Curso de HTML y CSS

**Clase 1**

**¡Bienvenido al Curso!**

En este curso vamos a aprender a diseñar y programar productos digitales, vamos a utilizar Adobe XD como herramienta de diseño y Paypal para soportar pagos en línea. En esta ocasión nos acompaña Rulótico González, diseñador y programador al mismo tiempo en SABLE STUDIO.

**Clase 2**

**¿Cómo se construye la tecnología web?**

El **Frontend** es la parte del software que interactúa con los usuarios, en cambio, el **Backend** es la parte que no puedes ver, esta oculta porque contiene la información privada o sensible de nuestros usuarios.

HTML nos permite crear la estructura de la página: títulos, párrafos, menús, etc. CSS nos permite configurar los estilos del HTML, los colores, formas, posiciones, etc. Además, JavaScript *(JS)* se encarga de la parte funcional, nos permite conectarnos con otros servicios (como PayPal) y nos ayuda a hacer las interacciones mucho más fluidas.

Los **Compiladores** nos ayudan a construir el frontend de nuestras páginas web de forma mucho más rápida y sencilla. En vez de escribir el mismo código una y otra vez, vamos a utilizar estas herramientas para obtener estos mismos resultados de forma automatizada.

El trabajo y las operaciones del backend se pueden resumir con las siglas **CRUD**: *Create*, *Read*, *Update* y *Delete*. Cuando el frontend se comunica con el backend, debe indicar el tipo de operación y los datos necesarios para que todo funcione correctamente.

Por ejemplo: cuando registramos un usuario, el frontend debe mandar los nombres y contraseña de este usuario para que, el backend, pueda guardar esta información en la base de datos y podamos consultarla más adelante.

**Clase 3**

**Comprendamos Internet**

Internet es un conjunto descentralizado de redes de comunicación interconectadas, en otras palabras, son dos (o más) computadoras que se conectan entre sí. Los Protocolos son un conjunto de reglas que hacen posible la comunicación entre diferentes elementos que forman parte de un sistema.

La **World Wide Web** es un sistema de distribución de documentos (de hipertexto o hipermedia) interconectados y accesibles vía internet, mientras que, los **Hipertextos** son textos que contienen enlaces a otros textos.

**FTP** es el protocolo de transferencia de archivos entre sistemas conectados a una red, así es cómo diferentes personas podemos compartir documentos entre nosotros.

Hay algunas computadoras que solo utilizamos para entregar documentos los conocemos como **servidores**, en cambio, las computadoras que solo leen y reciben estos documentos los conocemos como **clientes**. También existe el protocolo de comunicación **P2P** (*Peer to Peer*) donde una misma computadora trabaja como servidor y cliente al mismo tiempo.

La **tecnología de la la información** es la aplicación de ordenadores y equipos de telecomunicación para almacenar, recuperar, transmitir y manipular datos, con frecuencia, utilizado en el contexto de negocios o empresas.

La **Comunicación Síncrona** es comunicación en tiempo real, por ejemplo, en aplicaciones de mensajería y video-chat. Sin embargo, también existe la **Comunicación Asíncrona**, comunicación en tiempo NO real, así funciona Gmail, puede pasar que recibamos un nuevo correo pero no lo leemos inmediatamente lo envían, por eso es comunicación asíncrona.

**Clase 4**

**Etiquetas y sus atributos**

Las **Etiquetas** son fragmentos de texto rodeados por corchetes angulares (< y >) con funciones y usos específicos:

<html>Contenido</html>

Los **Atributos** afectan a los elementos por su presencia o enriquecen la definición de la misma. Por ejemplo, el atributo lang en la etiqueta html indica que el contenido de esta página esta en un idioma específico:

<html lang=""es"">Contenido en Español</html>

Las etiquetas **meta** (*meta tags*) son etiquetas que se incorporan en el encabezado de una página web y son invisibles para los usuarios normales, sin embargo, son de gran utilidad para navegadores u otros programas que pueden valerse de esta información:

<head>

<meta name=""description"" content=""Descripción de nuestra página"">

</head>

También encontramos etiquetas comunes como <footer> para agrupar el contenido del pie de página, <span> para guardar texto genérico y sin reglas de espaciado o tamaño predeterminadas, <img src=""imagen-url.png""> para incluir imágenes, entre otras.

**Clase 5**

**¿Cómo funciona CSS?**

El **CSS** son las hojas de estilo en cascada que definen la apariencia de nuestros documentos en HTML.

