

## Origen de Internet

El origen de Internet se remonta a la década de 1960 cuando el Departamento de Defensa de Estados Unidos comenzó a trabajar en un proyecto de investigación llamado ARPANET (Advanced Research Projects Agency Network). ARPANET fue diseñado para interconectar computadoras de diferentes instituciones de investigación y permitir la comunicación y el intercambio de información entre ellas.

En 1969, se estableció la primera conexión entre dos nodos de ARPANET, ubicados en la Universidad de California, Los Ángeles (UCLA) y el Instituto de Investigación de Stanford. Esta conexión inicial se realizó a través de una línea telefónica de baja velocidad.

A medida que ARPANET continuó expandiéndose, se desarrollaron nuevos protocolos y tecnologías para mejorar la eficiencia y confiabilidad de la red. En la década de 1970, se introdujo el protocolo de control de transmisión/protocolo de Internet (TCP/IP), que se convirtió en el estándar para la comunicación en la red. TCP/IP permitió la interconexión de redes y sentó las bases para el crecimiento y la evolución de Internet.

En la década de 1980, ARPANET se dividió en redes más pequeñas llamadas redes de área amplia (WAN) y se creó la National Science Foundation Network (NSFNET) en los Estados Unidos para conectar universidades y centros de investigación. A medida que más redes se unieron a NSFNET y se establecieron conexiones internacionales, Internet comenzó a tomar forma como una red global de comunicación.

### 1. Origen de la WEB

Fue en la década de 1990 cuando la World Wide Web, tal como la conocemos hoy, se desarrolló y se hizo accesible para el público en general.

## EVOLUCIÓN CONTINUA

La World Wide Web fue creada por Tim Berners-Lee, un científico de la computación británico, en 1989 mientras trabajaba en el CERN (Organización Europea para la Investigación Nuclear) en Suiza. Berners-Lee propuso un sistema de información que utilizaba hipervínculos para enlazar documentos a través de una red de computadoras. Su idea se basó en la combinación de varios elementos clave, como el protocolo de transferencia de hipertexto (HTTP), el lenguaje de marcado de hipertexto (HTML) y el sistema de direcciones de internet (URL).

En 1991, se lanzó la primera página web en el CERN, marcando el inicio de la web. En los años siguientes, la web experimentó un rápido crecimiento y se popularizó en todo el mundo a medida que más personas y organizaciones comenzaron a crear y acceder a sitios web.

Es importante destacar que la web es solo una parte de internet. La internet es una infraestructura global de redes interconectadas que permite la comunicación y el intercambio de información entre computadoras. La web es un servicio basado en internet que utiliza tecnologías específicas, como los protocolos mencionados anteriormente, para acceder y visualizar contenido a través de navegadores web.

## 2. Navegador

Un navegador web, también conocido como navegador de internet, es una aplicación de software que se utiliza para acceder y visualizar contenido en la World Wide Web. Los navegadores web permiten a los usuarios navegar por páginas web, seguir enlaces, ver imágenes, reproducir videos y realizar diversas acciones en línea.

Existen numerosos navegadores web disponibles en la actualidad, y algunos de los más populares son los siguientes:

## EVOLUCIÓN CONTINUA

- Google Chrome: Desarrollado por Google, es uno de los navegadores más utilizados a nivel mundial, conocido por su rapidez y amplia gama de características.
- Mozilla Firefox: Un navegador de código abierto que se enfoca en la privacidad y la personalización. Es conocido por su seguridad y su capacidad para admitir una amplia variedad de extensiones.
- Safari: El navegador web desarrollado por Apple, diseñado específicamente para los dispositivos de Apple, como Mac, iPhone y iPad. Es conocido por su rendimiento y su integración con el ecosistema de Apple.
- Microsoft Edge: El navegador desarrollado por Microsoft, que reemplazó a Internet Explorer. Está disponible para Windows y también tiene una versión basada en Chromium.
- Opera: Un navegador web que se destaca por su velocidad, seguridad y diversas características, como un bloqueador de anuncios incorporado y un modo de ahorro de datos.

