

# **Diseño de Páginas Web**

# **HTML 5**

**Por: Ing. Edwin Calle Terrazas**

---

El HTML5 (HyperText Markup Language, versión 5) es la quinta revisión del lenguaje de programación “básico” de la World Wide Web, el HTML.

Esta nueva versión pretende remplazar al actual (X)HTML, corrigiendo problemas con los que los desarrolladores web se encuentran, así como rediseñar el código actualizándolo a nuevas necesidades que demanda la web de hoy en día.

---



Ultimadamente el elemento **<div>** comenzó a dominar la escena. Con el surgimiento de webs más interactivas y la integración de HTML, CSS y Javascript, el uso de **<div>** se volvió una práctica común en la ubicación de los elementos en la estructura de una página web. Pero este elemento, así como **<table>** no provee demasiada información acerca de las partes del cuerpo que está representando, las imágenes a menús, textos, enlaces, códigos, formularios u otra cualquier cosa puede ir entre las etiquetas de apertura y cierre de un elemento **<div>**. En otras palabras, la palabra clave **div** solo especifica divisiones en el cuerpo de una página, así como las celdas de una tabla, pero no ofrece indicio alguno sobre qué **clase de división** es, cuál es su propósito en la estructura de una página o qué contiene.

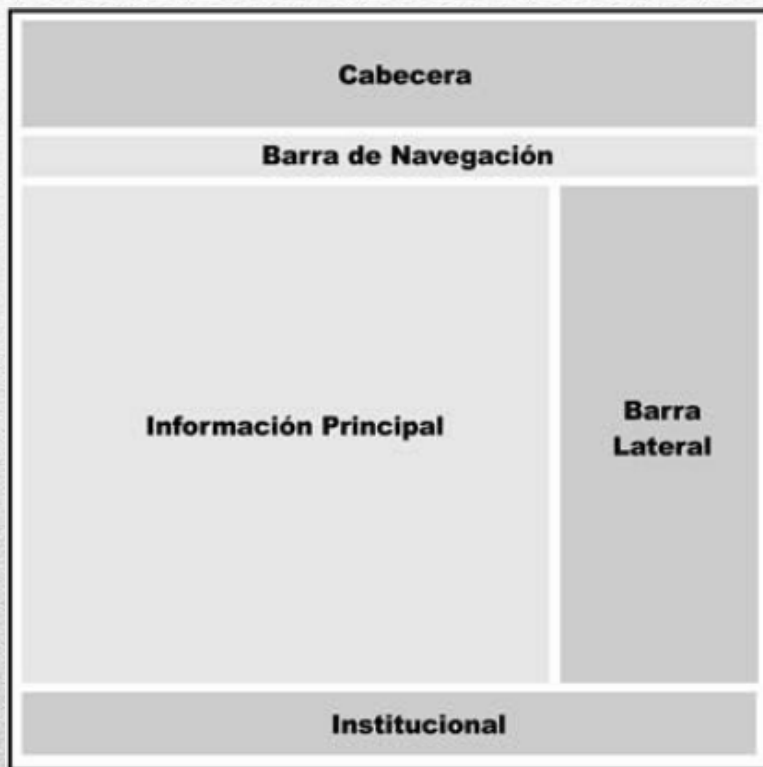
HTML5 incorpora nuevos elementos que ayudan a identificar cada sección del documento y organizar el cuerpo de una página. Las secciones más importantes son diferenciadas y la estructura principal ya no depende más de los elementos **<div>** o **<table>**

---



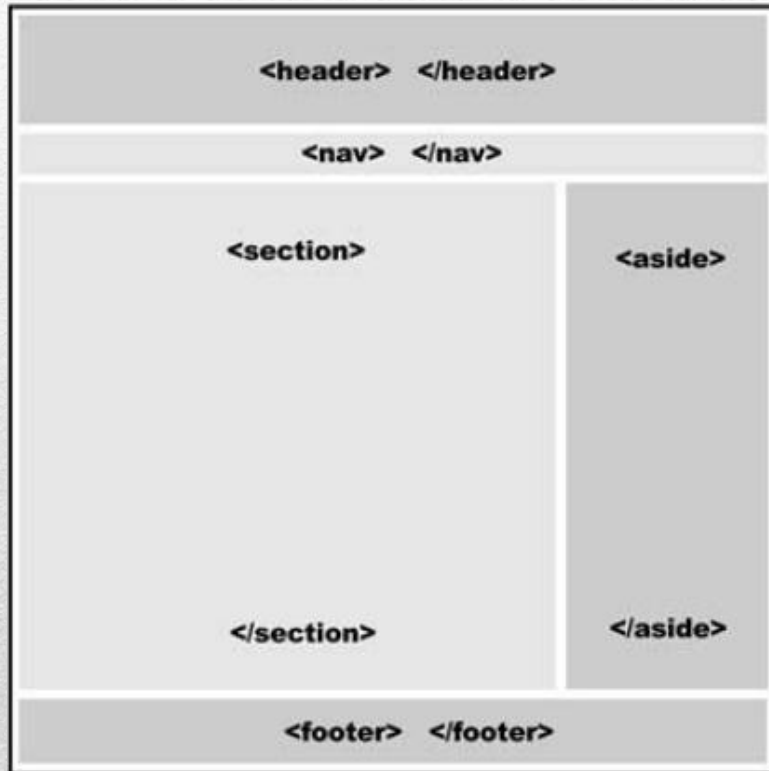
## *Representación visual de la estructura de un clásico diseño web*

