

Programación Web Desde Cero

Introducción al desarrollo web & HTML

¡Hola! 🖐️

En esta introducción, comenzaremos presentándote el mundo del desarrollo web, para luego adentrarnos en los conceptos básicos de HTML que nos llevarán a construir tu primera página web.

En el mundo digital y conectado de hoy, las habilidades en tecnología de la información son más valiosas que nunca. Ya sea que busques una ventaja competitiva en tu carrera, un cambio en tu trayectoria profesional o simplemente una nueva habilidad para cultivar, nuestro curso es el punto de partida perfecto para acceder a un mundo de oportunidades.

¿Qué es el desarrollo web?

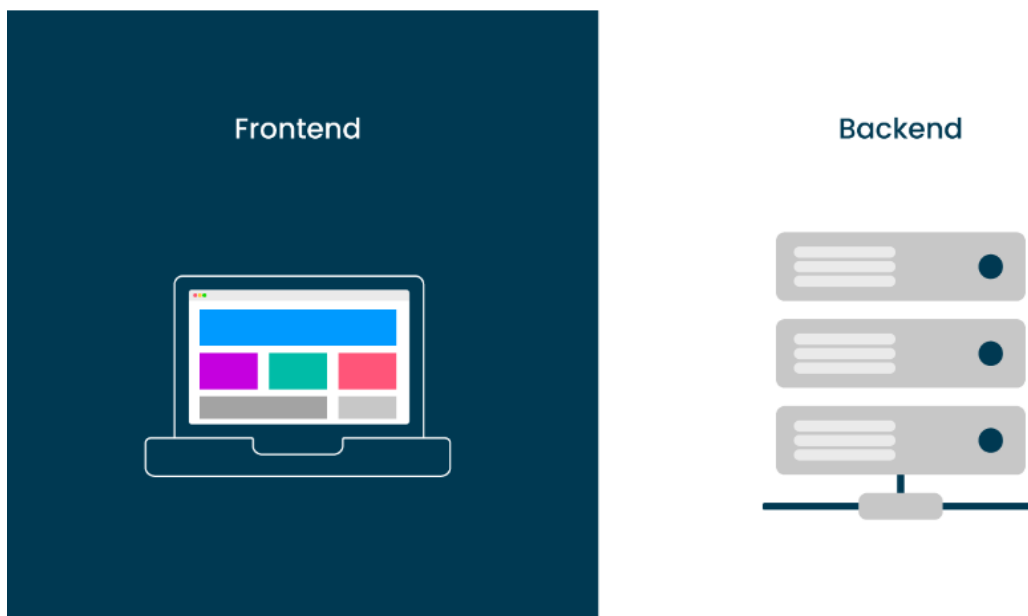
El **desarrollo web** es una disciplina que se ha vuelto indispensable en la era digital en la que vivimos. **Consiste en la creación y el mantenimiento de páginas y aplicaciones web a los que se puede acceder a través de internet.**

Su objetivo principal es proporcionar al usuario una experiencia **fluida y atractiva** al interactuar con la página web o aplicación. Para lograr esto, **es necesario contar con una amplia gama de habilidades técnicas y creativas para poder crear la estructura, diseño y funcionalidad de la misma.**

Esto abarca una amplia gama de aspectos, incluidos la programación, el diseño gráfico, la arquitectura de la información y la interacción con el usuario.

El desarrollo web se divide en dos categorías principales: desarrollo **frontend** (lado del cliente) y desarrollo **backend** (lado del servidor).

- El **desarrollo frontend** se refiere a la **creación de la parte visual y funcional del sitio web o aplicación** que los usuarios **pueden ver** y con la que **pueden interactuar**. Esto incluye la disposición de elementos visuales, la programación de botones y otros elementos interactivos, y la optimización del rendimiento para que los tiempos de carga sean rápidos. Los desarrolladores frontend también se encargan de garantizar que la web sea compatible con diferentes navegadores y dispositivos móviles.
- El **desarrollo backend**, es la parte detrás de escena del sitio web que maneja **la lógica, el almacenamiento de datos y la comunicación con el servidor**. Los desarrolladores backend trabajan con lenguajes de programación como Python, PHP, Ruby, Java y bases de datos para **crear y gestionar la funcionalidad del sitio**.
- El desarrollador **fullstack**, tiene conocimientos tanto en frontend como en backend. Pueden producir una aplicación web completa, desde la interfaz de usuario hasta la lógica del servidor y la base de datos.