### 3. Funcionamiento

Funciona mediante una combinación de tecnologías y protocolos que permiten la transferencia de datos entre servidores y clientes (navegadores web).

- Protocolo de transferencia de hipertexto (HTTP): El HTTP es el protocolo subyacente que permite la comunicación entre los servidores web y los navegadores. Cuando ingresas una dirección web en tu navegador (por ejemplo, <http://www.ejemplo.com>), el navegador envía una solicitud HTTP al servidor correspondiente.
- Lenguaje de marcado de hipertexto (HTML): El contenido de la web se crea utilizando HTML. Es un lenguaje de marcado que define la estructura y el formato de una página web, especificando los elementos como encabezados, párrafos, enlaces, imágenes, etc.

- **Servidores web:** Los servidores web son computadoras que almacenan y sirven contenido web. Cuando envías una solicitud HTTP a un servidor, este procesa la solicitud y envía una respuesta HTTP que contiene los archivos necesarios para mostrar la página web solicitada. Los archivos pueden incluir HTML, imágenes, hojas de estilo CSS, scripts JavaScript, etc.
- **Navegadores web:** Los navegadores web son aplicaciones que permiten a los usuarios acceder y visualizar contenido web. Cuando reciben una respuesta HTTP del servidor, los navegadores interpretan el código HTML y muestran la página web al usuario. También descargan y procesan otros archivos relacionados, como imágenes y scripts.
- **Hipervínculos:** Los hipervínculos son enlaces que conectan diferentes páginas y recursos en la web. Al hacer clic en un enlace, el navegador envía una nueva solicitud HTTP al servidor correspondiente y muestra la nueva página web o recurso vinculado.
- **URL (Uniform Resource Locator):** Una URL es la dirección que se utiliza para acceder a una página web o un recurso en particular en la web. Las URL están compuestas por diferentes partes, como el protocolo (por ejemplo, http://), el nombre de dominio (por ejemplo, www.ejemplo.com) y la ruta específica del recurso.

Resumiendo, la web funciona a través de una combinación de solicitudes y respuestas HTTP, lenguaje HTML para estructurar el contenido, servidores web para almacenar y entregar el contenido, y navegadores web para interpretar y mostrar las páginas web a los usuarios. Los hipervínculos y las URL permiten la navegación entre las diferentes páginas y recursos en la web.

#### 4. Arquitectura Cliente / Servidor

La arquitectura cliente-servidor es un modelo de diseño ampliamente utilizado en sistemas de comunicación, como en la web. En este modelo, el sistema está dividido en dos componentes principales: el cliente y el servidor, que se comunican entre sí para cumplir con sus respectivas funciones.

- **Cliente:** El cliente es la parte del sistema que interactúa directamente con el usuario final. Puede ser un software, una aplicación móvil o un navegador web. Su función principal es solicitar recursos o servicios al servidor y presentar los resultados al usuario.
- **Servidor:** El servidor es la parte del sistema que responde a las solicitudes del cliente y proporciona los recursos o servicios requeridos. Puede ser una computadora física o un conjunto de computadoras que ejecutan software especializado para servir solicitudes de los clientes.

El flujo de comunicación entre el cliente y el servidor en el modelo cliente-servidor es el siguiente:

- **El cliente envía una solicitud al servidor:** El cliente inicia una solicitud, como solicitar una página web o enviar datos para ser procesados, y la envía al servidor. La solicitud generalmente incluye información como el tipo de recurso solicitado, los parámetros de la solicitud y otros datos relevantes.
- **El servidor procesa la solicitud:** El servidor recibe la solicitud del cliente y la procesa. Esto puede implicar acceder a una base de datos, realizar cálculos o ejecutar cualquier otra operación necesaria para cumplir con la solicitud.
- **El servidor envía la respuesta al cliente:** Una vez que el servidor ha procesado la solicitud, genera una respuesta que contiene los

## EVOLUCIÓN CONTINUA

resultados o los recursos solicitados. La respuesta puede incluir datos, archivos, metadatos o cualquier otra información relevante.

