### El atributo título

En este caso, el atributo title es agregado a un elemento de tipo <p>. El valor del título puede ser mostrado como un globo de ayuda o tooltip cuando se pasa el puntero del mouse por encima.

```
<p title="I'm a tooltip">
```

This is a paragraph.

```
</p>
```

### Formato de texto

Html usa elementos especiales para darle formato al texto, como negrita o itálica.

### Elementos <b> y <strong>

El elemento <b> se utiliza para transformar en negrita el texto, sin darle ninguna importancia adicional.

```
<b>This text is bold</b>
```

This text is normal.

**This text is bold.**

Figure 12: Recuperado de W3schools (10/3/2020). HTML Tutorial.

El elemento `<strong>` le dá al texto más énfasis que negrita, con una importancia más alta semánticamente.

`<strong>This text is strong</strong>`

This text is normal.

**This text is strong.**

Figure 13: Recuperado de W3schools (10/3/2020). HTML Tutorial.

#### Elementos `<i>` y `<em>`

El elemento `<i>` se utiliza para definir al texto como itálica, sin importancia adicional.

`<i>This text is italic</i>`

This text is normal.

*This text is italic.*

Figure 14: Recuperado de W3schools (10/3/2020). HTML Tutorial.

El elemento `<em>` define al texto como enfatizado, con importancia adicional semántica.

`<em>This text is emphasized</em>`

This text is normal.  
*This text is emphasized.*

Figure 15: Recuperado de W3schools (10/3/2020). HTML Tutorial.

Elementos `<small>` y `<mark>`

El elemento `<small>` define al texto como "pequeño"

`<h2>HTML <small>Small</small> Formatting</h2>`

**HTML Small Formatting**

Figure 16: Recuperado de W3schools (10/3/2020). HTML Tutorial.

El elemento `<mark>` define al texto como resaltado

`<h2>HTML <mark>Marked</mark> Formatting</h2>`

**HTML Marked Formatting**

Figure 17: Recuperado de W3schools (10/3/2020). HTML Tutorial.

Elementos `<del>` y `<ins>`

El elemento `<del>` define al texto como "borrado" o tachado

`<p>My favorite color is <del>blue</del> red.</p>`

My favorite color is ~~blue~~ red.

Figure 18: Recuperado de W3schools (10/3/2020). HTML Tutorial.

El elemento `<ins>` define al texto como de "inserción" o subrayado

`<p>My favorite <ins>color</ins> is red.</p>`



My favorite color is red.

Figure 19: Recuperado de W3schools (10/3/2020). HTML Tutorial

#### Elementos <sub> y <sup>

El elemento <sub> define al texto como subíndice

**<p>This is <sub>subscripted</sub> text.</p>**

This is subscripted text.

Figure 20: Recuperado de W3schools (10/3/2020). HTML Tutorial.

El elemento <sup> define al texto como superíndice

**<p>This is <sup>superscripted</sup> text.</p>**

This is superscripted text.

Figure 21: Recuperado de W3schools (10/3/2020). HTML Tutorial.

#### Comentarios HTML

Los tags de comentario son usados para insertar comentarios en el código HTML.

**<!-- Write your comments here -->**

Notar que el signo de exclamación del tag de apertura no está presente en el de cierre.

Los comentarios no son mostrados en el navegador sino que sirve de ayuda para documentar el código fuente.

#### Más sobre Links

Ya vimos que los links están definidos por el tag <a>

**<a href="https://www.w3schools.com">This is a link</a>**

Un link no sólo puede ser texto sino que cualquier elemento HTML puede ser un enlace.

```
<a href="default.asp">  
      
</a>
```

Se puede hacer que un link no sólo apunte a una página externa sino que enlace alguna parte de la misma página.

Es decir, cuando se haga click en el link, la página hará un scroll hasta la zona indicada por algún "bookmark".

```
<h2 id="C4">Chapter 4</h2>
```

El bookmark anterior puede ser navegado con el siguiente link.

```
<a href="#C4">Jump to Chapter 4</a>
```

[Más sobre imágenes](#)

Ya vimos que las imágenes están definidas por el tag <img>

```

```

El tamaño puede estar definido de dos formas.

```
  

```

[Tablas](#)

Una tabla está definida por el tag <table>. A su vez, cada fila está definida por la etiqueta <tr>. Cada encabezado por <th> y cada celda por <td>

```
<table style="width:100%">
```

```
<tr>
  <th>Firstname</th>
  <th>Lastname</th>
  <th>Age</th>
</tr>
<tr>
  <td>Jill</td>
  <td>Smith</td>
  <td>50</td>
</tr>
</table>
```

Cada elemento `<td>` puede contener cualquier otro elemento, como texto, imagen, listas u otras tablas.

