INNOVARE SUITE

WEB DEVELOPMENT

PARTE 1

FULL STACK
DEVELOPER
JUNIOR

Introducción al desarrollo Web

El desarrollo Web es uno de los campos más emocionantes de la actualidad. No existe una sola industria que no haya sido o no esté siendo transformada por Internet. Es una revolución sin precedente, pero este es solo el comienzo. Sectores como la educación, la política y la salud, entre muchos otros hasta ahora están empezando a ser transformados.

Cualquier persona que pueda usar una pc se puede convertir en Desarrollador Web. A diferencia de los que muchos creen no necesitas ser bueno en matemáticas ni tener un cerebro especial. Tampoco necesitas gastar una fortuna y asistir a una universidad por 4 años. Necesitas ser perseverante, optimista y determinado.

Este curso te mostrará cómo convertirte en Desarrollador Web: la mentalidad que necesitas desarrollar, las tecnologías que debes aprender, por dónde empezar y enfocarte en lo verdaderamente importante que cualquier empresa necesita y solicita en la actualidad.

¿Qué hace un desarrollador Web?

Un desarrollador Web se encarga de los aspectos técnicos de un sitio o de una aplicación Web. A diferencia del Diseñador Web, que se encarga de la apariencia visual y la distribución de los elementos en el diseño, el desarrollador Web convierte un diseño estático en una aplicación completamente funcional disponible en Internet.

Existen varios roles en el desarrollo Web. Un desarrollador Web puede suplir uno o varios de estos roles al mismo tiempo.

FRONTEND

Es la persona que se encarga de convertir un diseño en código que entiendan los navegadores usando HTML, CSS y JavaScript.

Este es quizás el rol con más demanda en la actualidad y el que más rápido está evolucionando. La razón es que las empresas buscan que sus sitios y aplicaciones se adapten a diferentes pantallas, tengan animaciones, se actualicen sin necesidad de refrescar la página, reaccionen rápidamente al usuario, etc.

BACKEND

Se encarga de conectar la aplicación con la base de datos y otros servicios externos. Son los responsables de la autenticación y autorización de usuarios, lectura y escritura de información de la base de datos, envío de correos automatizados, tareas recurrentes y API's, entre otros.

Las personas que ejecutan este rol se encarga de desplegar la aplicación en Internet y monitorearla. También se encargan de manejar los servidores, hacer copias de seguridad de la base de datos y en general estar atentos al correcto funcionamiento de la aplicación en producción.

A los desarrolladores que hacen todos estos roles se les conoce como **FULL STACK DEVELOPERS.**

La mentalidad del Desarrollador Web

La característica más importante es ser capaz de adaptarse y aprender nuevas tecnologías rápidamente.

Por otro lado, es importante entender que la programación es una herramienta para resolver problemas. Cada problema es diferente, requiere su propio aprendizaje y tiene su propia dificultad.

¿Cómo empezamos?

Lo mínimo que debes aprender es HTML Y CSS, que son la base para la creación de las páginas Web. Necesitas un editor de texto: lo recomendable es instalar un IDE (entorno de desarrollo integrado) que te hará la vida más simple a la hora de programar, ya que son entornos preparados para la escritura de código. El recomendado en este caso es VSCode, que debes descargarlo de su página oficial e instalarlo en tu pc.

Hipertexto

SISTEMA QUE PERMITE ENLAZAR FRAGMENTOS DE TEXTOS ENTRE SÍ. LA LECTURA NO SEA LINEAL, SINO QUE EL USUARIO ACCEDA A LA INFORMACIÓN A TRAVÉS DE LOS ÍTEMS RELACIONADOS.

El siguiente paso es aprender el lenguaje de programación JavaScript que es el único lenguaje de programación que entienden los navegadores.

Otra de las herramientas importantes que vas a aprender a manejar es un sistema de control de versiones, preferentemente **Git**. Este sistema guarda el historial de cambio del código y te permite trabajar sobre el mismo proyecto con más personas de forma simultánea.

Aprenderás sobre la librería **ReactJS**, una de las más demandadas en la actualidad.

Y trabajaremos con SCRUM, un marco ligero que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptables para problemas complejos.

Utilizaremos la biblioteca Express que te permitirá implementar una API Rest en Node.js.

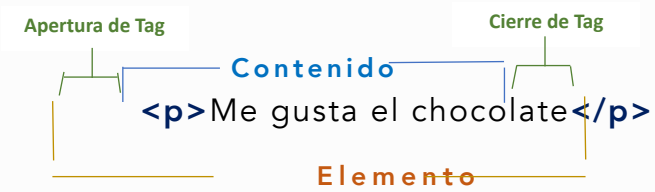
Como base de datos almacenaremos la información en MongoDB, una base de datos de tipo No Relacional.

Introducción a HTML: Páginas HTML

Las páginas html son las que alojan el contenido general que compone cada texto. Estas páginas están armadas con contenido HTML(Hipertext Marckup Language), contenido CSS, JavaScript, imágenes, videos, etc. Todo ese contenido es armado de forma óptima y estática, para que pueda ser visualizado a través de los navegadores. En los casos donde los contenidos requieren interacción con base datos, se denomina a las páginas web como dinámicas.

HTML no es un lenguaje de programación; es un lenguaje de marcado que define la estructura del contenido de nuestra página.