- El cliente recibe y muestra la respuesta: El cliente recibe la respuesta del servidor y la interpreta para presentar los resultados al usuario final. Esto puede implicar mostrar una página web, procesar los datos recibidos o realizar otras acciones según la naturaleza de la respuesta.

Es importante destacar que en el modelo cliente-servidor, el servidor es responsable de gestionar los recursos y proporcionar los servicios solicitados, mientras que el cliente se encarga de interactuar con el usuario y presentar los resultados de manera adecuada.

Este modelo de arquitectura permite una división clara de responsabilidades y facilita la escalabilidad y el mantenimiento de los sistemas, ya que los componentes del cliente y del servidor pueden actualizarse o modificarse de forma independiente sin afectar al otro.

## 5. Desarrollo FrontEnd vs BackEnd

### Desarrollo Frontend:

El desarrollo frontend se centra en la parte del software que se ejecuta en el lado del cliente, es decir, en el navegador web o en una aplicación móvil. Los desarrolladores frontend se encargan de crear la interfaz de usuario con la que los usuarios interactúan directamente. Algunas características clave del desarrollo frontend son:

- Interfaz de usuario (UI): Los desarrolladores frontend se ocupan de diseñar y desarrollar la apariencia visual de una aplicación. Esto implica crear diseños, utilizar lenguajes de marcado como HTML para estructurar el contenido y utilizar hojas de estilo CSS para estilizar y dar formato a los elementos.

## EVOLUCIÓN CONTINUA

- Experiencia de usuario (UX): El desarrollo frontend también se centra en la experiencia de usuario, es decir, en cómo los usuarios interactúan con la aplicación. Los desarrolladores frontend se esfuerzan por crear interfaces intuitivas y responsivas para garantizar una experiencia agradable al usuario.
- Lenguajes y tecnologías: Los desarrolladores frontend suelen utilizar lenguajes de programación como JavaScript, que les permite agregar interactividad y dinamismo a las aplicaciones. También hacen uso de bibliotecas y frameworks populares como React, Angular o Vue.js para facilitar el desarrollo y la gestión del estado de la aplicación.

**Desarrollo Backend:**

El desarrollo backend se enfoca en la parte del software que se ejecuta en el servidor. Los desarrolladores backend se ocupan de implementar la lógica empresarial, gestionar la base de datos y proporcionar los servicios y recursos necesarios para que funcione la aplicación. Algunos aspectos clave del desarrollo backend son:

- Lógica de negocio: Los desarrolladores backend implementan la lógica de negocio de una aplicación. Esto implica procesar datos, realizar cálculos, interactuar con sistemas externos y garantizar la seguridad y la integridad de los datos.
- Base de datos: Los desarrolladores backend trabajan con bases de datos para almacenar y recuperar información. Diseñan y gestionan esquemas de bases de datos, crean consultas y aseguran el acceso eficiente y seguro a los datos.
- APIs y servicios web: El desarrollo backend implica la creación de APIs (Interfaces de Programación de Aplicaciones) y servicios web que permiten a las aplicaciones frontend comunicarse con el servidor. Esto puede incluir la definición de puntos de acceso, el

manejo de solicitudes y respuestas, y la implementación de autenticación y autorización.

- Rendimiento y escalabilidad: Los desarrolladores backend se preocupan por la eficiencia y el rendimiento de las aplicaciones, garantizando que puedan manejar grandes volúmenes de tráfico y que sean escalables para adaptarse al crecimiento.

## 6. HTML

HTML (HyperText Markup Language) es el lenguaje de marcado estándar utilizado para crear y estructurar el contenido de las páginas web. Es el componente fundamental de la web y se utiliza para definir la estructura y el significado del contenido de una página web.