Una tabla por defecto se visualiza sin bordes. Para utilizar bordes se debe usar el atributo `border`

```
<table border="1">
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
</table>
```

Se puede hacer que una celda se estire para ocupar el espacio de varias celdas (similar a una combinación de celdas de una planilla de cálculo pero con la salvedad de que la celda no se combina sino que se estira).

Para hacer que la celda se estire a más de una columna use el atributo `colspan`.

Para que se estire a más de una fila use el atributo `rowspan`.

### Ejemplo de colspan

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>Cell that spans two columns</h2>
```

<p>To make a cell span more than one column, use the colspan attribute.</p>

```
<table border="1" style="width:100%">
```

```
<tr>
```

```
<th>Name</th>
```

```
<th colspan="2">Telephone</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Bill Gates</td>
```

```
<td>55577854</td>
```

```
<td>55577855</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

## Cell that spans two columns

To make a cell span more than one column, use the colspan attribute.

Name		Telephone	
Bill Gates	55577854	55577855	

Figure 22: Recuperado de W3schools (10/3/2020). HTML Tutorial.

### Ejemplo de rowspan

```
<!DOCTYPE html>
```

```
<html>

<body>

<h2>Cell that spans two rows</h2>

<p>To make a cell span more than one row, use the rowspan
attribute.</p>

<table border="1" style="width:100%">

  <tr>

    <th>Name:</th>

    <td>Bill Gates</td>

  </tr>

  <tr>

    <th rowspan="2">Telephone:</th>

    <td>55577854</td>

  </tr>

  <tr>

    <td>55577855</td>

  </tr>

</table>

</body>

</html>
```

<b>Cell that spans two rows</b>	
To make a cell span more than one row, use the rowspan attribute.	
<b>Name:</b>	Bill Gates
<b>Telephone:</b>	55577854
	55577855

Figure 23: Recuperado de W3schools (10/3/2020). HTML Tutorial.

El pie de tabla es igual a un pie de foto, es decir, es una leyenda (caption, en inglés) que acompaña la tabla para especificar detalles a modo de título.

```
<table border="1" style="width:100%">  
  <caption>Monthly savings</caption>  
  <tr>  
    <th>Month</th>  
    <th>Savings</th>  
  </tr>  
  <tr>  
    <td>January</td>  
    <td>6247$</td>  
  </tr>  
  <tr>  
    <td>February</td>  
    <td>3124$</td>  
  </tr>  
</table>
```

Monthly savings	
Month	Savings
January	\$100
February	\$50

Figure 24: Recuperado de W3schools (10/3/2020). HTML Tutorial.

### Elementos de bloque y de línea

Todos los elementos HTML tienen un valor de display por defecto dependiendo del tipo de elemento que sea.

La mayoría de los elementos tienen valor de display de bloque (block) o de línea (inline).

### Elementos de Bloque

Los elementos de bloque siempre empiezan en una nueva línea y abarcan todo el ancho disponible, es decir, se estiran de izquierda a derecha.

The <div> element is a block-level element.

Figure 25: W3schools (10/3/2020). HTML Tutorial. Recuperado de:  
<https://www.w3schools.com/html/default.asp>

`<div>Hello</div>`

`<div>World</div>`

Hello  
World

Figure 26: Recuperado de W3schools (10/3/2020). HTML Tutorial.

Los siguientes son otros elementos de bloque:

`<p> <form> <h1>-<h6> <table> <div> <ul> <li>`

### Elementos en Línea

Los elementos en línea no comienzan una nueva línea y sólo ocupan el ancho que se necesita.

This is an inline <span> element inside a paragraph.

Figure 27: Recuperado de W3schools (10/3/2020). HTML Tutorial.

`<span>Hello</span>`

`<span>World</span>`

Hello World

Otros elementos en línea:

`<a> <i> <b> <img> <input> <small> <big> <span> <br> <strong>`

### El elemento <div>

El elemento <div> es comúnmente usado como contenedor para otros elementos HTML.

En conjunto con CSS, el elemento <div> puede ser usado para agregar formato a un bloque de contenido.

`<div style="background-color:black;color:white;padding:20px;">`

`<h2>London</h2>`

**<p>London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>**

**</div>**

El elemento **<span>**

El elemento **<span>** es comúnmente usado como contenedor para texto.

Como con **<div>**, el tag **<span>** usado con CSS puede ser formateado para darle estilo a partes del texto.

**<h1>My <span style="color:red">Important</span> Heading</h1>**

Block, Inline, Inline-block

Como vimos los divs son de tipo block pero se puede cambiar su forma de mostrarse con el atributo display.