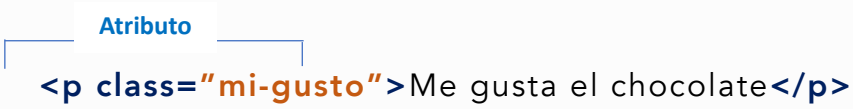
Anatomía de un elemento HTML



Las partes principales del elemento son:

1. *La etiqueta (tag) de apertura:* Establece dónde comienza a tener efecto el elemento, en el ejemplo, el comienzo del párrafo.
2. *La etiqueta de cierre:* Establece donde termina el elemento, donde termina el párrafo.
3. *El contenido:* en este caso es sólo texto.
4. *El elemento:* la etiqueta de apertura, más la de cierre, más el contenido.

Los elementos pueden tener también atributos:



Contienen información adicional acerca del elemento, que no queremos que aparezca en el contenido real del mismo. En el ejemplo, **class** es el nombre del atributo y **mi-gusto** el valor del atributo. Esto permite darle al elemento un nombre identificativo, que puede ser usado luego para referirse a él desde los estilos o demás cosas.

Los atributos siempre se incluyen en la etiqueta de apertura de un elemento.

Anidar elementos

Se trata de colocar elementos dentro de otros elementos.

```
<p>Me gusta el <strong>chocolate</strong></p>
```

En el ejemplo, hacemos un énfasis en la palabra 'chocolate', como se ve, las etiquetas deben abrirse y cerrarse ordenadamente, de tal forma que se encuentren claramente dentro o fuera el uno del otro.

Elementos vacíos

Son aquellos elementos que no poseen contenido, veamos el siguiente ejemplo:

```
<img src="images/chocolate.png" alt="Imagen de chocolate">
```

La etiqueta 'img' posee dos atributos, pero no tiene etiqueta de cierre, ni contenido encerrado. Esto es porque un elemento de imagen no encierra contenido al cual afectar. Su propósito es desplegar una imagen en la página HTML.

Tipos de elementos

HTML clasifica a todos los elementos en dos grupos: **inline** y **block**. Por defecto, los elementos en bloque comienzan en una nueva línea y los elementos en línea pueden comenzar en cualquier parte de una línea.

Elementos INLINE: **
, <a>, <img>, , <b>, , <mark>, <sub>, etc.**

Elementos BLOCK: ** , <p>, <h1>..<h6>, <ul>, <ol>, <li>, <table>, <form>, etc.**

Estructura de un documento HTML

Los elementos individuales vistos son combinados para formar una página HTML, ya que éstos por sí sólo no son muy útiles. Por ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mi página de prueba</title>
  </head>
  <body>
    <img src="images/chocolate.png" alt="Imagen de chocolate">
  </body>
</html>
```

Descripción de la estructura

- **<!DOCTYPE html>** el tipo de documento. Es un preámbulo requerido. Indica que estamos trabajando con la versión HTML5.
- **<html></html>** Este elemento encierra todo el contenido de la página entera.
- **<head></head>** Este elemento actúa como un contenedor de todo aquello que se incuye en la página HTML que no es contenido visible para los visitantes de la página. Incluye cosas como palabras clave, una descripción de la página que queremos que aparezca en resultados de búsquedas, código CSS para dar estilo al contenido, declaraciones del juego de caracteres, etc.
- **<meta charset="utf-8">** Establece el juego de caracteres que el documento usará en 'utf-8', que incluye casi todos los caracteres de todos los idiomas humanos. Puede manejar cualquier contenido de texto incluido en la página.
- **<title></title>** establece el título de la página, es el título que aparece en la pestaña o en la barra de título del navegador cuando se carga la página, y se usa para describirla cuando es añadida a los marcadores o como favorita.
- **<body></body>** Encierra todo el contenido que queremos que los usuarios vean cuando visiten la página, ya sea texto, imágenes, videos, juegos, pistas de audio reproducibles, y demás.

Comentarios: se pueden introducir comentarios dentro del código HTML mediante la inclusión de los caracteres **<!--** y **-->**.

Ej. **<!--Esto es un comentario-->**

Imágenes

Como ya se dijo, este elemento incrusta una imagen en la página. Lo logra a través del atributo 'src' (source), el cual contiene el path (ruta) del archivo de imagen.

También se incluye un atributo 'alt' (alternative), el cual contiene un texto que debería describir la imagen, y que podría ser accedido por usuarios que no pueden verla, quizás porque:

- Son ciegos o tienen deficiencias visuales. Usualmente éstos utilizan herramientas llamadas *lectores de pantalla* (Screen Readers), que leen el texto contenido en el atributo alt.
- Se produjo algún error en el código que impide que se cargue la imagen.

Así que debemos tener en cuenta esto para realizar la mejor descripción posible de la imagen que estamos mostrando.

Marcado de texto

Veremos algunos de los elementos HTML básicos que se utilizan para el marcado de texto.

Encabezados

Permiten especificar que ciertas partes del contenido son encabezados, o subencabezados del contenido. Html posee 6 niveles de encabezados:

<h1>Mi título principal</h1>

<h2>Mi título de nivel superior</h2>

<h3>Mi subtítulo</h3>

<h4>Mi sub-subtítulo</h4>

<h5>Mi sub-sub-subtítulo</h5>

<h6>Mi sub-sub-sub-subtítulo</h6>

Párrafos

Se utilizan para encerrar párrafos de texto; se usan comúnmente para contenido de texto regular.