Para que nuestros estilos CSS se apliquen correctamente a nuestras páginas web, debemos utilizar la etiqueta link con el atributo href y la ruta a nuestro archivo .css:

<html>

<head>

<link rel=""stylesheet"" href=""estilos.css"">

</head>

<body>

... etc ... etc....

</body>

</html>

Los **Selectores** nos permiten conectar las etiquetas de HTML con sus respectivos estilos en CSS.

Existen muchos tipos de selectores, por ejemplo, los selectores de **clase** buscan el contenido que tenga un cierto valor en su atributo class:

CSS (con punto antes del nombre de la clase):

.caja { color: red; }

HTML:

<div class=""caja"">Contenido</div>

También tenemos selectores de tipo **ID** (estos selectores solo pueden aplicar a un elemento, no va a funcionar si escribimos dos o más etiquetas con el mismo ID):

CSS (con `#` antes del nombre del ID):

#caja { color: red; }

HTML:

<div id=""caja"">Contenido</div>

En CSS utilizamos atributos para definir los estilos de nuestros elementos, podemos modificar el color de la letra, tamaño, color de fondo, espaciado, entre otras:

.caja {

color: red;

background: yellow;

font-size: 10px;

padding: 20px;

}

**Clase 6**

**¿Cómo funciona JavaScript?**

**JavaScript** es un lenguaje de programación que nos permite realizar actividades complejas en nuestras páginas web: almacenar valores en variables o realizar operaciones.

Para incluir JavaScript en nuestro HTML debemos utilizar la etiqueta script:

<html>

<head>

<script src=""script.js"">

</script>

</head>

</html>

**Clase 7**

En esta clase veremos algunas de las etiquetas más utilizadas y para qué sirven. Esta lista te será muy útil en el proceso de realización de nuestro proyecto.

<!doctype html>

Define que el documento está bajo el estándar de HTML.

**<head>**

Representa una colección de metadatos acerca del documento, incluyendo enlaces a, o definiciones de, scripts y hojas de estilo.

**<title>**

Define el título del documento, el cual se muestra en la barra de título del navegador o en las pestañas de página. Solamente puede contener texto y cualquier otra etiqueta contenida no será interpretada.

**<base>**

Define la URL base para las URLs relativas en la página.

**<link>**

Usada para enlazar JavaScript y CSS externos con el documento HTML actual.

**<meta>**

Define los metadatos que no pueden ser definidos usando otro elemento HTML.

<style>

Etiqueta de estilo usada para escribir CSS en línea.

**<body>**

Representa el contenido principal de un documento HTML. Solo hay un elemento <body> en un documento.

**<section>**

Define una sección en un documento.

**<nav>**

Define una sección que solamente contiene enlaces de navegación

**<article>**

Define contenido autónomo que podría existir independientemente del resto del contenido.

**<aside>**

Define algunos contenidos vagamente relacionados con el resto del contenido de la página. Si es removido, el contenido restante seguirá teniendo sentido

<h1>, <h2>, <h3>, <h4>, <h5>, <h6>

Los elemento de cabecera implementan seis niveles de cabeceras de documentos; <h1> es la de mayor jerarquía y <h6> es la de menor importancia. Un elemento de cabecera describe brevemente el tema de la sección que introduce.

**<header>**

Usualmente contiene un logotipo, el título del sitio Web y una tabla de navegación de contenidos.

**<footer>**

Define el pie de una página o sección. Usualmente contiene un mensaje de derechos de autoría, algunos enlaces a información legal o direcciones para dar información de retroalimentación.

**<address>**

Define una sección que contiene información de contacto.

**<main>**

Define el contenido principal o importante en el documento. Solamente existe un elemento <main> en el documento.

**<form>**

Representa un formulario, con atributos de controles que puede ser enviado a un servidor para procesamiento.

**<img>**

Representa una imagen.

**<iframe>**

Representa un contexto anidado de navegación, es decir, un documento HTML embebido.

**<a>**

Representa un hiperenlace, enlazando a otro recurso.

**<em>**

Representa un texto enfatizado, como un acento de intensidad.

**<strong>**

Representa un texto especialmente importante .

**<small>**

Representa un comentario aparte, es decir, textos secundarios como un descargo de responsabilidad o una nota de derechos de autoría, que no son esenciales para la comprensión del documento.

**Clase 8**

**Maquetación de main\_section y agregando estilos CSS**

Vamos a trabajar la primera parte de nuestra página web.

