



FIME

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

# Modulo III

## HTML<sub>5</sub> & CSS<sub>3</sub>



Instructor:

**M.S.I. Mario Alejandro Rocha Moreno**

**Correo electrónico: [thealezroc@gmail.com](mailto:thealezroc@gmail.com)**

**COORDINACION DE EDUCACIÓN CONTINUA**

# PONDERACIÓN

Prácticas 40%

Tareas 10%

Proyecto 20%

Examen 20%

Asistencia 10%

**-Más de una falta equivale a repetir modulo**

**-Dos retardos es igual a una falta**

## OBJETIVO GENERAL

### UNIDAD 1.- INTRODUCCIÓN BASES DE HTML

1.1 Roles de HTML y CSS en el desarrollo web.

1.1.1 ¿Qué es HTML?

1.1.2 ¿Qué es CSS?

1.1.3 Los Roles de las Tecnologías Web,

1.1.4 ¿Qué se puede hacer con HTML & CSS?

1.2 ¿Qué es HTML5?

1.3 Git Básico

1.3.1 ¿Qué es git?

1.3.2 ¿Por qué aprender Git?

1.3.3 Plataformas

1.3.4 Comandos Básicos

1.4 HTML Bases.

1.4.1 Introducción

1.4.1.1 Crear y abrir página HTML.

1.4.1.2 Doctype y Estructura Básica

1.4.2 Meta Tags y Motor de Búsqueda.

1.4.2.1 Charset.

1.4.2.2 Viewport.

1.4.2.3 Description.

1.4.2.4 Keywords.

1.4.2.5 Robots.

1.4.2.6 Author.



### 1.4.3 Componentes Básicos.

1.4.3.1 Cabecera, Párrafos y Tipografía

1.4.3.2 Enlaces, Imágenes y Atributos.

1.4.3.3 Listas y Tablas.

1.4.3.4 Formularios y Campos de entrada.

1.4.3.5 Niveles de Elementos en Bloque o en una Línea.

1.4.3.6 Divs & Spans, Clases y Ids

1.4.3.7 Entidades HTML.

1.4.3.8 Etiquetas HTML5.

## **OBJETIVO GENERAL:**

El alumno aprenderá los conceptos y herramientas necesarias para crear páginas web con HTML con CSS, además de poder hacer deploy en un servicio de hosting, haciendo pública su página web.

# UNIDAD 1.- INTRODUCCIÓN BASES DE HTML

## 1.1 Roles de HTML y CSS en el desarrollo web.

HTML fue desarrollado con la intención de simplificar, especificar y organizar el código. Para lograr este propósito, nuevas etiquetas y atributos fueron agregados y HTML fue completamente integrado a CSS. Estas incorporaciones y mejoras de versiones previas están relacionadas no solo con nuevos elementos sino también con cómo usamos los ya existentes.

### 1.1.1 ¿Qué es HTML?

Es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web, sus siglas HTML (Hypertext Markup Language) Lenguaje de Marcado para Hipertexto

### 1.1.2 ¿Qué es CSS?

Es un lenguaje para especificar cómo los documentos se presentan a los usuarios, sus siglas Cascading Style Sheets (CSS), traducido literalmente al español, como Hojas de estilo en cascada.

Es utilizado para darle estilo a los elementos y contenidos, así como también puedes tener variables, puedes realizar cálculos.

### 1.1.3 Los Roles de las Tecnologías Web,

**HTML** = Mostrar Contenido/ Estructura.

**CSS** = Estilo y Diseño.

**Javascript** = Programación Dinámica de Frontend (es la parte de un sitio web que interactúa con los usuarios).

**PHP, Python, C++, C#, JS** = Programación Backend (es la parte que se conecta con la base de datos y el servidor que utiliza dicho sitio web).

**Mysql, MongoDB** = Base de dato.

### 1.1.4 ¿Qué se puede hacer con HTML & CSS?

*Crear sitios web estáticos:* Portafolios, sitio de negocios, sitios informativos, etc.