Ya vimos los tipos display:block y display:inline. Además existe otro importante, display:inline-block.

Comparado con inline, inline-block permite setear ancho y alto del elemento. Además son respetados los padding y margins superiores e inferiores.

### display: inline

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. Aliquam venenatis gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.

### display: inline-block

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. Aliquam venenatis gravida nisl sit amet

facilisis. Nullam cursus fermentum velit sed laoreet.

### display: block

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat.

Aliquam

venenatis

gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.



Figure 28: Recuperado de W3schools (10/3/2020). HTML Tutorial.

### Layout (diseño)

Las páginas web a menudo muestran el contenido en múltiples columnas (como en una revista o diario).

HTML ofrece elementos semánticos para especificar cada parte.

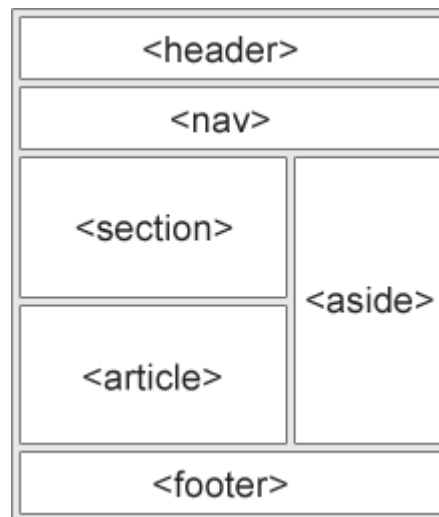


Figure 29: Recuperado de W3schools (10/3/2020). HTML Tutorial.

**<header>**

**<nav>**

**<section>**

**<article>**

**<aside>**

**<footer>**

**<details>** - define detalles adicionales

**<summary>** - define un título para los detalles adicionales.

### Formularios

El elemento <form> define un formulario para recolectar información del usuario.

Un formulario, a su vez, tiene elementos de formulario como inputs, checkboxes, radio buttons, etc.

## Elemento <input>

El elemento <input> es el elemento de formulario más importante.

Puede ser mostrado de diferentes formas, dependiendo del atributo type.

### Text input

`<input type="text">` define un campo de entrada de texto.

`<form>`

First name:<br>

`<input type="text" name="firstname">`<br>

Last name:<br>

`<input type="text" name="lastname">`

`</form>`

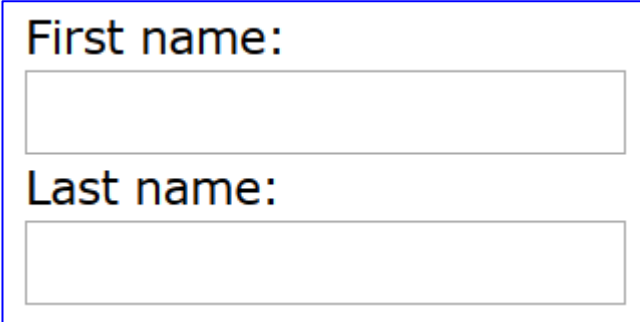


Figure 30: Recuperado de W3schools (10/3/2020). HTML Tutorial.

### Radio button input

`<input type="radio">` define un radio button.

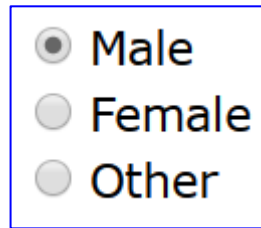
`<form>`

`<input type="radio" name="gender" value="male" checked>` Male<br>

`<input type="radio" name="gender" value="female">` Female<br>

`<input type="radio" name="gender" value="other">` Other

`</form>`



☒ Male  
☐ Female  
☐ Other

Figure 31: Recuperado de W3schools (10/3/2020). HTML Tutorial.

## Tipos de inputs

Estos pueden ser:

`<input type="button">`

`<input type="checkbox">`

`<input type="color">`

`<input type="date">`

`<input type="datetime-local">`

`<input type="email">`

`<input type="file">`

`<input type="hidden">`

`<input type="image">`

`<input type="month">`

`<input type="number">`

`<input type="number">`

`<input type="password">`

`<input type="radio">`

`<input type="range">`

`<input type="reset">`

`<input type="search">`

`<input type="submit">`

`<input type="tel">`

`<input type="text">`

`<input type="time">`

```
<input type="url">
```

```
<input type="week">
```

### Botón Submit

`<input type="submit">` declara un botón para enviar los datos del formulario al servidor.

```
<form action="/action_page.php" method="post">
```

```
First name:<br>
```

```
<input type="text" name="firstname" value="Mickey"><br>
```

```
Last name:<br>
```

```
<input type="text" name="lastname" value="Mouse"><br><br>
```

```
<input type="submit" value="Submit">
```

```
</form>
```

### Agrupando datos del form

La etiqueta `<fieldset>` sirve para agrupar elementos relacionados.