El desarrollo web es un campo emocionante y en constante evolución. Ya sea que estés interesado en el desarrollo frontend o backend, hay muchas oportunidades sorprendentes disponibles para los desarrolladores web en todo el mundo.

Principios básicos del desarrollo web

Existen principios básicos del desarrollo web que son fundamentales para crear sitios exitosos y atractivos. Estos principios incluyen la navegabilidad, interactividad y arquitectura de la información.

Veamos de qué se trata cada uno de ellos:

- La **navegabilidad** se refiere a **qué tan fácil es para los usuarios moverse y encontrar información** en un sitio web. Un diseño de navegación claro y fácil de usar es esencial para que los usuarios puedan encontrar rápidamente lo que están buscando y tengan una experiencia satisfactoria.
- La **interactividad** es la **capacidad de un sitio web para interactuar con los usuarios y ofrecerles una experiencia dinámica y personalizada**. Esto puede incluir animaciones, formularios, comentarios en tiempo real y otros elementos que permiten a los usuarios participar activamente en el sitio.
- La **arquitectura de la información** se refiere a **cómo se organiza y presenta la información en un sitio web**. Esto incluye la estructuración de contenidos, la categorización y el etiquetado de estos, y la creación de una jerarquía lógica que facilite a los usuarios encontrar y entender la información que necesitan.

Editores de código

Para que todo esto mencionado anteriormente cobre vida y sea llevado a la práctica, es necesario utilizar un **editor de código**.

Los **editores de código** son herramientas que **permiten a los programadores escribir y editar código de programación**. Estas herramientas proporcionan características como resaltado de sintaxis, autocompletado, atajos, sugerencias de

código y depuración para ayudar a los desarrolladores a escribir código de manera más eficiente y efectiva.

Para este curso, la herramienta que vamos a utilizar es **Visual Studio Code**. Es el lugar donde vas a escribir en distintos lenguajes para crear tus proyectos del curso:

Lo puedes descargar a continuación:

👉 Descargar – Visual Studio Code

Para realizar la instalación puedes revisar uno de los siguientes tutoriales en función del sistema operativo que tengas instalado en tu computadora:

- **Windows** → [Cómo instalar el Visual Studio Code en Windows](#)
 - **Mac** → [Cómo instalar el Visual Studio Code en Mac](#)
 - **Linux** → [Cómo instalar el Visual Studio Code en Linux](#)
-

Introducción a HTML

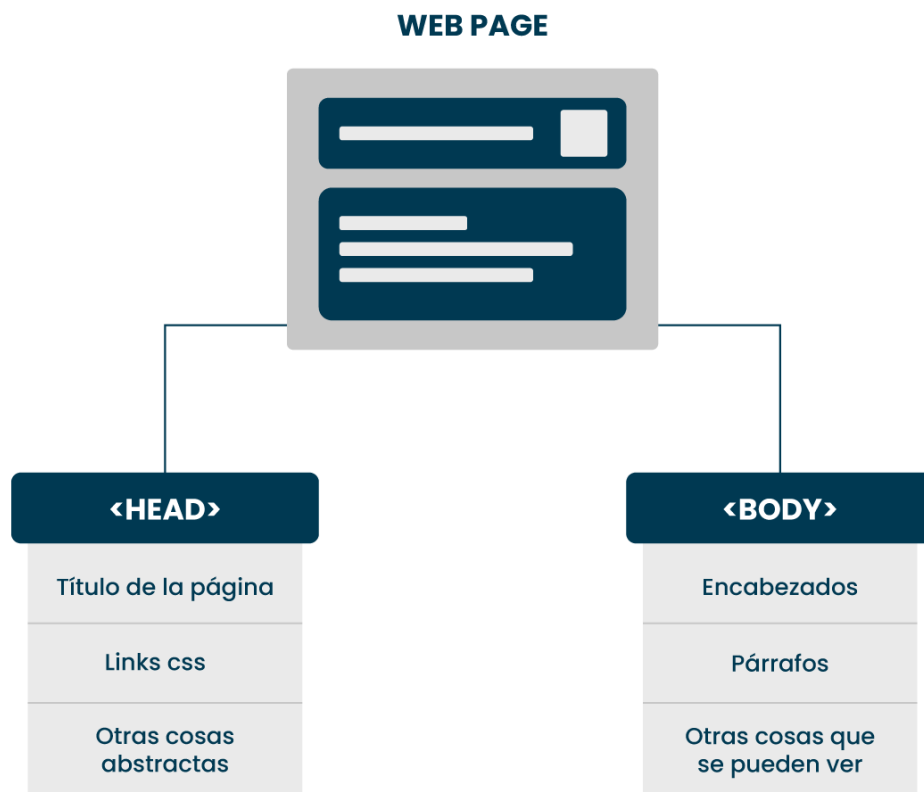
Ahora que ya sabes los conceptos básicos de la programación web, vamos a empezar a adentrarnos en **lenguaje fundamental** en el mundo del desarrollo web. Este es el **HTML** (Hypertext Markup Language)