Normalmente una página o aplicación web está dividida entre varias áreas visuales para mejorar la experiencia del usuario y facilitar la interactividad. Las palabras claves que representan cada nuevo elemento de HTML5 están íntimamente relacionadas con estas áreas



La **Cabecera**, se encuentra el espacio donde usualmente se ubica el logo, título, subtítulos y una corta descripción del sitio web o la página. La **Barra de Navegación** donde casi todos los desarrolladores ofrecen un menú o lista de enlaces con el propósito de facilitar la navegación a través del sitio. El contenido más importante de una página web se encuentra ubicado en el centro. La mayoría de las veces es dividida en varias filas y columnas. En este ejemplo se utilizan solo dos columnas: **Información Principal** y **Barra Lateral**, pero esta sección es flexible y normalmente diseñadores la adaptan acorde a sus necesidades insertando más columnas, o dividiendo cada columna entre bloques más pequeños. El contenido presentado en esta parte del diseño es usualmente de alta prioridad. En **Información Principal** podría contener una lista de artículos, descripción de productos, entradas de un blog o cualquier otra información importante. La **Barra Lateral** podría mostrar una lista de enlaces apuntando hacia cada algunos ítems. En un blog, por ejemplo, esta última columna ofrecerá una lista de enlaces apuntando a cada entrada del blog, información acerca del autor u otros, etc...

A partir de ahora podemos decir al navegador para qué es cada sección. HTML5 considera esta estructura básica y provee nuevos elementos para diferenciar y declarar cada una de sus partes.



1. Cabecera
2. Barra de Navegación
3. Sección de Información Principal
4. Barra Lateral
5. El pie o la barra Institucional



# Nuevos elementos

## **<header> </header>**

No debe ser confundido con <head> usado para construir la cabecera del documento. Del mismo modo que <head>, la intención de **<header>** es proveer información introductoria (títulos, subtítulos, logos), pero difiere con respecto a <head> en su alcance. Mientras que el elemento <head> tiene el propósito de proveer información acerca de todo el documento, **<header>** es usado solo para el cuerpo o secciones específicas dentro del cuerpo

## **<nav> </nav>**

El elemento **<nav>** fue creado para ofrecer ayuda para la navegación, como en menús principales o grandes bloques de enlaces. **<nav>** se encuentra dentro de las etiquetas **<body>** pero es ubicado después de la etiqueta de cierre de la cabecera (**</header>**), no dentro de las etiquetas **<header>**. Esto es porque **<nav>** no es parte de la cabecera sino una nueva sección. Un ejemplo de esta versatilidad es que el elemento **<nav>** podría ser insertado dentro del elemento **<header>** o en cualquier otra parte del cuerpo. Sin embargo, siempre se debe considerar que estas etiquetas fueron creadas para brindar información a los navegadores.

## **<section> </section>**

Como explicamos anteriormente, la columna **Información Principal** contiene la información más relevante de la página. El elemento en html5 que especifica estas secciones se llama simplemente **<section>**

---



**<aside>   </aside>**

la **Barra Lateral** contendrá una lista de enlaces. La información dentro de esta barra está relacionada con la información principal pero no es relevante por sí misma. Siguiendo el mismo ejemplo podemos decir que las entradas del blog son relevantes pero los enlaces y las pequeñas reseñas sobre esas entradas son solo una ayuda para la navegación pero no lo que al lector realmente le interesa. En html5 podemos diferenciar sección secundaria de información utilizando el elemento **<aside>**

**<footer>   </footer>**

Ya contamos con la cabecera del cuerpo, secciones con ayuda para la navegación, información importante y hasta una barra lateral con datos adicionales, por lo tanto lo único que nos queda por hacer es cerrar nuestro diseño para otorgarle un final al cuerpo del documento. Html5 provee un elemento específico para este propósito llamado **<footer>**

El elemento **<footer>** representará el final del cuerpo de nuestro documento. Sin embargo, **<footer>** puede ser usado múltiples veces dentro del cuerpo para representar también el final de diferentes secciones (del mismo modo que la etiqueta **<header>**). Estudiaremos esta última característica más adelante.