```
<p>Lorem ipsum dolor sit amet consectetur adipisicing elit.
Facilis labore eius nesciunt, voluptate modi quis tempora.
Odit nisi voluptatem praesentium unde commodi optio,
voluptate minima minus porro provident veritatis tempore
voluptatum harum vero.</p>
<p>Lorem ipsum dolor sit amet consectetur adipisicing
elit.</p>
```

Listas

El marcado de listas se realiza siempre en al menos dos elementos. Los dos tipos de listas más comunes son las listas ordenadas y las desordenadas:

1. **Listas desordenadas:** son aquellas en las que el orden de los ítems no es relevante, como en una lista de compras. Estas se encierran en un elemento **<ul>** (unordered list).
2. **Listas ordenadas:** son aquellas en las que el orden sí es relevante, como en una receta. Estas son encerradas en un elemento **<ol>** (ordered list).

Cada elemento de la lista se coloca dentro de un elemento **<li>** (list ítem).

```
<p>Lista desordenada</p>
  <ul>
    <li>Lunes</li>
    <li>Martes</li>
    <li>Miércoles</li>
  </ul>
```

Lista desordenada

- Lunes
- Martes
- Miércoles

** Lista ordenda **

```
<ol>
  <li>Enero</li>
  <li>Febrero</li>
  <li>Marzo</li>
</ol>
```

Lista ordenada

1. Enero
2. Febrero
3. Marzo

También podemos crear listas dependientes o listas anidadas, simplemente agregando más elementos <ul> u <ol> según corresponda. Puede ser útil cuando enumeramos ítems que contienen sub-ítems.

Vínculos

Éstos son los que hacen de la web, la Web. Para implementarlo se utiliza el elemento **<a>**, que es la abreviatura de ***anchor*** (ancla).

Ej.

```
<a href="https://www.pagina_donde_hace_el_enlace.com">La chocolatería</a>
```

Enlaces absolutos y relativos

Cuando utilizamos el elemento **<a>** con un atributo href (Hypertext Reference), como vimos más arriba, tenemos 3 tipos de enlaces:

- **Absoluto:** es un enlace que incluye todas las partes de una URL, como el del ejemplo visto.
- **Relativo:** hace referencia a una dirección que depende de la posición del archivo desde donde se utiliza. Por ej.

  ```
  <a href="img/imagen1.jpg">enlace a una imagen</a>
  ```
- **Enlaces internos (anclas):** Se utilizan para indicar un elemento dentro de la misma página, dirigiéndose a un sector específico. Para utilizarlo necesitamos el enlace propiamente dicho y el nombre del ancla al cual debemos dirigirlo.

  ```
  <a href="#quienes_somos">Ir a quienes somos</a>
  <a name="quienes_somos">quienes somos</a>
  ```

  Se pueden vincular a anclas dentro de la página de destino:

  ```
  <a href="archivo.html#seccion">contenido</a>
  ```

  Al igual que en los enlaces internos marcamos la sección con un ancla:

  ```
  <a name="seccion"></a>
  ```

Atributos de los enlaces

Dentro de los atributos principales de los enlaces podemos encontrar:

- **download:** Especifica cuál es el objetivo que se descargará cuando un usuario haga clic en el hipervínculo.
- **href:** Especifica la URL de la página a la que se dirige el enlace.
- **target:** Especifica dónde abrir el documento vinculado. Podemos utilizar algunas de estas opciones:
  - **_blank:** Abre el documento vinculado en una nueva ventana o pestaña.
  - **_self:** Abre el documento vinculado en el mismo marco en el que se hizo click.

- **title:** Especifica información adicional sobre un elemento. La información generalmente se muestra como un texto de información sobre herramientas cuando el mouse se mueve sobre el elemento.

Si aún no tenemos determinado donde queremos hacer el hipervínculo podemos colocar dentro de **href** una **#.**

Enlace a correo electrónico

Es posible crear enlaces o botones que, cuando se pulsan, abren un nuevo correo saliente en lugar de enlazar a un recurso o página. Esto se consigue con el elemento ancla **<a>** y el elemento **mailto:** seguido del esquema de la URL.

En su forma más básica, un enlace **mailto:** contiene simplemente la dirección de correo electrónico de los destinatarios. Por ejemplo:

```
<a href="mailto:nombre@full-stack.com">Enviar correo</a>
```

Esto da como resultado un enlace que al hacer click abre una ventana de correo saliente en el gestor de correo con la dirección de mail ingresada como destinatario.

Si no especificamos la dirección en **mailto:** se abrirá la ventana de correo saliente, pero en este caso sin el destinatario predeterminado, esto puede ser útil cuando se desea compartir enlaces que los usuarios pueden pulsar para enviar un correo y elegir un destinatario posteriormente.

Además de la dirección de correo electrónico, podemos proporcionar otra información. De hecho, podemos incluir cualquier campo de contenido estándar en el encabezado de cualquier mensaje en la URL **mailto** que se proporcione. Los más utilizados son el **subject** (asunto), **cc** (con copia a) o **bcc** (copia oculta), y **body** (cuerpo del mensaje, que no es un campo de la cabecera, pero permite especificar un mensaje breve).

Ejemplo:

```
<a href="mailto:nombre@full-stack.com?