- Marcado de elementos: HTML utiliza etiquetas o elementos para marcar diferentes partes del contenido. Por ejemplo, la etiqueta `<h1>` se utiliza para indicar un encabezado de nivel 1, la etiqueta `<p>` se utiliza para indicar un párrafo y la etiqueta `<img>` se utiliza para insertar una imagen.
- Estructura de árbol: Los elementos HTML se organizan en una estructura jerárquica de árbol, lo que significa que algunos elementos están contenidos dentro de otros. El elemento raíz es `<html>`, seguido por `<head>` y `<body>`, y luego otros elementos como encabezados, párrafos, listas, enlaces, etc.
- Atributos: Los elementos HTML pueden tener atributos que proporcionan información adicional sobre el elemento. Los atributos se agregan a las etiquetas y tienen un nombre y un valor. Por ejemplo, el atributo `src` en la etiqueta `<img>` especifica la fuente de la imagen.
- Hipervínculos: HTML permite crear hipervínculos entre páginas web utilizando la etiqueta `<a>`. Al hacer clic en un enlace, el navegador web sigue el enlace y carga la página vinculada.



## EVOLUCIÓN CONTINUA

- Multimedia: HTML también permite la inclusión de contenido multimedia, como imágenes, audio y video. Esto se logra mediante etiquetas específicas como `<img>`, `<audio>` y `<video>`.
- Versiones de HTML: A lo largo del tiempo, HTML ha evolucionado y se han lanzado diferentes versiones. HTML5 es la versión más reciente y cuenta con características y funcionalidades avanzadas, como soporte para video y audio integrado, gráficos en 2D y 3D, almacenamiento local y mucho más.

HTML se combina con otros lenguajes y tecnologías, como CSS (Cascading Style Sheets) para aplicar estilos y diseño a las páginas web, y JavaScript para agregar interactividad y funcionalidad dinámica. En conjunto, estas tecnologías forman la base para crear páginas web interactivas y atractivas.

## 7. Estructura Básica

Un documento HTML consta de una estructura básica que incluye una declaración, la sección `<html>`, y dentro de ella, las secciones `<head>` y `<body>`.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Título de la página</title>
```

```
  </head>
```

```
  <body>
```

```
    Contenido de la página web.
```

```
  </body>
```

```
</html>
```

- **<!DOCTYPE html>:** Esta es una declaración que indica al navegador que el documento es HTML5, la versión más reciente de HTML.
- **<html>:** Es el elemento raíz que contiene todo el contenido de la página web.
- **<head>:** Es la sección que contiene metadatos y enlaces a archivos externos, como hojas de estilo CSS y scripts JavaScript. No se muestra directamente en la página, pero contiene información importante para el navegador y los motores de búsqueda.
- **<title>:** Es una etiqueta que se encuentra dentro de la sección <head>. Define el título de la página que se mostrará en la barra de título del navegador.
- **<body>:** Es la sección principal que contiene el contenido visible de la página web. Aquí es donde se agregan elementos como encabezados, párrafos, imágenes, enlaces y otros elementos.

## 8. Composición de un elemento

Un elemento HTML está compuesto por varias partes, que se agrupan dentro de una etiqueta y pueden incluir contenido y atributos.

```
<etiqueta atributo1="valor1" atributo2="valor2">Contenido</etiqueta>
```

- **Etiqueta:** Es el nombre del elemento HTML y se especifica entre los símbolos < y >. Por ejemplo, <p> es la etiqueta para un párrafo, <h1> para un encabezado de nivel 1, <img> para una imagen, etc. Existen numerosas etiquetas HTML que se utilizan para diferentes tipos de contenido y funcionalidades.
- **Atributos:** Los atributos proporcionan información adicional sobre el elemento y se especifican dentro de la etiqueta de apertura. Los atributos tienen un nombre y un valor, y se escriben en el formato

## EVOLUCIÓN CONTINUA

nombre="valor". Por ejemplo, `src="imagen.jpg"` es un atributo para especificar la ruta de una imagen. Los atributos pueden ser opcionales y dependen del elemento específico que estés utilizando.

- **Contenido:** El contenido de un elemento HTML se encuentra entre la etiqueta de apertura y la etiqueta de cierre. Puede ser texto, otros elementos HTML o una combinación de ambos. Por ejemplo, en `<p>Este es un párrafo.</p>`, "Este es un párrafo." es el contenido del elemento de párrafo.

Es importante tener en cuenta que algunos elementos HTML no requieren contenido y se denominan elementos vacíos o auto-cerrados. Estos elementos no tienen una etiqueta de cierre y pueden tener atributos opcionales. Por ejemplo, `` es una etiqueta de imagen que no tiene contenido y se cierra automáticamente con el símbolo `/>`.

## 9. Etiqueta básicas

Además de las etiquetas que vimos anteriormente que dan la estructura de un documento HTML, existen muchas otras etiquetas con las que trabajamos.

### 9.1.H

La etiqueta `<h>` en HTML se utiliza para definir los encabezados o títulos de una página o sección. Hay diferentes niveles de encabezado, desde `<h1>` hasta `<h6>`, donde `<h1>` es el de mayor importancia y `<h6>` es el de menor importancia. Estas etiquetas de encabezado se utilizan para proporcionar una estructura y jerarquía al contenido de la página.