En el diseño podemos ver que utilizamos la imagen curve.png por debajo de los productos, para esto debemos añadir la imagen en nuestro CSS en vez de en el HTML, así podremos posicionarla utilizando la propiedad background-image: url(""../images/curve.png"");.

**Clase 9**

**Maquetación de tarjetas de producto**

Vamos a utilizar el atributo margin-top de CSS para darle espacio a nuestros elementos de producto, entre más grande sea el valor que configuremos en este atributo, más grande será el espaciado entre estos elementos y los de arriba, además, podemos utilizar valores negativos (margin-top: -50px) para disminuir este espaciado y conseguir el efecto que diseñamos.

También vamos a añadir sombras a nuestros elementos utilizando el atributo box-shadow: 0 0 2px black; de CSS. Si quieres configurar el color y dirección de estas sombras, puedes hacer click derecho en tus elementos, seleccionar ““Inspector de Elementos”” y configurar las sombras como mas te gusten, recuerda copiar y pegar tus modificaciones porque, por defecto, no se guardan cuando refresques la página.

**Clase 10**

**Fuentes personalizadas y variables de CSS**

Las Variables de CSS nos permiten centralizar los valores repetitivos por todo nuestro CSS, vamos a utilizar estas funcionalidades para ahorrarnos mucho si queremos cambiar el valor de algún color o tamaño de nuestra página.

También vamos a utilizar fuentes personalizadas, para esto debemos ir a [Google Fonts](https://fonts.google.com/) y elegir la que mejor se acomode a nuestro diseño, en este caso es la fuente Open Sans.

**Clase 11**

A continuación veremos las reglas de nomenclatura que dan orden a nuestra definición de CSS.

Para darle nombre a los estilos podemos utilizar diferentes métodos, aunque es recomendable seguir estos pasos:

-No utilizar palabras en español: la mayoría de los lenguajes no aceptan ñ o tildes, es por eso que evitamos esta característica.

-Para los espacios utilizamos guión medio o guión bajo, recuerda usar solo uno de los dos caracteres.

-Describir el nombre de manera general a particular, por ejemplo: home\_bar

También podemos definir las siguientes etiquetas, las cuales nos permiten crear secciones dentro de HTML y evitamos un concepto llamado ***divismo***.

**<section>** Lo usamos para dividir secciones  
**<footer>** Define el footer ya sea de toda la página o solo de un componente.  
**<iframe>** Nos ayuda a incrustar otra página en nuestro html  
**<video>** Nos ayuda a incrustar video dentro de nuestra web

**Clase 12**

**Reglas responsive**

**Responsive Design** consiste en crear estilos que se adapten a cualquier tamaño y posición de nuestros dispositivos electrónicos. Para esto, la mayoría de elementos organizados horizontalmente deben pasar a organizarse verticalmente.

La forma de añadir código CSS que se ejecute para tamaños de pantalla específicos es la siguiente:

@**media** (max-width: 600px) { /\* 600px es solo un ejemplo \*/

/\* Todos nuestros estilos responsive \*/

}

**Clase 13**

**Animaciones y transiciones**

Las animaciones nos permiten cambiar los estilos de nuestros elementos durante un tiempo. Para esto debemos configurar nuestra animación con algunos estilos iniciales y finales para aplicarla a alguno de nuestros selectores de CSS:

@**keyframe** fadeIn {

**from** {

opacity: 0;

}

**to** {

opacity: 1;

}

}

.fadeIn {

animation-name: fadeIn;

animation-duration: 4s;

animation-fill-mode: 4s;

}

No olvides repetir los atributos de animación o transiciones utilizando el prefijo -webkit-:

.fadeIn {

animation-name: fadeIn;

-webkit-animation-name: fadeIn;

animation-duration: 4s;

-webkit-animation-duration: 4s;

animation-fill-mode: 4s;

-webkit-animation-fill-mode: 4s;

}

**Clase 14**

**Atributos para asociar labels a sus respectivos inputs**

Seguramente has llenado miles de formularios desde tu computadora sin problemas. Pero la probabilidad de que tengas un mal rato cuando te encuentras con un formulario desde tu celular es muy alta. Por ser dispositivos mucho más pequeños es mucho más difícil atinarle al input correcto para empezar a escribir los datos que nos pide el formulario.

De hecho, suele pasar que, debido a todos estos problemas, terminamos abandonando esa página web y nos olvidamos de ella para siempre. Nuestra tarea es evitar este tipo de problemas a toda costa.

Solucionar esto es muy fácil. Vamos a crear la funcionalidad de que, al tocar el label que corresponde a cada input, la página reaccione como si hubiéramos hecho click directamente en el input.