*Crear la parte visual de cualquier cosa:* Tiendas, Red Social, etc.

## 1.2 ¿Qué es HTML5?

Es la última versión de HTML. El término representa dos conceptos diferentes:

Se trata de una nueva versión de HTML, con nuevos elementos, atributos y comportamientos.

Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance.

Esta nueva versión, en conjunto con CSS3, define los nuevos estándares de desarrollo web, rediseñando el código para resolver problemas y actualizándose así a nuevas necesidades. No se limita solo a crear nuevas etiquetas o atributos, si no que incorpora muchas características nuevas y proporciona una plataforma de desarrollo de complejas aplicaciones web mediante APIs.



## 1.3 Git Básico.

### 1.3.1 ¿Qué es git?

Es un software de control de versiones diseñado por Linus Torvalds.

Donde el Control de versiones es la gestión de cambios hechos sobre algún producto, por ejemplo, lo que se realiza cuando haces cambios en el código de tu sitio web y subes a la nube, lo que implica un cambio de versión, git te ayuda a facilitar estas tareas, así como el trabajar en equipo sobre un producto.

### 1.3.2 ¿Por qué aprender Git?

Si quieres dedicarte al **desarrollo de software** Git es una de las herramientas más usadas a nivel mundial, un buen uso de ella te permitirá trabajar de forma profesional. Si trabajas solo, usar Git es útil, pero si lo haces en **equipo es fundamental**.

#### Companies & Projects Using Git

Google

facebook

Microsoft

twitter

Linkedin.

NETFLIX



### 1.3.3 Plataformas.

Existen múltiples plataformas donde pueden utilizar git de manera gratuita, por ejemplo: Github, Gitlab, Bitbucket, etc.

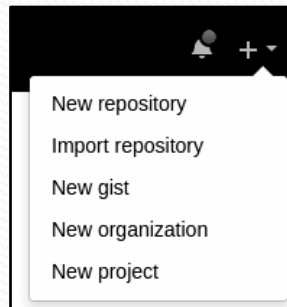
#### Práctica

Trabajaremos en Github así que la práctica consiste en:

1. Entrar a <https://github.com/>
2. Registrarse.
3. Enviar al grupo tu perfil. (Asegúrate que sea el correcto, abre el enlace como escribiste tu usuario y debe abrir la vista de tu perfil de Github.).

Ejemplo: <https://github.com/tuusuarioaqui>

4. Crear tu repositorio con el nombre html-css, el repositorio debe ser privado



## 5. Por último, agregar a tu maestro en turno como colaborador.

<> Code

! Issues 0

🔗 Pull requests 0

📁 Projects 0

📊 Insights

⚙️ Settings

Options

**Collaborators**

Webhooks

Notifications

Integrations & services

Deploy keys

Collaborators

Push access to the repository

This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

**Search by username, full name or email address**

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

Add collaborator

0 of 3 collaborators

### 1.3.4 Comandos Básicos.

#### **Tu identidad.**

Lo primero que deberías hacer cuando instalas Git es establecer tu nombre de usuario y dirección de correo electrónico. Esto es importante porque las confirmaciones de cambios (commits) en Git usan esta información, y es introducida de manera inmutable en los commits que envías:

```
$ git config --global user.name "Tu Nombre Aquí"
$ git config --global user.email fime@example.com
```

#### **Iniciar Repositorio.**

Para iniciar un repositorio sobre cualquier carpeta donde te encuentres es utilizando el comando siguiente:

```
git init
```

Así como también se le señala a qué repositorio será dirigido:

```
git remote add origin https://github.com/fime/ejemplo.git
```

La otra alternativa es crear una copia de un Repositorio utilizando el siguiente comando:

```
git clone https://github.com/fime/ejemplo.git
```

## Cambios en el Repositorio.

Al momento de realizar cambios en un repositorio se pueden realizar diferentes comandos para subir cambios y para manejar los mismos.