```
<h1>Título de nivel 1</h1>
```

```
<h2>Título de nivel 2</h2>
```

```
<h3>Título de nivel 3</h3>
```

```
...
```

```
<h6>Título de nivel 6</h6>
```

Cada etiqueta de encabezado indica un nivel diferente de importancia y tamaño visual. Por ejemplo, `<h1>` tiene un tamaño de fuente más grande y se considera el encabezado principal, mientras que `<h6>` tiene un tamaño de fuente más pequeño y se utiliza para encabezados de menor importancia.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Ejemplo de encabezados</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>Título principal de la página</h1>
```

```
    <h2>Sección 1</h2>
```

```
    <p>Contenido de la sección 1...</p>
```

```
    <h2>Sección 2</h2>
```

```
    <p>Contenido de la sección 2...</p>
```

```
    <h3>Subsección 2.1</h3>
```

```
    <p>Contenido de la subsección 2.1...</p>
```

```
<h3>Subsección 2.2</h3>
```

```
<p>Contenido de la subsección 2.2...</p>
```

```
</body>
```

```
</html>
```

En este ejemplo, se utilizan etiquetas de encabezado para proporcionar una estructura jerárquica al contenido de la página. `<h1>` se usa como el título principal de la página, `<h2>` se utiliza para las secciones principales, y `<h3>` se usa para las subsecciones dentro de las secciones principales.

Recuerda que el uso adecuado de las etiquetas de encabezado ayuda a mejorar la accesibilidad y la legibilidad de tu página web, proporcionando una estructura clara y organizada para los usuarios y los motores de búsqueda.

## 9.2.P

La etiqueta `<p>` en HTML se utiliza para definir un párrafo de texto. Proporciona una manera de agrupar un bloque de texto en un solo párrafo.

```
<p>Texto del párrafo</p>
```

Dentro de la etiqueta `<p>`, puedes agregar cualquier contenido de texto, como palabras, frases, enlaces u otros elementos HTML.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Ejemplo de párrafos</title>
```

```
</head>
```

```
<body>
```

```
<h1>Título principal</h1>
```

`<p>Este es un párrafo de texto. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut nec faucibus turpis, et varius dolor.</p>`

`<p>Este es otro párrafo. Pellentesque non imperdiet velit, ac cursus elit. Nullam a neque id mi ullamcorper euismod.</p>`

`</body>`

`</html>`

En este ejemplo, se utilizan dos etiquetas `<p>` para crear dos párrafos diferentes. Cada párrafo contiene texto que se muestra como un bloque separado en la página web.

La etiqueta `<p>` es útil para separar y agrupar contenido de texto en párrafos individuales, lo que facilita la legibilidad y el formato de tu contenido. Además, los navegadores web suelen aplicar estilos predeterminados a los párrafos, como el espacio antes y después del párrafo, para mejorar la visualización y la presentación del texto.

## 10.HR

La etiqueta `<hr>` en HTML se utiliza para insertar una línea horizontal en una página web. Esta etiqueta se usa para separar visualmente secciones o elementos de contenido en la página.

`<hr>`

La etiqueta `<hr>` es una etiqueta de autocierre, lo que significa que no tiene una etiqueta de cierre correspondiente. Se utiliza sola como un elemento independiente para crear una línea horizontal.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Ejemplo de línea horizontal</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>Título principal</h1>
```

```
    <p>Este es un párrafo de texto.</p>
```

```
    <hr>
```

```
    <h2>Subtítulo</h2>
```

```
    <p>Otro párrafo de texto.</p>
```

```
  </body>
```

```
</html>
```