El tag `<legend>` define un epígrafe o título para el elemento `<fieldset>`

```
<form action="/action_page.php">
```

```
<fieldset>
```

```
<legend>Personal information:</legend>
```

```
First name:<br>
```

```
<input type="text" name="firstname" value="Mickey"><br>
```

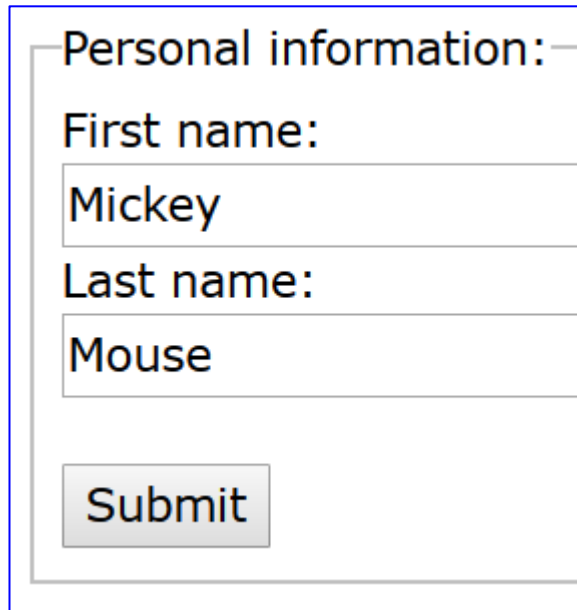
```
Last name:<br>
```

```
<input type="text" name="lastname" value="Mouse"><br><br>
```

```
<input type="submit" value="Submit">
```

```
</fieldset>
```

```
</form>
```



Personal information:

First name:  
Mickey

Last name:  
Mouse

Submit

Figure 32: Recuperado de W3schools (10/3/2020). HTML Tutorial.

### Elemento <select>

El elemento <select> define un combo box.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>The select Element</h2>
```

```
<p>The select element defines a drop-down list:</p>
```

```
<form action="/action_page.php">
```

```
  <select name="cars">
```

```
    <option value="volvo">Volvo</option>
```

```
    <option value="saab">Saab</option>
```

```
    <option value="fiat">Fiat</option>
```

```
    <option value="audi">Audi</option>
```

```
  </select>
```

```
<br><br>
```

```
<input type="submit">  
</form>  
  
</body>  
</html>
```

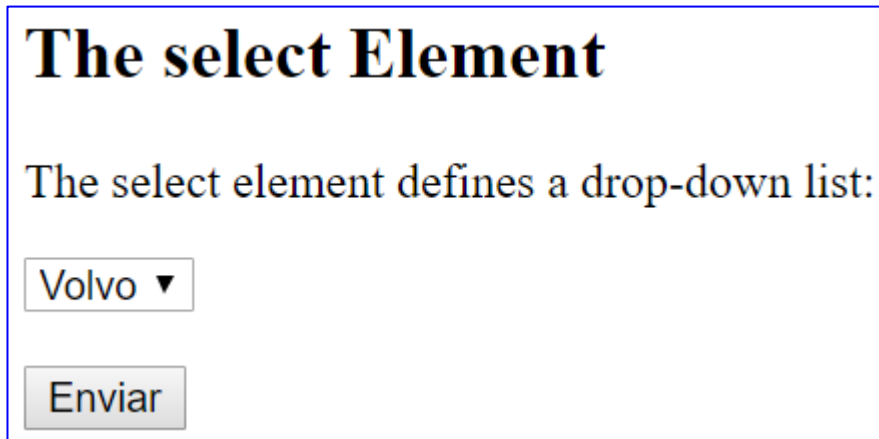


Figure 33: Recuperado de W3schools (10/3/2020). HTML Tutorial.

### Elemento <textarea>

Define un input box de múltiples líneas.

rows especifica la cantidad de líneas visibles y cols el ancho.

```
<textarea name="message" rows="10" cols="30">
```

The cat was playing in the garden.

```
</textarea>
```

El elemento <button> define un botón clickeable.

```
<button type="button" onclick="alert('Hello World!')">Click  
Me!</button>
```

## Introducción a CSS; sus estructuras principales para la manipulación visual del código HTML.

Existen 3 formas de usar estilo.

- Inline usando el atributo style="" en los tags, como venimos haciendo.

``

- Internal usando el elemento `<style>` en la sección del `<head>`.
- External usando un archivo con extensión CSS

**NOTA: ese orden es el orden de precedencia (importancia)**

Se logra usando el tag `<style>` en la sección del `<head>`

La sintaxis corresponde a CSS

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1  {color: blue;}
p   {color: red;}
</style>
</head>
<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph.</p>
</body>
</html>
```

Un archivo externo CSS se utiliza para darle estilo a múltiples páginas.