Hay dos formas de hacerlo:

**Primera forma**: Podemos encerrar nuestros inputs dentro de sus respectivos labels:

<label>

Nombre:

<input type=”text” />

</label>

<label>

Edad:

<input type=”number” />

</label>

**Segunda forma**: Sólo debemos añadir un par de atributos a los elementos de nuestro formulario:

* Añadimos el atributo id en la etiqueta input y le damos un valor, recuerda que el valor de nuestros IDs deben ser únicos e irrepetibles para evitar problemas en nuestra página web.
* Añadimos el atributo for en la etiqueta label con el valor del id del input que queremos enlazar.

<label for=”nombre”>Nombre:</label>

<input type=”text” id=”nombre” />

<label for=”edad”>Edad:</label>

<input type=”number” id=”edad” />

**Atributos para enviar los datos de nuestros formularios**

El objetivo de los formularios es enviar y almacenar los datos de nuestros usuarios en algún sitio, ¿verdad?

La forma de hacerlo es mucho más fácil de lo que crees. Solo debemos añadir algunos atributos en la etiqueta <form> y en los inputs y botones de nuestro formulario.

* **Action**: Este atributo de la etiqueta <form> nos permite establecer el lugar o URL donde los datos de los usuarios deben ser enviados una vez ellos terminen de llenar el formulario.
* **Name**: Este atributo de las etiquetas <input> nos permite que los datos que los usuarios hayan escrito se envíen correctamente cuando han terminado de llenar el formulario. El valor de esta etiqueta debe indicar el nombre con el cual queremos almacenar o interactuar con esta información en el futuro.
* **Type**: El atributo type=”submit” le permite a los usuarios indicar que han terminado de llenar el formulario y lo pueden enviar para almacenar la información. También podemos usar este atributo en inputs para que estos pasen a tener el rol de botón de envío del formulario.

<form action=”URL\_DONDE\_QUIERO\_ENVIAR\_LO\_DATOS”>

<label for=”nombre”>Nombre:</label>

<input type=”text” id=”nombre” name=”user\_name” />

<label for=“edad“>Edad:</label>

<input type=”number” id=”edad” name=”user\_age” />

<button type=”submit”>Enviar mis datos</button>

</form>

**Clase 15**

**Librería vs Framework**

Planea, prototipa y crea tu producto digital. Aprende más en el curso Profesional de Desarrollo Web en Platzi disponible en tu suscripción en: <https://platzi.com/web>  
Adquiere hoy la suscripción de Platzi en: <https://platzi.com/precios>

En esta clase vamos a aprender qué son los frameworks y librerías, entenderás cómo el uso de estos dos elementos de programación te ayudarán a encontrar soluciones.

Cada lenguaje de programación tiene sus propios frameworks y librerías, por ejemplo, JavaScript tiene frameworks como Angular.js y librerías como React.js para simplificar problemas de visualización de contenido. ¡Las librerías y frameworks existen para optimizar tu tiempo!

**Clase 16**

**Sistema de Grillas**

Los sistemas de grillas nos ayudan a configurar el espacio y tamaño de nuestros elementos. En este caso, vamos a separar nuestra grilla en 12 posibles tamaños, así que debemos realizar la operación 100/12 y multiplicar el resultado por cada posible tamaño de nuestra grilla.

.col-1 {

width: 8.3333333333%;

}

.col-2 {

width: 16.6666666666%;

}

.col-3 {

width: 24.9999999999%;

}

...

**Clase 17**

**Configuración del botón de pago con Paypal**

El botón de pago es una herramienta de PayPal que ayuda a configurar la cantidad y el método de pago para que nuestro usuarios paguen por nuestros productos o servicios de forma fácil y rápida.

Cuando configuremos nuestro botón, vamos a copiar la URL que generamos con PayPal y la vamos a pegar en el atributo action de nuestro formulario, además debemos añadir el atributo method=""POST"" para que nuestro formulario funcione correctamente.

**Clase 18**

**Subiendo nuestro proyecto a Internet con Github**

Para subir nuestro proyecto a internet vamos a utilizar **Github Pages**, una herramienta que se conecta con nuestros repositorios de Github.

Antes de utilizar esta herramienta vamos a descargar **Github Desktop**, una herramienta que nos facilita la comunicación con la plataforma de Github. Vamos a utilizarla para subir nuestro proyecto a Github y, más adelante, subirla a internet con Github Pages.

**FIN.**