En este ejemplo, se utiliza la etiqueta `<hr>` después del primer párrafo para insertar una línea horizontal que separa visualmente el contenido. Proporciona una división clara entre las secciones de la página web.

El estilo y el aspecto de la línea horizontal pueden variar según el navegador y los estilos CSS aplicados a la página web. Sin embargo, en general, la línea

horizontal se extiende horizontalmente a través de la página web y puede tener un espaciado o un estilo determinado.

Recuerda que la etiqueta `<hr>` es una forma sencilla de crear una línea horizontal y se utiliza comúnmente para separar visualmente elementos de contenido en una página web.

## 11.BR

La etiqueta `<br>` en HTML se utiliza para insertar un salto de línea o un salto de línea forzado en un texto o contenido. Esta etiqueta se utiliza para indicar que el texto debe continuar en una nueva línea.

Texto antes del salto de línea `<br>` Texto después del salto de línea

La etiqueta `<br>` es una etiqueta de autocierre, lo que significa que no tiene una etiqueta de cierre correspondiente. Se utiliza sola como un elemento independiente para crear un salto de línea.

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Ejemplo de salto de línea</title>
```

```
  </head>
```

```
  <body>
```

```
    <h1>Título principal</h1>
```

```
    <p>Este es un párrafo de texto. <br> El texto continúa  
    en una nueva línea.</p>
```



`<p>Otro párrafo de texto con <br> un salto de línea forzado.</p>`

`</body>`

`</html>`

En este ejemplo, se utiliza la etiqueta `<br>` para insertar saltos de línea en los párrafos de texto. Cada vez que se encuentra `<br>`, el texto continúa en una nueva línea.

El uso de la etiqueta `<br>` es útil cuando se desea insertar un salto de línea manualmente dentro de un párrafo de texto o en cualquier lugar donde se requiera un cambio de línea. Ten en cuenta que su uso excesivo puede dificultar la mantenibilidad del código y es recomendable utilizar técnicas de diseño responsivo y CSS para el espaciado y la presentación del contenido.

## 12. `<strong>` `<em>`

La etiqueta `<strong>` se utiliza para resaltar o enfatizar un texto importante, generalmente haciendo que el texto se muestre en negrita. Indica que el contenido dentro de la etiqueta tiene un significado semántico fuerte. Los navegadores web por defecto muestran el texto dentro de `<strong>` en negrita, aunque el estilo exacto puede variar según la hoja de estilos utilizada.

`<p>Este es un texto <strong>importante</strong> en negrita.</p>`

En este ejemplo, la palabra "importante" se muestra en negrita debido al uso de la etiqueta `<strong>`, lo que ayuda a resaltarla y comunicar su significado semántico.

La etiqueta `<em>` se utiliza para enfatizar o destacar un texto, generalmente haciendo que el texto se muestre en cursiva. Indica que el contenido dentro

de la etiqueta tiene énfasis, resaltando su significado o importancia. Al igual que con la etiqueta `<strong>`, el estilo exacto de cómo se muestra el texto en cursiva puede variar según la hoja de estilos.

`<p>Este es un texto <em>enfaticado</em> en cursiva.</p>`

En este ejemplo, la palabra "enfaticado" se muestra en cursiva debido al uso de la etiqueta `<em>`, lo que ayuda a destacarla y resaltar su importancia o énfasis.

Es importante tener en cuenta que el uso de estas etiquetas (`<strong>` y `<em>`) no debe considerarse simplemente como una forma de aplicar estilos de texto, sino como una forma de comunicar el significado y la intención semántica del contenido al usuario y a los motores de búsqueda. Se recomienda utilizar estas etiquetas apropiadamente para mejorar la accesibilidad y la estructura de tus documentos HTML.

### 13. `<sup>` `<sub>`

La etiqueta `<sup>` se utiliza para indicar texto sobrescrito, es decir, texto que se muestra ligeramente elevado sobre la línea de texto normal. Se utiliza comúnmente para mostrar exponentes, notas a pie de página, abreviaturas y otros tipos de texto que deben aparecer en una posición superior.

`<p>La fórmula química del agua es H<sub>2</sub>O.</p>`

En este ejemplo, el número "2" se muestra como subíndice usando la etiqueta `<sub>`, que indica la cantidad de átomos de hidrógeno en la molécula de agua.

La etiqueta `<sub>` se utiliza para indicar texto subíndice, es decir, texto que se muestra ligeramente desplazado hacia abajo de la línea de texto normal. Se utiliza comúnmente para mostrar números subíndice, como en las