Además se separa el estilo del contenido.

```
<!DOCTYPE html>
```

```
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

### Sintaxis CSS

La siguiente es la forma en como se escribe CSS.

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

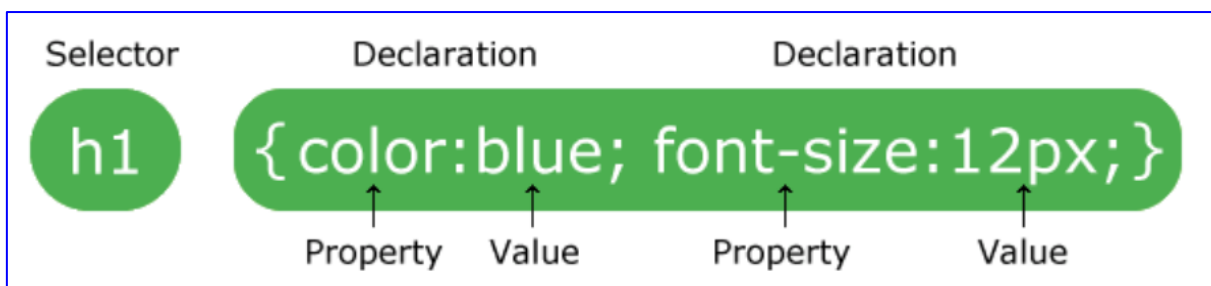


Figure 34: Recuperado de W3schools (10/3/2020). CSS Tutorial. t

### El atributo style

El atributo style es usado para especificar el estilo de un elemento, como el color, fuente, tamaño, etc.



**<p style="color:red;">I am a paragraph</p>**

La sintaxis entonces es

**<tagname style="property:value;">**

En donde property es una propiedad CSS y value es un valor CSS

#### Color de fondo de la página

La propiedad background-color define el color de fondo de algún elemento.

**<body style="background-color:powderblue;">**

**<h1>This is a heading</h1>**

**<p>This is a paragraph.</p>**

**</body>**

#### Color de texto

La propiedad color define el color de texto de algún elemento HTML

**<h1 style="color:blue;">This is a heading</h1>**

**<p style="color:red;">This is a paragraph.</p>**

#### Fuentes

La propiedad font-family define la fuente a ser usada por algún elemento HTML

**<h1 style="font-family:verdana;">This is a heading</h1>**

**<p style="font-family:courier;">This is a paragraph.</p>**

#### Tamaño de fuente

La propiedad font-size define el tamaño del texto de algún elemento de HTML

**<h1 style="font-size:300%;">This is a heading</h1>**

**<p style="font-size:160%;">This is a paragraph.</p>**

### Alineación de texto

La propiedad text-align define la alineación horizontal del texto de algún elemento HTML

```
<h1 style="text-align:center;">Centered Heading</h1>
```

```
<p style="text-align:center;">Centered paragraph.</p>
```

### Colores

Los colores están especificados por nombres predefinidos, RGB, HEX, HSL, RGBA, HSLA.

#### Por nombre

En HTML, un color puede estar especificado por nombre.

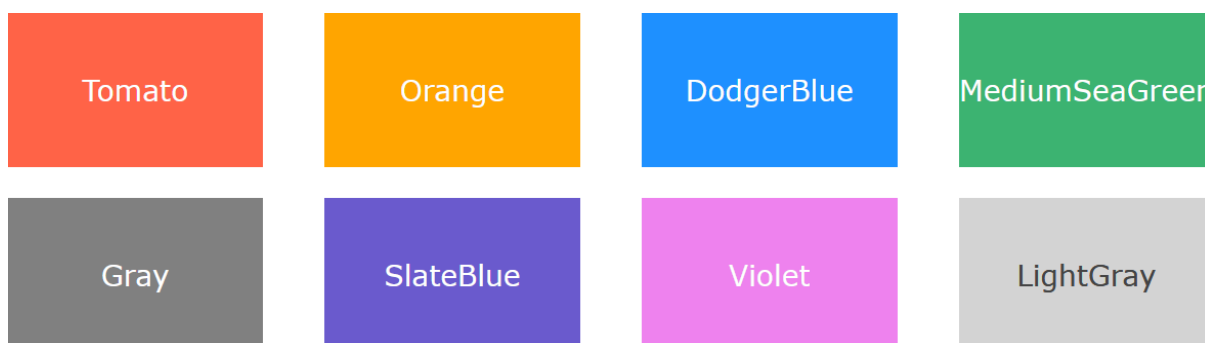


Figure 35: Recuperado de W3schools (10/3/2020). CSS Tutorial.

### Color de fondo

Se puede definir un color de fondo para los elementos de HTML.

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>
```

```
<p style="background-color:Tomato;">Lorem ipsum...</p>
```