Estructura del HTML

HTML (Hypertext Markup Language) es el lenguaje de marcado estándar usado para crear la estructura de un sitio web. **Consiste en una serie de elementos representados por etiquetas, que le indican al navegador cómo mostrar el contenido.**

Ahora te mostraremos en el siguiente video cómo se crea una estructura básica de HTML:

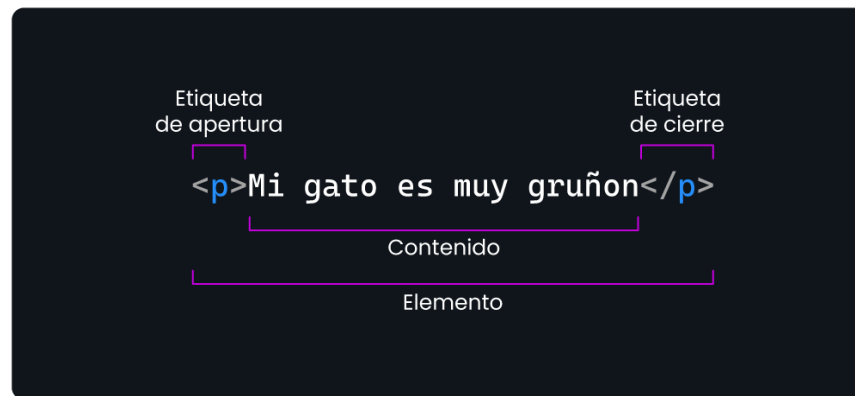
 [Estructura básica HTML | Introducción al Desarrollo Web & HTML | Egg](#)



💡 Te facilitamos el enlace de la página que mencionamos en el video sobre la organización: "[World Wide Web Consortium](#)". Sugerimos que guardes los links bibliográficos que vamos mencionando para que construyas tu propia fuente de recursos.

Anatomía de un elemento HTML

El lenguaje HTML está compuesto por elementos. Estos elementos tienen una etiqueta de apertura, una de cierre y contenido, como se muestra a continuación:



Las partes principales de un elemento html son:

1. **La etiqueta de apertura:** consiste en el nombre de la etiqueta (en este caso, p), encerrado por paréntesis angulares (`<` `>`) de apertura y cierre. Establece dónde comienza o empieza a tener efecto la etiqueta, en este caso, dónde es el comienzo del párrafo.
2. **La etiqueta de cierre:** es igual que la etiqueta de apertura, excepto que incluye una barra de cierre (`/`) antes del nombre de la etiqueta. Establece dónde termina la etiqueta, en este caso dónde termina el párrafo.
3. **El contenido:** este es el contenido de la etiqueta, que en este caso es solo texto.
4. **El elemento:** la etiqueta de apertura, más la etiqueta de cierre, más el contenido equivale al elemento.