## EVOLUCIÓN CONTINUA

fórmulas químicas o matemáticas, fórmulas químicas y notas a pie de página.

<p>La fórmula química del dióxido de carbono es CO<sub>2</sub>.</p>

En este ejemplo, el número "2" se muestra como subíndice usando la etiqueta <sub>, que indica la cantidad de átomos de oxígeno en la molécula de dióxido de carbono.

Las etiquetas (<sup> y <sub>) se utilizan para representar contenido con una posición especial y deben usarse adecuadamente según el contexto semántico. Su estilo visual puede variar según las hojas de estilo utilizadas, pero su función principal es indicar la posición relativa del texto en relación con el texto circundante.

#### 14.<del><ins>

La etiqueta <del> se utiliza para mostrar texto eliminado o tachado. Se utiliza para indicar que cierto contenido ha sido eliminado o ya no es válido en una versión actualizada del texto.

<p>El precio original es \$100, pero ahora está en oferta por <del>\$80</del>.</p>

En este ejemplo, el precio original de \$80 se muestra con la etiqueta <del>, lo que indica que ha sido eliminado o ya no es válido.

La etiqueta <ins> se utiliza para mostrar texto insertado o subrayado. Se utiliza para indicar que cierto contenido ha sido agregado o modificado en una versión actualizada del texto.

<p>El nuevo producto está disponible en <ins>varios colores</ins>.</p>

En este ejemplo, "varios colores" se muestra con la etiqueta <ins>, lo que indica que ha sido insertado o agregado en la versión actualizada.

Estas etiquetas (<del> y <ins>) son útiles para resaltar los cambios realizados en el contenido, como en documentos versionados o en colaboraciones en línea, donde es importante mostrar las diferencias entre las versiones antiguas y nuevas.

## 15. &nbsp;

La entidad HTML &nbsp; (non-breaking space) se utiliza para insertar un espacio en blanco que no se colapsa en HTML. A diferencia de un espacio en blanco normal, un espacio no colapsante asegura que los elementos adyacentes se mantengan en la misma línea y no se separen automáticamente.

Texto antes&nbsp;del espacio&nbsp;en blanco&nbsp;Texto después.

En el ejemplo anterior, &nbsp; se utiliza para insertar espacios no colapsantes entre las palabras "antes", "del", "espacio", "en", "blanco" y "Texto". Esto asegurará que estas palabras permanezcan juntas en la misma línea, incluso si hay cambios en el ancho del contenedor o en el ajuste de la página.

## 16. Atributos

Un atributo HTML es un valor adicional que se utiliza para modificar o proporcionar información adicional sobre un elemento HTML. Los atributos se agregan a las etiquetas HTML y se definen utilizando el formato nombre="valor". Proporcionan instrucciones o datos que afectan el comportamiento o la apariencia del elemento.

### Atributos básicos:

Algunos atributos básicos se pueden aplicar a muchas etiquetas HTML, como:

## EVOLUCIÓN CONTINUA

- **class:** Se utiliza para especificar una o varias clases CSS para el elemento. Permite aplicar estilos y manipular el elemento con CSS y JavaScript.
  - **Ejemplo:** `<div class="caja">Contenido de la caja</div>`
- **id:** Se utiliza para especificar un identificador único para el elemento. Puede ser útil para seleccionar y manipular el elemento con CSS y JavaScript.
  - **Ejemplo:** `<h1 id="titulo">Título de la página</h1>`

**Atributos específicos de elementos:**

Algunos elementos HTML tienen atributos específicos que se utilizan para controlar su comportamiento o apariencia. Aquí hay algunos ejemplos:

- **src:** Se utiliza en la etiqueta `<img>` para especificar la ruta de origen de una imagen.
  - **Ejemplo:** ``
- **href:** Se utiliza en la etiqueta `<a>` para especificar la URL a la que debe dirigirse el enlace.
  - **Ejemplo:** `<a href="https://www.ejemplo.com">Enlace a ejemplo.com</a>`
- **disabled:** Se utiliza en elementos como `<input>` o `<button>` para deshabilitar la interacción con el elemento.
  - **Ejemplo:** `<input type="text" disabled>`

Estos son solo algunos ejemplos de atributos HTML, pero hay muchos más atributos que se utilizan en diferentes elementos para controlar su comportamiento y apariencia. Es importante consultar la documentación y las especificaciones de HTML para conocer todos los atributos disponibles y su uso adecuado en cada contexto.

## 17.<abbr>

La etiqueta <abbr> en HTML se utiliza para marcar una abreviatura o acrónimo en el texto y proporcionar una descripción completa de la abreviatura. Esto ayuda a los lectores a comprender el significado completo de la abreviatura o acrónimo.