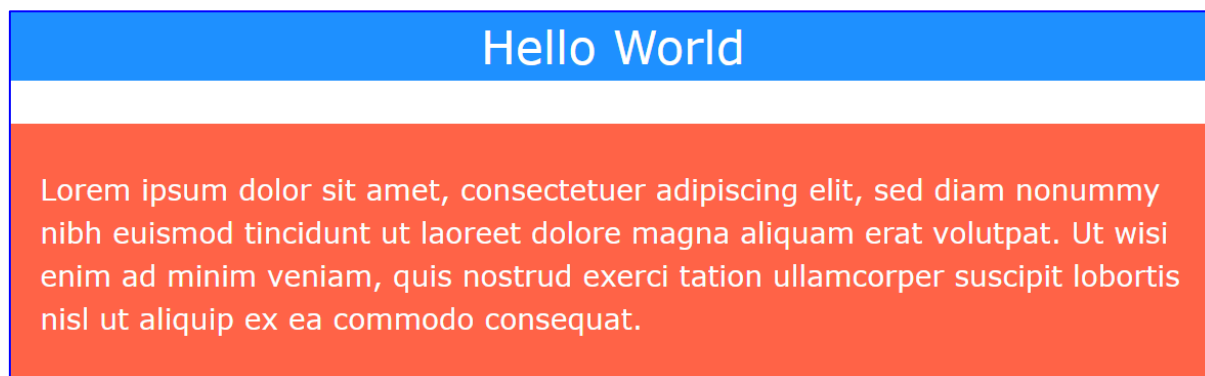


Figure 36: Recuperado de W3schools (10/3/2020). CSS Tutorial.

### Color de texto

Se puede definir un color de texto para los elementos de texto de HTML.

```
<h1 style="color:Tomato;">Hello World</h1>
```

```
<p style="color:DodgerBlue;">Lorem ipsum...</p>
```

```
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Figure 37: Recuperado de W3schools (10/3/2020). CSS Tutorial.

### Color de borde

Se puede definir un color de borde según el borde del box model:

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
```

```
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
```

```
<h1 style="border:2px solid Violet;">Hello World</h1>
```

Hello World

Hello World

Hello World

Figure 38: Recuperado de W3schools (10/3/2020). CSS Tutorial.

### Color por valor

Los colores se pueden definir también por valores RGB, HEX, HSL, RGBA y HSLA.

```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
```

```
<h1 style="background-color:#ff6347;">...</h1>
```

```
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>
```

```
<h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1>
```

```
<h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>
```

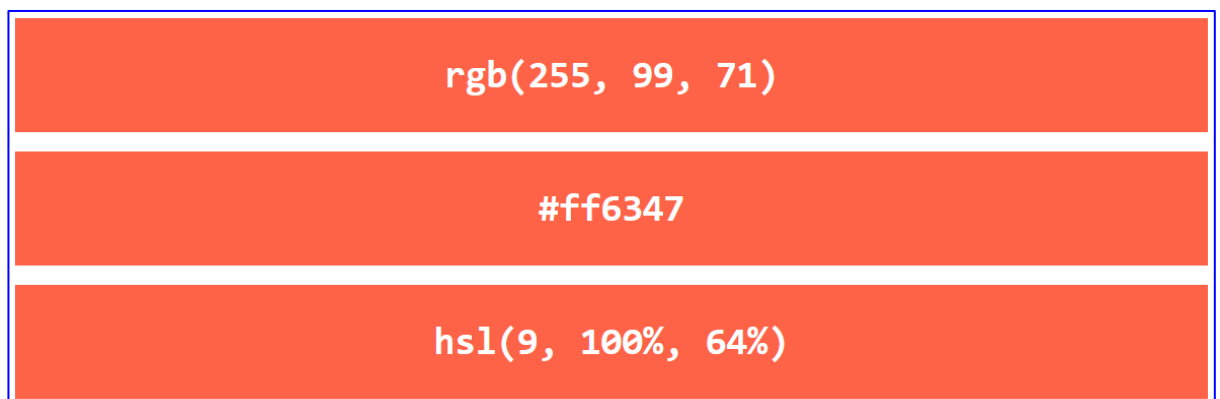


Figure 39: Recuperado de W3schools (10/3/2020). CSS Tutorial.

