

3. Navegadores web.

Los navegadores son las herramientas que interpretan los lenguajes de marcas y muestran las páginas web a las personas usuarias. En esta sección, exploraremos los diferentes tipos de navegadores, sus características y cómo afectan la visualización de las páginas web.

3.1. Navegadores modo texto.

Los navegadores en modo texto son navegadores web que muestran únicamente el texto de las páginas web y omiten el contenido gráfico. Resultan útiles en conexiones de banda ancha limitada, ya que suelen cargar las páginas más rápidamente que los navegadores gráficos debido a la menor demanda de ancho de banda. Además, son beneficiosos para usuarios con discapacidad visual o ceguera parcial.

A continuación, se presentan algunos navegadores en modo texto:

- Lynx: Uno de los navegadores de solo texto más antiguos y todavía en uso. Es altamente configurable y tiene una interfaz de usuario similar a la de una línea de comandos.
- Links: Otro navegador en modo texto muy popular.
- ELinks: Una versión avanzada y con todas las funciones del navegador Lynx.
- w3m: Navegador de texto que puede mostrar tablas y marcos.
- WebIE: Navegador de texto gratuito para Windows que muestra solo el contenido de texto de un sitio web.
- Emacs / W3 & EWW para GNU Emacs: Navegadores en modo texto que se ejecutan dentro del editor de texto Emacs.
- Charlotte (por CMS / de VM): Navegador en modo texto que se ejecuta en el sistema operativo VM de IBM.
- brows: Navegador moderno en modo texto que admite gráficos, video y páginas web actuales.

3.2. Los navegadores más utilizados.

Los navegadores más utilizados hoy en día incluyen Google Chrome, Mozilla Firefox, Microsoft Edge y Safari. Cada uno de estos navegadores tiene sus propias características y ventajas, pero todos ofrecen una experiencia de navegación moderna y completa.

3.2.1. “Google Chrome, Mozilla Firefox, Microsoft Edge y Safari”.

La siguiente tabla proporciona una visión general de algunos de los navegadores web más populares, destacando su desarrollador, características, ventajas y desventajas.

Navegador	Desarrollador	Ventajas	Desventajas
Google Chrome	Google	Alta velocidad de navegación. Gran cantidad de extensiones. Sincronización con servicios de Google.	Alto consumo de memoria RAM. Cuestiones de privacidad.
Mozilla Firefox	Mozilla Foundation	Enfoque en privacidad y seguridad. Alta personalización. Complementos y extensiones diversos.	Menor compatibilidad con algunas páginas web. Velocidad menor que Chrome.
Microsoft Edge	Microsoft	Integración con el ecosistema de Microsoft. Buen rendimiento en Windows. Bajo consumo de batería	Menor disponibilidad de extensiones. Preferencia por productos Microsoft.
Safari	Apple	Optimizado para dispositivos Apple. Alto rendimiento. Bajo consumo de energía	Disponible solo en dispositivos Apple. Menos extensiones y personalización.

Actividad
3

Decide qué navegador es más recomendable para cada escenario, y justifica tu elección:

1. Necesidad de alta velocidad de navegación y gran cantidad de extensiones para desarrolladores.
2. Integración con el ecosistema de Microsoft, buen rendimiento en dispositivos Windows, y bajo consumo de batería.
3. Alta optimización para dispositivos Apple, alto rendimiento y bajo consumo de energía.

3.3. Diferencias de visualización.

Los estándares web son un conjunto de normas desarrolladas para guiar la creación de sitios web, mantenidas por el World Wide Web Consortium (W3C) y otros grupos de estándares. Estos estándares incluyen HTML, CSS y JavaScript, los lenguajes de programación fundamentales para construir y diseñar sitios web.

A pesar de que los navegadores modernos intentan seguir estos estándares, cada uno puede interpretar y renderizar el código HTML, CSS y JavaScript de manera ligeramente diferente. Esto se debe a varias razones, entre ellas las diferencias en la interpretación del código. Cada navegador

EDITORIAL TUTOR FORMACIÓN

posee su propio motor de renderizado (como Blink para Chrome y Gecko para Firefox) que interpreta el código y lo muestra en la pantalla, lo que puede llevar a pequeñas variaciones en la forma en que el código es interpretado.

Además, la compatibilidad con características puede variar. Algunos navegadores no son compatibles con todas las características de HTML, CSS y JavaScript, especialmente si se trata de características más nuevas, lo que puede resultar en que algunas partes de un sitio web no se muestren correctamente en ciertos navegadores.

Las actualizaciones del navegador también juegan un papel importante. Los navegadores se actualizan regularmente para mejorar la seguridad, añadir nuevas características y mejorar la compatibilidad con los estándares web. Sin embargo, no todas las personas usuarias actualizan sus navegadores de inmediato, por lo que algunos pueden utilizar versiones más antiguas que no soportan todas las características de los estándares web actuales.

Debido a estas razones, aunque los sitios web están diseñados para verse y funcionar de manera consistente en todos los navegadores, en la práctica puede haber pequeñas diferencias en la apariencia y funcionalidad de un sitio web dependiendo del navegador utilizado. Esto es algo que los desarrolladores web deben considerar al diseñar y probar sus sitios.