```
<abbr title="Descripción completa">Abreviatura</abbr>
```

title: Es el atributo obligatorio de la etiqueta <abbr>, y se utiliza para proporcionar la descripción completa de la abreviatura o acrónimo.

```
<p>La <abbr title="Organización de las Naciones Unidas">ONU</abbr> es una organización internacional.</p>
```

En este ejemplo, la abreviatura "ONU" se marca con la etiqueta <abbr> y se proporciona una descripción completa utilizando el atributo title. Cuando los usuarios pasan el cursor sobre la abreviatura, se muestra una descripción emergente que indica la descripción completa "Organización de las Naciones Unidas".

El uso de la etiqueta <abbr> es útil para proporcionar claridad y comprensión a los lectores sobre las abreviaturas o acrónimos utilizados en el texto. Ayuda a evitar confusiones y garantiza que los lectores comprendan completamente el significado de las abreviaturas utilizadas en el contenido.

## 18.Meta etiqueta UTF 8

La etiqueta <meta> con el atributo charset="UTF-8" se utiliza para indicar que el documento HTML utiliza la codificación de caracteres UTF-8. UTF-8 es una codificación de caracteres universal que permite representar una amplia gama de caracteres de diferentes idiomas y símbolos.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>

  <meta charset="UTF-8">

  <title>Título de la página</title>

</head>

<body>

  Contenido de la página.

</body>

</html>
```

En el ejemplo anterior, se coloca la etiqueta `<meta>` dentro de la sección `<head>` del documento HTML. El atributo `charset="UTF-8"` indica que el documento HTML está codificado utilizando UTF-8.

Es importante incluir la etiqueta `<meta charset="UTF-8">` en la parte superior del documento HTML para asegurarse de que los caracteres especiales y los idiomas no se muestren incorrectamente en el navegador. Sin esta etiqueta, los caracteres especiales pueden aparecer como símbolos extraños o caracteres no reconocidos.

La etiqueta `<meta charset="UTF-8">` es un estándar ampliamente utilizado en el desarrollo web actual, ya que permite representar de manera adecuada la mayoría de los caracteres y símbolos utilizados en diferentes idiomas y alfabetos.

Recuerda que al utilizar UTF-8 como la codificación de caracteres, debes asegurarte de que los archivos del documento HTML y cualquier otro archivo asociado (como hojas de estilo CSS y archivos JavaScript) estén codificados correctamente en UTF-8 para evitar problemas de visualización de caracteres.