Etiquetas

Es importante entender cómo funcionan las etiquetas HTML en términos de apertura y cierre. La mayoría de las etiquetas HTML tienen una etiqueta de apertura y una de cierre, que definen el inicio y el final de un elemento. Estas etiquetas ayudan al navegador a comprender la estructura y jerarquía del contenido en la página.

La sintaxis para una etiqueta de apertura es el nombre de la etiqueta rodeado de paréntesis angulares, como **<etiqueta>**. La etiqueta de cierre tiene una sintaxis similar, pero con una barra inclinada hacia delante antes del nombre de la etiqueta: **</etiqueta>**. Lo que se encuentra encerrado entre las etiquetas de apertura y cierre es el **contenido del elemento**.

Aquí hay algunos ejemplos de elementos HTML con etiquetas de apertura y cierre:

- **Encabezados:** Hay seis niveles de encabezados, desde <h1> (el más grande) hasta <h6> (el más pequeño).

<h1>Encabezado 1</h1>

<h2>Encabezado 2</h2>

...

<h6>Encabezado 6</h6>

- **Párrafo:**

<p>Este es un párrafo.</p>

- **Lista sin numerar:**

Elemento de lista 1

Elemento de lista 2

- **Lista numerada:**

Elemento de lista 1

```
<li>Elemento de lista 2</li>  
</ol>
```

Etiquetas sin cierre

Algunas etiquetas HTML no requieren una etiqueta de cierre separada. En su lugar, se cierran dentro de la misma etiqueta, a menudo porque no tienen contenido que encerrar.

Ejemplos de etiquetas de cierre automático incluyen:

- **Salto de línea:**

```
<br />
```

- **Imagen:**

```

```

- **Entrada:**

```
<input type="text" name="ejemplo" />
```

📌 Para ver toda la lista de elementos HTML que existen podemos recurrir a la [página de W3 Schools](#) en la que verás los distintos elementos y la explicación de cómo se usa cada uno de ellos.

Checklist de buenas prácticas

1. **Siempre cierra tus etiquetas:** Asegúrate de que cada etiqueta de apertura tenga una etiqueta de cierre correspondiente, a menos que sea una etiqueta de cierre automático. **Cerrar correctamente tus etiquetas ayuda a mantener la estructura y jerarquía de tu documento HTML.**



```
<p>Este es un párrafo con algo de <strong>texto en negrita</strong> en él.</p>
```



```
<P>Este es un párrafo con algo de <strong>texto en negrita en él.
```

2. **Anida las etiquetas correctamente:** Al trabajar con elementos anidados, asegúrate de que las etiquetas internas estén cerradas antes de cerrar las etiquetas externas. Un anidamiento incorrecto puede llevar a errores inesperados de representación o validación. Por ejemplo, la forma correcta de anidar una etiqueta **** dentro de una etiqueta **<p>** es:



```
<p>Este es un párrafo con algo de <strong>texto en negrita</strong> en él.</p>
```



```
<p>Este es un párrafo con algo de <strong>texto en negrita en él.</p></strong>
```

3. **Usa nombres de etiquetas en minúsculas:** Aunque HTML no distingue entre mayúsculas y minúsculas, es una buena práctica usar nombres de etiquetas en minúsculas para mantener la coherencia y legibilidad.



```
<p>Este es un párrafo con algo de <strong>texto en negrita</strong> en él.</p>
```



```
<P>Este es un párrafo con algo de <Strong>texto en negrita</Strong> en él.</P>
```

4. **Indentar elementos anidados:** Para mejorar la legibilidad de tu código HTML, es una buena práctica indentar elementos anidados. Esto te ayuda a ver la estructura y jerarquía de tu documento más fácilmente.



```
<div>  
  <p>Este es un párrafo con algo de <strong>texto en negrita</strong> en él.</p>  
</div>
```



```
<div>  
<p>Este es un párrafo con algo de <strong>texto en negrita</strong> en él.</p>  
</div>
```

Resumen

¡Felicidades!

Esperamos que esta información te haya sido de utilidad y que te lleves para repasar:

- Qué es el desarrollo web y los términos asociados al mismo
- Qué herramientas se utilizan para trabajar y escribir código
- Qué es HTML y cuáles son los elementos principales que lo conforman