El siguiente comando te sirve para poder ver que archivos hay nuevos, editados o borrados en el repositorio en base al último cambio en el mismo:

git status

Ejemplo del output:

On branch master  
Your branch is up to date with 'origin/master'.

Changes not staged for commit:  
(use "git add/rm <file>..." to update what will be committed)  
(use "git checkout -- <file>..." to discard changes in working directory)

modified: clase/index.html  
deleted: screenshots\_manual/tag\_syntax.html

Untracked files:  
(use "git add <file>..." to include in what will be committed)

clase/fime.html

Ahora se agregan los cambios por archivos a staging con el comando:

```
git add clase/index.html
```

Dependiendo del cambio realizado y el archivo agregado se le pone un comentario que describe el cambio al agregar el comentario pasa a formar parte permanente en el repositorio este se mostrará en la historia del mismo:

```
git commit -m "Cambio de título en el index de la clase"
```

Con el comando **git log** puedes ver el historial de tu repositorio:

```
git log
```

commit	f1ca7f97686d01d4fd187a362fb072b346c1f1d4	(HEAD	->	master,	origin/master)
Author:	Nombre	del	Autor	<correo@gmail.com>	
Date:	Sun	Feb	10	21:51:44	2019 -0600

Avance de clases fime

Por último, para ver reflejados los cambios en la plataforma (Github, Gitlab o Bitbucket) se suben los cambios con el siguiente comando:

```
git push -u origin master
```

## Actualizar el Repositorio.

Cuando se trabaja en Git comúnmente se presentan diferentes factores que implican actualizar tu repositorio localmente como los casos de trabajo en equipo, un compañero hizo un cambio y ahora tú quieres ver ese cambio en tu repositorio localmente, o cuando en clase realizamos cambios los subes y ahora quieres continuar trabajando en otra computadora para esos casos se utiliza el siguiente comando:

```
git pull
```

## 1.4 HTML Bases.

### 1.4.1 Introducción.

#### 1.4.1.1 Crear y abrir página HTML.

##### Sintaxis de las Etiquetas:

1. Los elementos están rodeados por símbolos de mayor qué y menor qué (<elemento>).
2. Usualmente tienen un inicio y un fin de etiqueta.
3. Algunas etiquetas no necesitan de etiqueta de cierre.

#### **4. PRACTICA 1 (ATOM) :**

Ahora crearemos una página HTML y la abriremos en el explorador.

Abrimos nuestro editor de texto (Atom) y creamos un archivo con el nombre index.html

Y colocamos lo siguiente:



### 1.4.1.2 Doctype y Estructura Básica.

1. **<!DOCTYPE html>**: es con lo que debe iniciar cualquier archivo de html para que el explorador lo identifique como un archivo de **HTML5**.
2. **<html>**: es el elemento raíz de una página HTML.
3. **<head>**: es el que contiene la metainformación de un documento.
4. **<title>**: es el que especifica el título de un documento, este se mostrará en el tab de tu explorador.
5. **</head>**: cierre de etiqueta.
6. **<body>**: este elemento contiene todo lo que será visible en la página.
7. **<h1>**: elemento que muestra un título con letras grandes, o cabecera.
8. **<p>**: es un elemento de párrafo

#### Práctica 2

### 1.4.2 Meta Tags y Motor de Búsqueda

#### Práctica 3

#### 1.4.2.1 Charset.

Es la que permite definir qué codificación de caracteres utilizará la página, la codificación de caracteres es el método que permite convertir un carácter de un lenguaje natural en un símbolo de otro sistema de representación, como un número o una secuencia de pulsos eléctricos en un sistema electrónico, aplicando normas o reglas de codificación.

### 1.4.2.2 Viewport.

Esta fue introducida en **HTML5** y ayuda a tomar el control de cómo el usuario verá la página dependiendo de los dispositivos en los cuales se despliegue el contenido.

“width=device-width”: determina que se siga el tamaño de la pantalla del dispositivo esto varía dependiendo de cada dispositivo.

“initial-scale=1.0”: determina el zoom inicial del explorador.