cc=name2@alguien.com&bcc=name3@alguienmas.com&subject=El%20destinatario%20del%20email&body=El%20cuerpo%20del%20email"> Enviar un correo electrónico cc, bcc, asunto y cuerpo</a>
```

Nota: Los valores de cada campo deben tener la URL codificada, es decir, sin caracteres imprimibles (caracteres invisibles, tabulaciones, retornos de carro y saltos de página) y espacios con codificación porcentual (%20, indica espacio). También se debe tener en cuenta el uso del signo **?** para separar la URL principal de los valores de los campos, y el símbolo **&** para separar cada campo dentro del enlace **mailto:**.

Metadatos: el elemento <meta>
Son datos que describen datos, y HTML tiene el elemento **<meta>** para introducirlos; éstos se incorporan en el **<head>** de nuestra página.
Ya vimos la codificación de caracteres del documento que se utiliza en la parte inicial (**utf-8**) que incluye casi todos los caracteres de casi cualquier idioma.

Muchos elementos <meta> incluyen atributos *name* y *content*:

- **name** especifica el tipo de metadato del que se trata; es decir, qué tipo de información contiene.
- **content** especifica el contenido del metadato en sí.

Sería útil incluir en estos atributos, por ejemplo, el autor de la página y una descripción concisa de la página.
Especificar una descripción que incluya palabras claves relacionadas con el contenido de la página resultaría útil para poder hacer que la misma aparezca entre más arriba en las búsquedas relevantes realizadas por los *motores de búsqueda*.
Al navegar por la web también podemos encontrar otros tipos de metadatos. Muchas de las funciones que se pueden ver en los sitios web son creaciones propietarias diseñadas para proporcionar ciertos sitios (de redes sociales por ejemplo) información específica que pueden usar.

Por ejemplo. Open Graph Data es un protocolo de metadatos que Facebook inventó para proporcionar metadatos más ricos para los sitios web.
Twitter también tiene sus metadatos propios, las Twitter Cards.

Íconos personalizados
Para enriquecer un poco más el diseño de nuestro sitio podemos añadir íconos personalizados, que se mostrarán en determinados contextos. El más común de ellos es el **favicon** (**favorite icon**).
El **favicon** es el primer ícono de este tipo: un ícono cuadrado de 16 píxeles que se utiliza en varios lugares. Por ejemplo, en la pestaña del navegador que contiene la página abierta y al lado de las páginas que establecemos como favoritas en el panel de marcadores.
Los formatos de imagen más comunes son **.ico**, **.gif** o **.png**, pero usar el formato **ico** garantiza que funcionará en todos los navegadores.
Para agregar el **favicon** escribimos en el <head> la siguiente línea:

```
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon" />
```

Formatos de texto
Solo haremos una breve descripción de los más usados, aunque actualmente la mayoría ya han sido sustituidos por CSS, pero conocer de ellos nos permite modificar aspectos concretos de la fuente (estilo, índices y subíndices) sin tener la necesidad de definir un estilo específicamente para modificar solo un atributo.
Algunos de los formatos son:

<b> vs **:** Tienen el mismo efecto visual, **<b>** indica negrita y **** indica que se debe resaltar fuertemente el texto (cada navegador lo resalta como desea).
<u>: texto subrayado.
<mark>: texto marcado, resalta el texto como lo haríamos con un marcador.
<small>: texto más pequeño.
<i>: Texto en Itálica o cursiva.
<del>: Texto tachado.
<sup>: Texto en superíndice.
<sub>: Texto en subíndice.
: Texto con énfasis.

Cada una de ellas debe tener su correspondiente etiqueta de cierre.

**
:** Inserta un salto de línea.
<hr>: Traza una línea horizontal.

A diferencia de los tags anteriores, éstos últimos no requieren el de cierre.

Horas y fechas
HTML también proporciona el elemento **<time>** para marcar horas y fechas en un formato legible por la máquina. Por ejemplo:

```
<time datetime="2023-01-20">20 Enero 2023</time>
```

Esto es útil porque hay muchas formas en que se pueden escribir las fechas, la fecha anterior se podría escribir como:

- 20 enero 2023
- 20 de enero de 2023
- Ene 20 2023
- 20/01/23
- … y así…..

Estas diferentes formas no pueden ser reconocidas fácilmente por las computadoras, el elemento **<time>** permite adjuntar una fecha/hora inequívoca y legible por la máquina para este propósito. Existen muchas otras opciones que son posibles ya sea tanto para fecha como para la hora: por ejemplos podemos querer especificar sólo el año y el mes, el mes y el día, sólo la hora y los minutos, fecha y hora, fecha y hora con desplazamiento de zona horaria, etc.

Estructura Web

Partes básicas de un documento

Las páginas web pueden y se deben diferenciar entre ellas, pero todas pueden contener elementos comunes parecidos, a menos que estén destinados a mostrar un vídeo o un juego a pantalla completa, o que formen parte de un proyecto artístico concreto, o por el solo hecho de que están mal estructuradas, éstos elementos comunes son:

- **Encabezado (header):** normalmente formado por una gran franja que cruza la parte superior de la página con un gran título, un logotipo y quizás un lema. Esta parte suele permanecer invariable mientras navega entre las páginas de un mismo sitio web.
- **Barra de navegación (nav):** Son los enlaces a las secciones principales del sitio web. Normalmente está formado por un menú con botones, enlaces o pestañas. Al igual que el encabezado, este contenido suele permanecer invariable en las diferentes páginas del sitio. Muchos diseñadores consideran el menú de navegación como parte del encabezado y que no sería un componente individual, aunque no es necesario que sea así.
- **Contenido principal (main):** Es la gran parte central de la página y contiene la mayor parte del contenido de una página web concreta. ¡Esta es la parte que definitivamente debe variar mucho de una página a otra!
- **Barra lateral (aside):** Suele incluir algún tipo de información adicional, enlaces, citas, publicidad, etc. Normalmente está relacionado con el contenido principal de la página, pero en otras ocasiones encontraremos elementos recurrentes como un menú de navegación secundario.
- **Pie de página (footer):** Es la parte inferior de la página, que generalmente contiene la letra pequeña, el copyright o la información de contacto. Es el sitio donde se coloca la información que no suele ser tan importante o es secundaria con respecto a la página en sí misma.

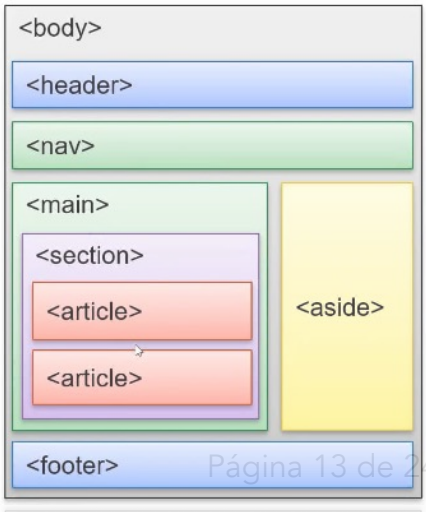
Podemos encontrar sitios con más columnas, algunas mucho más complejas, pero este modelo servirá para que nos podamos hacer una idea general sobre el tema.

Se pueden crear secciones de contenido dentro del código HTML basadas en su funcionalidad, usando elementos que representan sin ambigüedad las diferentes secciones de contenido descritas, de forma que las tecnologías de accesibilidad y los lectores de pantalla puedan reconocer esos elementos y asistir en tareas como *encontrar el menú de navegación*, o *el contenido principal, por ejemplo.*

A través de ellas los navegadores y buscadores reconocen patrones y una estructura determinada.

Debemos respetarlas porque ayudan al navegador a entender su significado para mostrarlo en pantalla y ayudan a los buscadores a reconocer el contenido y la estructura del sitio.

Estructura de una página web típica:



Existen situaciones, a veces, en la que no encontramos un elemento semántico adecuado para agrupar ciertos elementos o englobar algún contenido. Podríamos querer agrupar elementos para referenciarlos como una entidad que comparta cierto CSS o JavaScript. Para casos como estos, disponemos del elemento ** ** y de ****. Preferentemente éstos se deben utilizar con sus atributos **class**, para conferirles algún tipo de etiquetado que permita determinarlos con facilidad.

**** es un elemento no-semántico que se utiliza en el interior de una línea. Se utiliza cuando no se nos ocurre el uso de ningún otro elemento semántico de texto en el que incluir el contenido, o si no deseamos añadir ningún significado específico.

** ** Se lo puede utilizar cuando no se nos ocurra el uso de otro elemento semántico mejor, o si no deseamos añadir ningún significado concreto.

Inserción de elementos multimedia

Insertar una imagen en una página web:

Se utiliza el elemento **<img>**, es un elemento vacío, o sea, no contiene texto o etiqueta de cierre, que requiere de por lo menos un atributo, **'src'** (source). Éste contiene una ruta que apunta a la imagen que queremos poner en la página, que puede ser una URL relativa o absoluta, de la misma forma que el elemento **<a>** contiene los valores del atributo **'href'**.

Por ejemplo, si nuestra imagen se llama chocolate.jpg, y está en el mismo directorio que nuestra página HTML, deberíamos insertar la imagen de la siguiente manera:

```
<img src="chocolate.jpg">
```

O si ésta se encuentra en el directorio de nuestra página dentro de una carpeta, por ejemplo llamada **images**, especificamos simplemente la ruta:

```
<img src="images/chocolate.jpg">
```

Y así sucesivamente.

Para incrustar la imagen usando la URL absoluta, sería:

```
<img src="https://www.example.com/images/chocolate.jpg">
```

Lo que hace que el navegador trabaje más buscando la dirección IP. Así que trataremos mantener las imágenes de nuestro sitio en el mismo servidor de nuestro HTML.

El otro atributo que deberíamos agregar es **<alt>**. Su valor debe ser una descripción textual de la imagen para usarla en situaciones en que la imagen no puede ser vista/mostrada o tarda

demasiado en mostrarse por, quizás una conexión lenta a internet, etc. Al código visto recién le podríamos agregar lo siguiente:

```
<img src="images/chocolate.jpg" alt="Un trozo de chocolate sobre un plato">
```

También podemos usar los atributos **width** (ancho) y **height** (alto) para especificar la anchura y altura de la imagen.

Se puede agregar el atributo **title** a las imágenes para proporcionar más información de ayuda si es necesario. No es recomendable usarlo ya que suele presentar problemas de accesibilidad, lo mejor es incluir dicha información en el texto principal del artículo, en lugar de adjuntarla en la imagen.

Inserción de audio y video

<audio> este elemento especifica un estándar para incrustar audio en una página web. Hay dos formas diferentes de especificar la URL del archivo fuente del audio. El primero usa el atributo fuente:

```
<audio src="audio.mp3" controls>Elemento de audio no compatible con su navegador</
audio>
```

La segunda forma usa el elemento **<source>** dentro del elemento **<audio>**:

```
<audio controls>
        <source src="audio.mp3" type="audio/mpeg">
        <source src="audio.ogg" type="audio/ogg">
</audio>
```

Se pueden vincular múltiples elementos <source> a diferentes archivos de audio. El navegador utilizará el primer formato reconocido.
El elemento **<audio>** crea un reproductor de audio dentro del navegador. El texto entre las etiquetas <audio> y </audio> se mostrará en navegadores que no admitan este elemento.

Atributos de <audio>
controls: especifica que se deben mostrar los controles de audio, como los botones de reproducción/pausa, etc.
autoplay: Cuando se define este atributo, el audio comienza a reproducirse tan pronto como esté listo, sin pedir permiso al usuario.
Uso:

```
        <audio controls autoplay>
```

loop: Es utilizado para que el audio se reproduzca cada vez que finaliza.
Uso:

```
        <audio controls autoplay loop>
```

<video> este elemento nos permite incrustar video fácilmente. Es similar al elemento de audio. Podemos especificar la URL de origen del video usando un atributo en un elemento de video o usando elementos de origen dentro del elemento de video.

Ejemplo:

```
<video controls>
    <source src="video.mp4" type="video/mp4">
    <source src="video.ogg" type="video/ogg">
    Video no soportado por su navegador
</video>
```

Otro aspecto que tienen en común los elementos de audio y video es que no todos los principales navegadores admiten los mismos tipos de archivos. Si el navegador no es compatible con el primer tipo de video, probará con el siguiente.
También contiene atributos como **controls, autoplay** y **loop.**

Otras características:
widht, height que pueden controlar el tamaño de estos atributos o con CSS. En ambos casos, los videos mantienen su relación ancho-alto nativa. Si la relación de aspecto no se mantiene con los tamaños establecidos, el video crecerá para rellenar el espacio horizontalmente y el espacio sin rellenar sólo recibirá un color de fondo sólido de forma predeterminada.
muted: hace que los medios se reproduzcan con el sonido apagado de forma predeterminada.
poster: La URL de una imagen que se mostrará antes de reproducir el video. Está destinado a ser utilizado para una pantalla de presentación o pantalla publicitaria (miniatura del video).
preload: Se utiliza para almacenar en búfer archivos grandes; puede tomar uno de los siguientes tres valores:

- **"none"** no almacena el archivo en el búfer.
- **"auto"** almacena el archivo multimedia.
- **"metadata"** almacena solo los metadatos del archivo.

Existen otros elementos de incrustación como **<iframe>** para incrustar otras páginas web, **<embed>** y **<object>** para incrustar PDFs, SVG e incluso Flash (una tecnología en camino de despedida), que permiten integrar una amplia variedad de tipos de contenido en nuestra página.
Los elementos **<object>** y el menos utilizado **<embed>**, fueron muy útiles a principios del desarrollo web, ya que permitían a los desarrolladores incorporar contenido enriquecido en páginas web como video, animaciones, etc. Desde entonces, pasaron de moda debido a muchos problemas, incluidos el acceso, la seguridad, el tamaño del archivo, entre otros. En la actualidad la mayoría de los dispositivos móviles ya no son compatibles con estos complementos, y el soporte de escritorio está camino a desaparecer.
Finalmente, el elemento **<iframe>**, junto con otras formas de incrustación de contenido, como **<canvas>**, **<video>**, etc., proporciona una forma de insertar un documento web entero dentro de otro.

Creación de tablas

Las tablas se definen a través del tag **<table>**, y se complementan con otros tags que permiten crear sus celdas internas **<tr>**, los encabezados **<th>** y la inserción de datos en sus celdas **<td>.**

Ejemplo:

```
<table style =”width:50%”>
  <tr>
    <th>Nombre</th>
    <th>Apellido</th>
    <th>Edad</th>
  </tr>
  <tr>
    <td>Fernando</td>
    <td>Perez</td>
    <td>41</td>
  </tr>
<tr>
    <td>Julio</td>
    <td>Conte</td>
    <td>20</td>
  </tr>
</table>
```

Si ejecutamos el ejemplo anterior se mostrará una tabla al 50% de la página, sin bordes, con el encabezado en negrita y centrado, y el contenido de cada celda alineado por defecto sobre la izquierda.

Para agregar un borde debemos agregar lo siguiente a la primer línea donde definimos el ancho de la tabla:

```
<table style=”border: 1px solid black; width:50%”>
```

Esto mostrará el borde externo de la tabla.

Para que toda la tabla y sus celdas tengan un borde bien marcado, debemos recurrir a las propiedades CSS, que nos permitirán definir todos sus bordes de una sola vez.

Todo aquello que hace a la alineación y formateo de las tablas se aplica con CSS. Lo veremos en el capítulo de CSS.

Algunos atributos de tabla:

rowspan: permite agrupar varias celdas.

```
<th rowspan=”N”>Título de la fila</th>
<td>contenido de la fila</td>
```

En este caso, N corresponde al número de celdas que vamos a agrupar dentro del tag **table data <td>.**

colspan: permite agrupar el contenido de las columnas de una tabla.