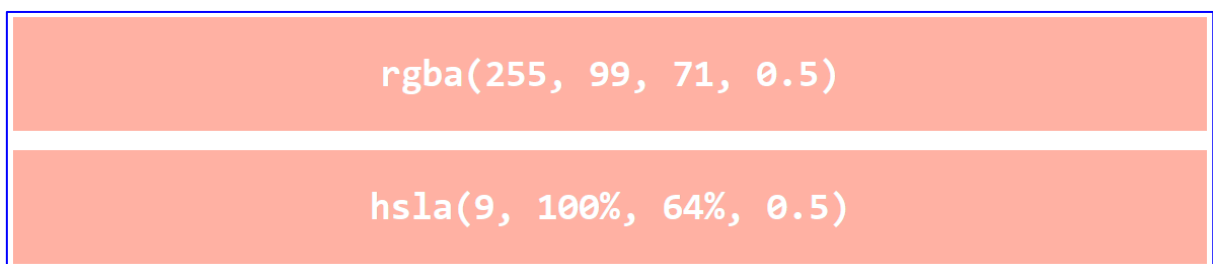


Figure 40: Recuperado de W3schools (10/3/2020). CSS Tutorial.

### Image floating

Hay una propiedad float que permite que la imagen "flote" a la derecha o a la izquierda del texto.

```
<<p>
```

The image will float to the right of the text.</p>

```
<p>
```

The image will float to the left of the text.</p>

### Float the image to the right:

A paragraph with a floating image. A paragraph with a floating image. A paragraph with a floating image.



### Float the image to the left:



A paragraph with a floating image. A paragraph with a floating image. A paragraph with a floating image.

Figure 41: Recuperado de W3schools (10/3/2020). CSS Tutorial.

### Imagen de fondo

Para agregar una imagen de fondo en un elemento HTML, use la propiedad background-image.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body style="background-image:url('clouds.jpg');">
```

```
<h2>Background Image</h2>
```

<p>By default the background image will repeat itself if it is smaller than the element where it is specified, in this case the BODY element.</p>

```
</body>
```

```
</html>
```

### CSS padding

El padding es el espacio que hay entre el texto y el borde

```
p {
```

```
border: 1px solid powderblue;
```

```
padding: 30px;
```

```
}
```

### CSS margin

El margin es el espacio que hay por fuera del borde.

```
p {
```

```
border: 1px solid powderblue;
```

```
margin: 50px;
```

```
}
```

### Box Model

A todos los elementos se los considera como cajas

El modelo de cajas es esencialmente una caja que rodea cada elemento

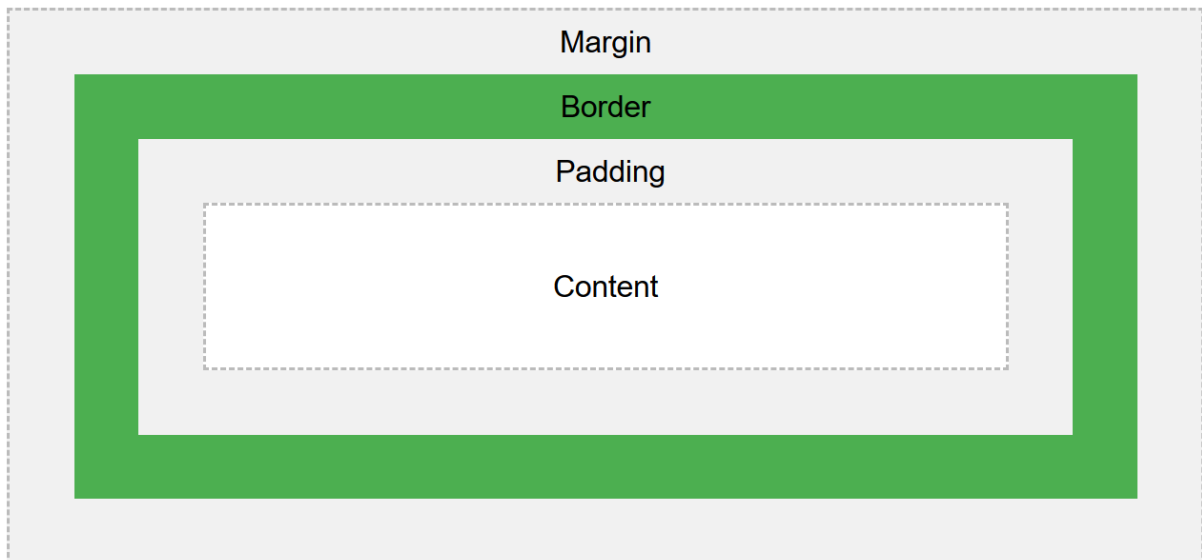


Figure 42: Recuperado de W3schools (10/3/2020). CSS Tutorial.

### Atributo Id

Identifica un único elemento para aplicarle estilo.