### 1.4.2.3 Description.

Esto es la descripción del sitio el cual será mostrado al realizar una búsqueda de tu sitio.

Ejemplo:

**Google Noticias**

<https://news.google.com.mx/> ▼

Amplia cobertura actualizada de **noticias**, agregada a partir de fuentes de **noticias** de todo el mundo por Google Noticias.

La parte de “Amplia cobertura actualizada...” es la descripción del sitio.

### 1.4.2.4 Keywords.

Son las palabras claves que el buscador como google, utilizan para posicionar nuestra página al momento de realizar una búsqueda.

### 1.4.2.5 Robots.

Sirve para dar instrucciones a los rastreadores para indexar nuestra página, por ejemplo:

**index / noindex:** con la que indicaremos si deseamos indexar o no la página web en sus motores de búsqueda para que aparezca o no en los resultados de búsqueda de los mismos. Si no queremos indexarla, con noindex le diremos que **no** muestre esta página en sus resultados.

**follow / nofollow:** indica si debe rastrear y seguir, o no, los enlaces que contiene la página en cuestión.

**archive / noarchive,** señala si queremos que el robot pueda almacenar o no el contenido de la página web en la memoria caché interna del buscador.

**snippet / nosnippet:** para que sólo muestre el título y no la descripción en los resultados de búsqueda.

**translate / notranslate:** para ofrecer o no la traducción de la página en los resultados de búsqueda.

**noimageindex:** para no indexar las imágenes de la página.

**unavailable\_after [RFC-850 date/time]:** cuando no queremos que se muestre la página en los resultados de búsqueda a partir de la fecha y hora señaladas.

### 1.4.2.6 Author.

Para definir el autor del sitio.

## 1.4.3 Componentes Básicos.

### 1.4.3.1 Cabecera, Párrafos y Tipografía.

En un documento HTML se pueden indicar seis tipos de cabeceras (cada uno con distinto tamaño de letra predeterminado) por medio de las etiquetas <h> </h>.

<h1>Texto muy grande</h1>

<h2>Texto grande</h2>

<h3>Texto algo más grande de lo normal</h3>

<h4>Texto normal</h4>

<h5>Texto pequeño</h5>

<h6>Texto muy pequeño</h6>

**Práctica 4**

### 1.4.3.2 Enlaces, Imágenes y Atributos.

En HTML, los enlaces se marcan con la etiqueta `<a></a>` y el atributo `href=""`, en el cual se agrega la ubicación del archivo de destino. Este archivo puede estar en la misma carpeta que el archivo que lo llama, en una carpeta dentro del mismo sitio o en otro sitio web.

Dentro de esta etiqueta, se puede colocar cualquier elemento HTML, tales como un botón, una imagen o una ruta. En el caso del texto, el navegador por defecto lo muestra en color azul y subrayado, en el caso de las imágenes, algunos navegadores pueden agregar un borde azul.

PRACTICA 5



Atributo	Descripción
align	Especifica el alineamiento horizontal del elemento.
alt	Texto alternativo en caso de que la imagen no se pueda mostrar.
bgcolor	Color de fondo del elemento.
border	El ancho del borde.
class	Usualmente empleado con CSS para aplicar estilos a elementos con propiedades en común.
color	Este atributo establece el color de texto utilizando un nombre de color o en formato hexadecimal #RRGGBB.
height	Especifica la altura de los elementos enumerados aquí. Para todos los demás elementos, usa la propiedad CSS height en su lugar.
id	Identificador el elemento para acceder al mismo desde CSS o Javascript, este indicador debe ser único para cada elemento.
src	La URL del contenido integrable.
style	Define los estilos CSS que anulan los estilos establecidos previamente.
title	Texto a ser mostrado cuando el cursor esté sobre el elemento.
width	Para los elementos enumerados aquí, esto establece el ancho del elemento.

### 1.4.3.3 Listas y Tablas.

HTML ofrece diversas opciones para especificar listas de información. Las listas ordenadas utilizan la etiqueta `<ol></ol>`, en donde cada punto de la lista se indica con la etiqueta `<li></li>`. En este tipo de listas, el documento genera automáticamente un número (en orden) para cada punto que se agrega.