---

```
<html>
<head> <title>Título del documento</title> </head>
<body>
<header> <h1>TITULO DE LA PAGINA</h1> </header>
<nav>
<table border="2">
<tr><td> <a href="principal.html"> Principal </a> </td>
      <td> <a href="fotos.html">Fotos</a> </td>
      <td> <a href="videos.html">Videos </a> </td> </tr>
</table>
</nav>
<section>
Esta es la sección de información principal
</section>
<aside>
<blockquote>informacion 1</blockquote>
<blockquote>informacion 2</blockquote>
<a href="enlace.html"> Es un enlace </a>
</aside>
<footer> Derechos Reservados Copy 2013-2014 </footer>
</body>
</html>
```

# TITULO DE LA PAGINA

[Principal](#) [Fotos](#) [Videos](#)

Esta es la sección de informacion principal

informacion 1

informacion 2

[Es un enlace](#)

Derechos Reservados Copy 2013-2014



# Mejoras en los formularios

El elemento input adquiere gran relevancia al ampliarse los elementos que se permiten en el “type”.

`<input type="search">` cajas de búsqueda.

`<input type="number">` para adicionar o restar números mediante botones

`<input type="range">` selecciona un valor entre dos valores predeterminados

`<input type="color">` seleccionar un color

`<input type="tel">` números telefónicos

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

`<input type="email">` direcciones de email

`<input type="date">` seleccionar un día en un calendario

`<input type="month">` seleccionar un mes en un calendario

`<input type="week">` para semanas.

`<input type="time">` para fechas

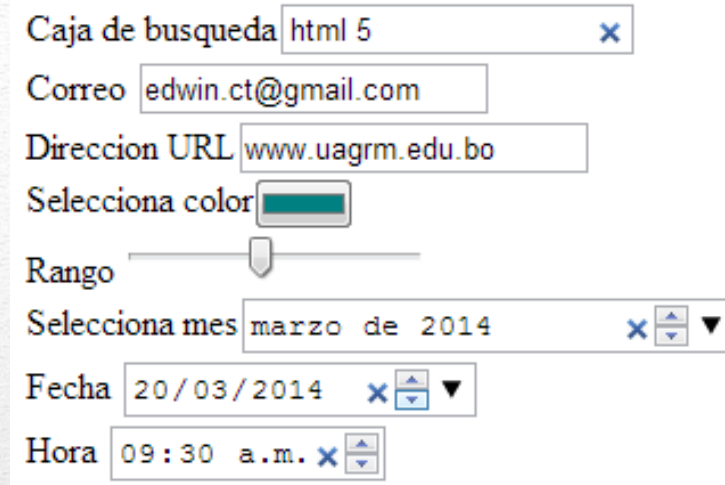
`<input type="datetime">` para una fecha exacta, absoluta y tiempo.

`<input type="datetime-local">` para fechas locales y frecuencia.

---

# Ejemplo

```
<html>
<head> </head>
<body>
<form name="form1" method="post">
Caja de busqueda <input type="search"> <br>
Correo <input type="email"> <br>
Direccion URL<input type="url"> <br>
Selecciona color <input type="color"> <br>
Rango <input type="range"> <br>
Selecciona mes <input type="month"> <br>
Fecha <input type="date" name="d1"> <br>
Hora <input type="time"> <br>
</form>
</body>
</html>
```



A screenshot of a web browser displaying a form with the following elements:

- Caja de busqueda**: A search input field containing the text "html 5".
- Correo**: An email input field containing "edwin.ct@gmail.com".
- Direccion URL**: A URL input field containing "www.uagrm.edu.bo".
- Selecciona color**: A color selection input field showing a teal color.
- Rango**: A range input field represented by a horizontal slider.
- Selecciona mes**: A month selection input field showing "marzo de 2014".
- Fecha**: A date selection input field showing "20/03/2014".
- Hora**: A time selection input field showing "09:30 a.m.". Each of these last three fields includes a small 'x' icon for clearing the input.



# La declaración DOCTYPE de HTML5

Un cambio notable en la especificación de HTML5 ya se puede ver en la primera línea de cualquier documento HTML. La línea inicial de la mayoría de los archivos HTML es la que se llama DOCTYPE, que le indica al navegador el tipo de documento que tiene que traducir. Ejemplo, cuando un diseñador utiliza la versión transicional de HTML (4.0.1), la declaración DOCTYPE es:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

Si utiliza la versión estricta de XHTML 1.0, la misma declaración se queda en:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

En HTML5 la declaración DOCTYPE es bastante más sencilla:

```
<!DOCTYPE html>
```

DOCTYPE sirve para validar documentos. Los navegadores actuales y futuros pueden comprobar la sintaxis de documentos que se declaran a sí mismos como HTML. Cualquier incorrección sintáctica (por ejemplo una etiqueta mal escrita o un atributo que falta), devuelve un error. La validación es un proceso opcional, pero los diseñadores y los desarrolladores lo utilizan mucho para detectar posibles fallos en el código o elementos que puedan faltar.