```
<p id="p01">I am different</p>
```

Se aplica el estilo usando el id como selector

```
#p01 {
```

```
color: blue;
```

```
}
```

### Atributo class

Identifica una clase de elementos para aplicarle estilo a todos los de la misma clase.

```
<p class="error">I am different</p>
```

Se aplica el estilo usando la clase como selector

```
p.error {  
  color: red;  
}
```

### Ejemplo de estilo aplicado a clases

```
<!DOCTYPE html>  
<html>  
  <head>  
    <style>  
      .cities {  
        background-color: black;  
        color: white;  
        margin: 20px;  
        padding: 20px;  
      }  
    </style>  
  </head>  
  <body>  
  
    <div class="cities">  
      <h2>London</h2>  
      <p>London is the capital of England.</p>  
    </div>
```

```
<div class="cities">  
  <h2>Paris</h2>  
  <p>Paris is the capital of France.</p>  
</div>  
  
</body>  
</html>
```

### Múltiples clases

Los elementos pueden tener más de una clase.

Para ello se especifican las clases separadas por espacio.

```
<h2 class="city main">London</h2>  
<h2 class="city">Paris</h2>  
<h2 class="city">Tokyo</h2>
```



## Conclusión Unidad I

Queda a criterio del estudiante continuar con el estudio de CSS viendo temas como POSITION, Flexbox y Grid para poder realizar maquetado más complejo.

Para Grid se recomienda la lectura de A Complete Guide to Grid de \*CSS-TRICKS como así también sobre Flexbox en A Complete Guide to Flexbox en el mismo recurso.

Si bien este es un curso introductorio al modelado web debido a la restricción de tiempos de cursado, se recomienda continuar con los temas mencionados anteriormente. Sin embargo, con lo visto hasta ahora se podrán realizar trabajos interesantes.

## Bibliografía

Carmen Lázaro Guillamón (2009) Cooperación al desarrollo y alternativas frente a la crisis. Universitat Jaume.

CSS-TRICKS (10/3/2020). A Complete Guide to Grid. Recuperado de: <https://css-tricks.com/snippets/css/complete-guide-grid/>

CSS-TRICKS (10/3/2020). A Complete Guide to Flexbox. Recuperado de: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Escuela de suboficiales, Armada Argentina (2018) *Informática III*. Recuperado de: <http://www.essa.ara.mil.ar/cens/MATERIAS%20TERCER%20A%C3%91O/3%C2%B0%20A%C3%91O/02-INFORMATICA%20III/SEGUNDO%20CUATRIMESTRE/SEGUNDO%20CUATRIMESTRE.pdf>

Farías Valeria (2013). *Trabajo Práctico Redes; Redes, Internet y Educación*. Recuperado de: <https://es.calameo.com/books/0033959084b58957b9c68>

Medium (10/3/2020). Red de computadoras. Recuperado de: <https://medium.com/@KVTacoa/red-de-computadoras-e3941854e4d3>

Oracle (10/3/2020) Parte II Administración de TCP/IP. Recuperado de: <https://docs.oracle.com/cd/E19957-01/820-2981/6nei0r0sr/index.html>

Tanenbaum, A. S., & Tanenbaum, A. S. (2003). *Computer networks, fourth edition: Problem solutions*. Prentice Hall PTR.

Tecnologías y protocolos de red (10/3/2020). Recuperado de: [https://www.dsi.uclm.es/personal/miguelfgraciani/mikicurri/Docencia/LenguajesInternet0910/web\\_LI/Teoria/Protocolos%20de%20bajo%20nivel/Protocolo%20TCP.htm](https://www.dsi.uclm.es/personal/miguelfgraciani/mikicurri/Docencia/LenguajesInternet0910/web_LI/Teoria/Protocolos%20de%20bajo%20nivel/Protocolo%20TCP.htm)

W3schools (10/3/2020). CSS Tutorial. Recuperado de: <https://www.w3schools.com/css/default.asp>

W3schools (10/3/2020). HTML Tutorial. Recuperado de: <https://www.w3schools.com/html/default.asp>



**Atribución-NoComercial-SinDerivadas**

Se permite descargar esta obra y compartirla, siempre y cuando no sea modificado y/o alterarse su contenido, ni se comercializarse. Referenciarlo de la siguiente manera:

Universidad Tecnológica Nacional Regional Córdoba (2020). Material para la Tecnicatura en Programación Semipresencial de C. Argentina.