Las listas no ordenadas se enmarcan en la etiqueta `<ul></ul>`, indicando con la etiqueta `<li></li>` cada punto de la lista. Con el atributo “type”, podemos definir el símbolo de cada uno de los puntos, por ejemplo: puntos negros (`li type=”disc”`), puntos sin fondo (`li type=”circle”`) o cuadros (`li type=”square”`).

Por último, si lo que se requiere es hacer un listado de definiciones, se pueden utilizar las etiquetas `<dl>`, `<dt>` y `<dd>`.

La etiqueta `<dl>` hace referencia al término “Definiton list” e indica que, dentro de ella, entre ella y su cierre, va a ir una definición.

La etiqueta `<dt>` viene del término “Definition term” y dentro de ella se agrega el término que vamos a definir, por ejemplo, el título de la definición.

La etiqueta `<dd>` viene de “Definition description” y nos dice que dentro de ésta va la definición.

Atributo	Descripción
bgcolor	Color de fondo del elemento.
border	El ancho del borde.
bordercolor	El color del borde.
cellspacing	Define el espacio entre las celdas de la tabla.
cellpadding	Espacio que se debe dejar alrededor del texto dentro de una celda.
width	Para los elementos enumerados aquí, esto establece el ancho del elemento.

## PRACTICA 7



### 1.4.3.4 Formularios y Campos de entrada.

Los formularios HTML ayudan a aumentar la interactividad en las páginas web y permiten recibir información de los usuarios. Los formularios HTML están compuestos por campos de texto y botones. Después de que el usuario haya rellenado el formulario e introducido los valores en los campos, éstos son enviados para poder procesarlos.

Los formularios se indican por medio de la etiqueta `<form>` y su cierre `</form>`. Todo lo que vaya dentro de estas etiquetas, será parte del formulario. Esta etiqueta contiene una serie de atributos básicos:

Atributo	Descripción
action	Indica el tipo de acción que va a realizar el formulario.
method	Le indica al formulario la forma en la que el formulario será enviado (post y get).
enctype	Indica la forma en la que viajará la información que se mande en el formulario (text/plain).

Los campos de entrada de los formularios se definen por medio del tag `<input>` y sus diversos valores y atributos. El atributo `type= "text"` le da a la caja una apariencia sencilla la cual se utiliza para guardar información corta. Los siguientes atributos ayudan a completar el estilo de la caja:

Atributo	Descripción
size	Indica el tamaño de la caja.
maxlength	Sirve para definir la cantidad máxima de letras que se pueden escribir en la caja.
value	Define un texto pre escrito sobre la caja.
name	Le agrega nombre a un campo

## PRACTICA 8

### 1.4.3.5 Niveles de Elementos en Bloque o en una Línea.

Los elementos, en HTML son usualmente elementos "en bloque" o elementos "en línea". Los elementos en bloque comienzan en una nueva línea, pero los elementos en línea pueden comenzar en cualquier parte de una línea. Un elemento en línea es aquella que ocupa el espacio mínimo necesario en horizontal, y permite que otro elemento se coloque a su lado. En cambio, un elemento en bloque, ocupa todo el ancho disponible y no permite que otro elemento se coloque a su lado. Los elementos en bloque pueden contener elementos en línea y otros elementos en bloque. Inherente a esta distinción estructural es la idea de que los elementos en bloque crean estructuras "más grandes" que los elementos en línea.

Los navegadores suelen mostrar el elemento a nivel de bloque con un salto de línea antes y después del elemento.