```
<th>Nombre y apellido</th>
<th colspan="2">Correo electrónico</th>
```

En este caso la columna correo electrónico ocupará dos columnas.

Es posible anidar tablas, esto es una tabla dentro de otra, aunque no es lo más recomendable, porque se obtiene un marcado más confuso y menos accesible para los usuarios que usan lectores de pantalla, pero, a veces, puede ser necesario, si, por ejemplo, deseamos importar contenido de forma sencilla desde otras fuentes.

scope: este atributo se puede agregar al elemento **<th>** para indicar a los lectores de pantalla exactamente para qué celdas es el encabezado.

id y **headers:** son una alternativa del atributo **scope**, para crear asociaciones entre encabezados y celdas. Esto se logra añadiendo un **id** único a cada elemento **<th>** y **headers** a cada elemento **<td>.** Cada atributo **headers** debe contener una lista de **id** de todos los elementos **<th>** que actúan como encabezado de esa celda, separados por espacios.

Esto le da a la tabla una definición limpia de la posición de cada celda en la tabla definida por los encabezados de cada columna y fila de la que forma parte. Para que funcione bien, la tabla necesita tanto encabezados de columna como encabezados de fila.

FORMULARIOS EN HTML5

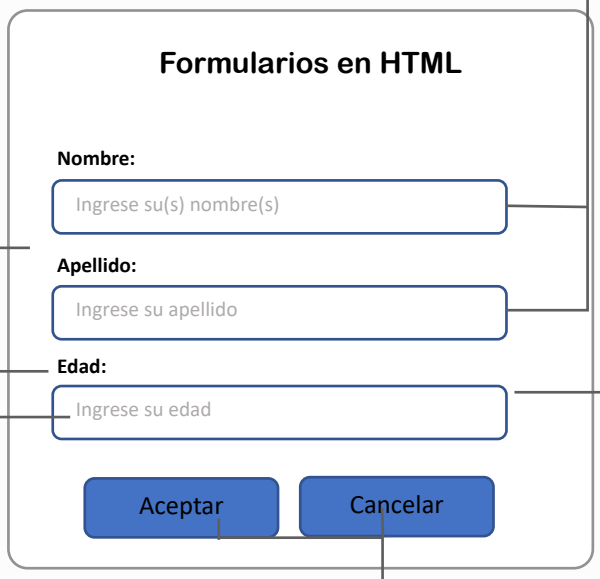
Los formularios se utilizan para recopilar información del usuario. A través de estos, los sitios nos permiten, entre otras funcionalidades, ingresar datos, como registro de usuarios, formularios de contacto, alta de productos en un sistema, etc.

Los formularios se definen utilizando el elemento **<form>,** con sus etiquetas de apertura y cierre.

Componentes de los formularios web

<form>: contiene todos los elementos **input type** que conformarán un formulario web para la carga de datos. Contiene una serie de atributos importantes que ya describiremos.

<label for> permite especificar una etiqueta, o label, que describa el tipo de campo con el que está relacionada. Se utiliza para agregar una descripción breve. En algunos casos, esta etiqueta, es reemplazada directamente por el atributo **placeholder**. Esto permite ahorra espacio en pantalla. Sobre todo para los forms que deben visualizarse en los dispositivos móviles.



<input type="text"> permite el ingreso de texto, números tipeados o seleccionados desde una lista. Posee una serie de atributos, como **name, id, placeholder** y **disabled**, que permiten darle un contexto único y tabajar su comportamiento en general.

Algunos elementos, como **input type** (number, tel, email, url, month, year, date, range), poseen, según el navegador web donde se ejecuten, la posibilidad de acceder a funciones determinadas, como un control spin, un calendario para seleccionar fechas, y otras tantas funciones.

Atributo **placeholder**: permite especificar un texto descriptivo que guía al usuario sobre el dato a cargar en el **input type**. Este atributo solo puede ser utilizado en aquellos elementos **input type** que permitan ingresar datos, como los de tipo **text, number**, o derivados de ambos. Los **input type checked, radio button, button,** no le dan soporte.

Los elementos **<input type="button">** pueden ser reemplazados también por el elemento HTML **<button>**, aunque el uso de estos botones mediante **input type** permite especificar que el tipo de botón sea **submit** (para guardar o trasmitir los datos del formulario directamente) o **reset** (para cancelar, limpiar y permitir cargar nuevamente los datos).

El elemento **form** y todos los componentes (o **input types**) declarados pueden ser modificados en el aspecto estético como también en su comportamiento, a través de una hoja de estilos CSS.

Nombraremos algunas características y los atributos principales del elemento **form**:

autocomplete: por defecto, el atributo viene desactivado (**off**). Seteando el **autocomplete** en **on**, se activará la función de autocompletado.

method: es el método HTTP que utiliza el navegador web para enviar los datos cargados en el formulario. Soporta los métodos **POST** y **GET**.

name: el formulario debe tener un nombre único. Este nombre debe especificarse en el atributo **id="nombredelform".**

novalidate: este atributo permite especificar si el formulario debe validar o no los datos que van a ser enviados.

target: sirve para especificar un nombre o una página de destino, que mostrará al usuario un mensaje o una respuesta determinada luego de enviar el formulario. Los valores aceptados son: **_self, _blank, _parent, _top.**

Input types

Todos los componentes que corresponden al ingreso o validación de datos que conforman el contenido de un formulario web se trabajan a través de los componentes denominados **input types**. Hoy en día disponemos de una gran variedad de **input types** diferentes, que con seguridad pueden ajustarse a nuestras necesidades.

Veamos una descripción de las más utilizadas:

input type	Caracterísicas	
text	Todo tipo de caracteres. Puede usarse como una caja de texto común y limitar la cantidad de caracteres a ingresar desde su atributo **maxlength**.	
number	Sólo acepta el ingreso de números. Por defecto los números que acepta son con decimales; en algunos navegadores se activa el control **spin**, para incrementar o decrementar el número en escala de 1.	
checkbox	Permite tildarlo o destildarlo. Por defecto maneja los estados **1** o **0** a través de su atributo **value**.	
color	Permite seleccionar un color desde una paleta de colores.	
date	Permite especificar una fecha utilizando año, mes y día. En algunos navegadores web aparece un calendario gráfico, para seleccionar la fecha de manera visual.	

email	Permite ingresar solo direcciones de correo electrónico.
month	Permite especificar un mes y un año mediante un input similar a **date.**
password	Crea una máscara propia de un password, y todo lo que se escriba no podrá ser visible de manera simple al ojo del usuario.
radio	Permite crear uno o más **radio buttons**, para seleccionar de manera unívoca una opción entre varias disponibles.
range	Es similar a **input type=" number",** puede limitar el contenido por ingresar entre dos valores: por ejemplo entre "0 y 9".
reset	Se comporta como un botón y permite borrar, de forma completa, todos los datos ingresados en un formulario.
search	Este componente genera una caja de texto específica para realizar búsquedas.
submit	Se comporta como el botón de enviar datos cargados en un formulario, invocando el valor cargado en el atributo **form action**, del elemento **<form>.**
tel	Permite ingresar teléfonos limitando los valores entre "**0 y 9**", *****, **#** y **-**.
time	Permite ingresar horas, minutos y segundos, limitando los valores entre las **0:00:00** y **23:59:59** horas.
url	Permite ingresar una dirección de una página web.
week	Permite ingresar el número de una semana del año, limitando sus valores entre **1** y **54.**
textArea	Permite escribir muchos caracteres en múltiples líneas. Los atributos **cols** y **rows** manejan el alto y el ancho que ocupa este control en pantalla. También soporta el límite de caracteres con el atributo **maxlength.**
button	Puede reemplazar a **input type="button"** solo cuando la acción por ejecutar para enviar datos del formulario debe invocar a una función de JavaScript.
file	Permite seleccionar un archivo desde la plataforma operativa. Por lo general accede, según el sistema operativo, a la estructura de archivos que SO dispone para ser visualizada desde una web.
hidden	Establece un componente **input type="text"** oculto. Se pueden almacenar datos específicos, como ser un ID de transacción operativa, invisible al usuario, pero con posibilidad de grabar su valor al confirmar el form.

image	Permite definir una imagen específica que reemplace el **input type submit.** A diferencia de una imagen clásica, este input cuenta con los mismos atributos que **submit**.
datalist	Representa la lista de elementos como sugerencias cuando se llena un campo **input.** Se puede usar un atributo **list** en un elemento **input** para enlazar a un campo de ingreso específico con un elemento **datalist** determinado.
autofocus	Permite especificar que una parte del formulario debe tener foco para ingresar información cuando se carga la página. Sólo un elemento del formulario en un documento puede tener el atributo **autofocus**, que es de tipo boolean. Este atributo puede ser aplicado a los elementos **input, button, select** y **textarea.**

Código del formulario visto más arriba (sin aplicar CSS):