---



## Los elementos <figure> y <figcaption> de HTML5

Puedes encontrar otros elementos nuevos que sirven para que la estructura interna de las páginas se entienda mejor. Por ejemplo, los elementos <figure> y <figcaption> nos permiten identificar imágenes y sus rótulos dentro del contenido.

```
<html>
<head> <title>Título del documento</title> </head>
<body>
<figure></figure>
</body> </html>
```

Este elemento está pensado para identificar o destacar ciertas partes de la página, como imágenes, diagramas o gráficos. Podemos además emparejar la figura con un rótulo añadiendo un elemento llamado <figcaption>.

```
<html>
<head> <title>Título del documento</title> </head>
<body>
<figure>

<figcaption> Este es mi rotulo</figcaption>
</figure>
</body> </html>
```

---



```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<header>
<video controls="" poster="imagen.jpg">
<source src="recursos/proxy.mp4" type="video/mp4">
<source src="recursos/proxy.webm" type="video/webm">
<source src="recursos/proxy.ogg" type="video/ogg">
</video>
</header>

</body>
</html>
```

---

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body>
<audio>
<video controls="" autoplay="" style="width:500px"/>
<source src="recursos/camino.mp3" type="audio/mp3"/>
<source src="recursos/camino.m4a" type="audio/m4a"/>
<source src="recursos/camino.ogg" type="audio/ogg"/>
</audio>
</header>

</body>
</html>
```

---



# Dibujo de un cuadro con relleno y color

```
<html>
<head> <title>Título del documento</title>

<script type="text/javascript">
function dibujar() {
var lienzo = document.getElementById('lienzo');
var lienzo1 = lienzo.getContext('2d');
lienzo1.fillStyle="rgb(0,0,200)";
lienzo1.fillRect(45,35,45,100);
}
</script>
</head>

<body onload="dibujar()">
<h1> Ejemplo de lienzo </h1>
<canvas id="lienzo" width="600" height="500">
</canvas>

</body>
</html>
```

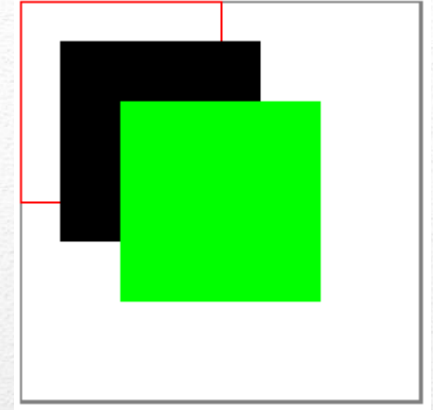
## Ejemplo de lienzo



## Dibujo de cuadros con relleno y color

```
<html>
<head> <title>Título del documento</title>
<script type="text/javascript">
function setup() {
  var canvas = document.getElementById('lessonCanvas');
  if (canvas.getContext) {
    var c1 = canvas.getContext('2d');
    c1.strokeRect(0, 0, 200, 200); //construimos un cuadrado sin relleno
    c1.strokeStyle = "rgb(255, 0, 0)"; //construimos un cuadrado color rojo mas pequeño
    c1.strokeRect(0.5, 0.5, 100, 100);
    //construimos un cuadrado que se posiciona en x=20, y=20 del mismo tamaño q el anterior
    c1.fillRect(20, 20, 100, 100);
    //construimos un cuadrado q se posiciona en x=50 y=50 del tamaño anterior de color verde
    c1.fillStyle = "rgb(0, 255, 0)";
    c1.fillRect(50, 50, 100, 100);
  }
}
</script> </head>
<body onload="setup()">
<canvas id="lessonCanvas" width="300"height="300" style="margin:100px;"></canvas>

</body>
</html>
```





## Dibujo de un cuadro a través de líneas

```
<html>
<head> <title>Título del documento</title>
<script type="text/javascript">
function setup() {
  var canvas = document.getElementById('lessonCanvas');
  if (canvas.getContext) {
    var c1 = canvas.getContext('2d');
    c1.moveTo(0, 0);
    c1.lineTo(200, 0);
    c1.lineTo(200, 200);
    c1.lineTo(0, 200);
    c1.lineTo(0, 0);
    c1.stroke();
  }
}
</script>
</head>
<body onload="setup()">
<canvas id="lessonCanvas" width="300"height="300" style="margin:100px;"></canvas>
</body> </html>
```