Las etiquetas de línea más usadas son:

`<a>`, `<span>`, `<strong>`, `<img>`, `<input>`, `<code>`

Las etiquetas de bloque más usadas son:

`<h1>`, `<h2>`, `<h3>`, `<p>`, `<ul>`, `<li>`, `<div>`, `<header>`, `<nav>`, `<section>`, `<article>`, `<footer>`,  
`<form>`, `<table>`

### 1.4.3.6 Divs & Spans, Clases y Ids.

Las etiquetas <div> y <span>, no tienen significado especial, solo que la primera se define como un elemento de bloque y por tanto el navegador mostrará un salto de línea antes y después de la misma, mientras que la segunda es un elemento de línea.

### 1.4.3.7 Entidades HTML.

En ocasiones, es necesario insertar símbolos dentro del documento HTML, para ello, existen las entidades. Las entidades son estructuras que, mediante el uso de la codificación, nos permite representar un símbolo. Algunas entidades son:

Carácter	Código	Entidad	Carácter	Código	Entidad
!	&#33;	—	"	&#34;	—
#	&#35;	—	\$	&#36;	—
%	&#37;	—	&	&#38;	—
'	&#39;	—	(	&#40;	—
)	&#41;	—	*	&#42;	—
+	&#43;	—	,	&#44;	—
-	&#45;	—	.	&#46;	—
/	&#47;	—	:	&#58;	—
;	&#59;	—	<	&#60;	—
=	&#61;	—	>	&#62;	—
?	&#63;	—	@	&#64;	—

### 1.4.3.8 Etiquetas HTML5.

HTML5 utiliza unas etiquetas que le aportan semántica. El lenguaje es capaz de diferenciar, por ejemplo, entre el encabezado del documento y el pie. Google y otros buscadores son capaces de detectar dicha estructura.

Las etiquetas referentes a la estructura en HTML5 serían:

#### **<header>**

Representa un grupo que sirve de introducción a un contenido.

Normalmente define la cabecera del documento, en donde suelen ir el título del sitio y el logo.

#### **<hgroup>**

Sirve para agrupar por ejemplo un título con su subtítulo. Por ejemplo, el título y el lema:

#### **<nav>**

Para incluir dentro (generalmente en una lista) el menú principal de una web. También se puede utilizar para los menús secundarios.

## **<main>**

Especifica el contenido principal del documento.

El contenido que alberga debería ser único. Es decir, que ninguno de los contenidos debería estar repetido en otras páginas del sitio web, como ocurre por ejemplo con barras laterales, menús, avisos legales, logos del sitio o formularios de búsqueda.

## **<article>**

Representa un contenido que puede ser distribuido o reutilizado como parte independiente, por ejemplo, en sindicación. Por ejemplo, podría ser una entrada de un foro o de bitácora, el artículo de un periódico, o un comentario que envía un usuario.

## **<section>**

Representa una sección general de un documento o aplicación. Es un grupo temático de contenido, y suele tener un encabezado.

Por ejemplo, los capítulos de un libro, las pestañas de un contenido dividido en pestañas, o las secciones numeradas de una tesis. La página de inicio de un sitio web podría dividirse en secciones tales como la introducción, novedades, y la información de contacto.

## **<aside>**

Es una sección que de alguna forma está relacionada con el contenido principal que la rodea, pero se puede considerar aparte de dicho contenido.

## **<footer>**

Representa el pie del contenido más cercano. Suele contener información como el autor, el copyright, o enlaces relacionados con el documento.

Podemos encontrar un pie en cualquiera de las otras secciones del documento (header, article o section).

## **<figure> y <figcaption>**

### ***<figure>***

Se utiliza para meter dentro una imagen con una foto, ilustración o gráfico que ilustra el contenido de un artículo o sección.

### ***<figcaption>***

Sería el pie de foto con la leyenda del gráfico, y su empleo es opcional.

El figure iría dentro de una sección o artículo, para que se relacione con él.

Por ejemplo:

```
<article>
  <h1>TÍTULO DEL ARTÍCULO</h1>
  <p>párrafo del artículo</p>
  <section>
    <h1>TÍTULO DE LA SECCIÓN</h1>
    <p>párrafo de la sección</p>
    <figure>
      
      <figcaption>texto del pie de foto</figcaption>
    </figure>
  </section>
</article>
```