```
<form>
        <h2>Formularios en HTML</h2>
        <label for="nombre">Nombre:</label><br>
        <input type="text" name="nombre" id="nombre" placeholder="Ingrese su(s) nombre(s)"/><br>
        <label for="apellido">Apellido:</label><br>
        <input type="text" name="apellido" id="apellido" placeholder="Ingrese su apellido"/><br>
        <label for="edad">Edad:</label><br>
        <input type="number" name="edad" id="edad" placeholder="Ingrese su edad"/><br><br>
        <input type="submit" name="submit" id="submit" value="Aceptar"/>
        <input type="reset" name="cancel" id="cancel" value="Cancelar"/>
</form>
```

Sintaxis para validación restringida:

- **'required':** este atributo en los elementos **<input>, <select>** y **<textarea>** indica que se debe ingresar algún dato.
- **'pattern':** en el elemento **<input>** restringe el valor para que concuerde con una expresión regular específica.
- **'min'** y **'max':** del elemento **<input>** restringen los valores máximos y mínimos que pueden ser ingresados.
- **'step':** del elemento **<input>**, cuando se usa en combinación con los atributos **min** y **max**, restringe la granularidad de los valores ingresados. Un valor que no se corresponda con un valor permitido no será validado.
- **'maxlength':** de los elementos **<input>** y **<textarea>** restringe el máximo número de caracteres que el usuario puede ingresar.
- Los valores **url** y **email** para **type** restringen el valor para una URL o dirección de correo válida respectivamente.

Además, podemos prevenir la validación restringida especificando el atributo **novalidate** en el elemento **<form>**, o el atributo **formnovalidate** en el elemento **<button>** y en el elemento **<input>.** Estos atributos indican que el formulario no será validado cuando se envíe.

WEB DEVELOPMENT

INNOVARE SOFTWARE