4. Marcas para dar formato al documento.

Las marcas de formato son esenciales para definir la apariencia de una página web. Aquí aprenderemos sobre las etiquetas que nos permiten organizar y estilizar el contenido, desde la creación de párrafos hasta la definición de listas y la aplicación de estilos.

4.1. Marcas de inicio y final.

Las marcas de inicio y final son esenciales para definir la estructura básica de un documento HTML. Las etiquetas `<html>`, `<head>` y `<body>` son las principales marcas que delimitan las diferentes secciones de un documento:

`<html>`: Esta etiqueta es la raíz de un documento HTML. Todo el código HTML debe estar contenido dentro de las etiquetas de inicio `<html>` y final `</html>`, indicando al navegador que el documento es un archivo HTML.

`<head>`: La etiqueta `<head>` se utiliza para contener la información meta del documento HTML, que no se visualiza en el contenido principal de la página web. Esto incluye el título de la página (`<title>`), enlaces a archivos CSS (`<link>`), scripts de JavaScript (`<script>`), entre otros.

`<body>`: La etiqueta `<body>` contiene todo el contenido visible para las personas usuarias en el navegador, como texto, imágenes, enlaces, tablas, listas, entre otros elementos.

4.2. Marcas de aspecto.

El uso de CSS (Cascading Style Sheets) ofrece una gran flexibilidad en la personalización del aspecto de los elementos HTML. Permite definir estilos para elementos individuales, grupos de elementos o incluso para toda la página web. Además, facilita la creación de diseños responsivos que se adaptan al tamaño de la pantalla del dispositivo, mejorando la accesibilidad y usabilidad del contenido.

4.3. Marcas de párrafo.

Las etiquetas `<p>` se utilizan para definir párrafos de texto en HTML. Cada párrafo se representa en una nueva línea y puede contener texto, enlaces, imágenes y otros elementos en línea.

No solo contienen texto, sino que también pueden incluir otros elementos en línea como enlaces (`<a>`), imágenes (``) y etiquetas de formato como `` o ``. Esto permite una gran variedad en la presentación y organización del contenido textual en las páginas web.

4.4. Marcas de fuentes y colores.

Aunque la etiqueta `` fue utilizada en versiones anteriores de HTML para definir el estilo del texto, se desaconseja su uso en HTML5. En su lugar, se deben usar las propiedades de CSS para definir fuentes y colores. Por ejemplo, la propiedad `font-family` en CSS define la tipografía, `font-size` establece el tamaño de la fuente, y `color` especifica el color del texto.

4.5. Listas ordenadas.

Las listas en HTML pueden ser ordenadas () o desordenadas (). Las listas ordenadas utilizan números o letras para los elementos de la lista, mientras que las listas desordenadas utilizan viñetas. Las listas ordenadas () y desordenadas () son esenciales para presentar información de manera clara y estructurada en HTML. Cada elemento de la lista se define con la etiqueta .

En las listas ordenadas, los elementos se numeran automáticamente, mientras que en las listas desordenadas se utilizan viñetas. Estas estructuras permiten organizar el contenido de forma coherente y fácil de seguir para las personas usuarias. A continuación, se expone un ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Listas en HTML</title>
</head>
<body>
  <h2>Lista ordenada</h2>
  <ol>
    <li>Primer ítem</li>
    <li>Segundo ítem</li>
    <li>Tercer ítem</li>
  </ol>

  <h2>Lista desordenada</h2>
  <ul>
    <li>Primer ítem</li>
    <li>Segundo ítem</li>
    <li>Tercer ítem</li>
  </ul>
</body>
</html>
```

Lista ordenada

1. Primer ítem
2. Segundo ítem
3. Tercer ítem

Lista desordenada

- Primer ítem
- Segundo ítem
- Tercer ítem

Pie de página: Resultado del código anterior renderizado en un navegador web.

Actividad
4

Redacta un documento HTML básico que incluya las etiquetas de inicio y final principales (<html>, <head>, <body>) y define su estructura básica con un título de página, una breve descripción en un párrafo y un enlace a una hoja de estilo CSS externa.

5. Enlaces y direccionamientos.

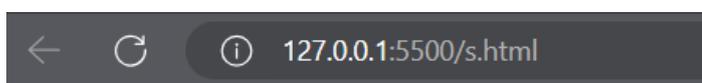
Los enlaces y direccionamientos son fundamentales para la navegación web, permitiendo a las personas usuarias moverse entre diferentes páginas y secciones de un sitio. Esta sección cubre cómo crear y gestionar enlaces, los diferentes tipos de enlaces y su importancia para la navegación eficiente dentro y fuera de un sitio web.

5.1. Creación de un enlace.

La etiqueta `<a href>` se utiliza para crear enlaces en HTML. Estos enlaces pueden dirigir a otras páginas web, archivos o ubicaciones dentro de la misma página.

Un ejemplo de código HTML para crear un enlace es el siguiente:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Creación de enlaces</title>
</head>
<body>
  <p>Visita <a href="https://www.info.com">Info</a> para más información.</p>
</body>
</html>
```



Visita [Info](#) para más información.

Pie de página: Resultado del código anterior renderizado en un navegador web.

5.2. Tipos de enlace.

Los enlaces en HTML pueden ser internos (anclas) o externos (vínculos a otras páginas o sitios).

5.2.1. Anclas.

Las anclas se utilizan para crear enlaces dentro de la misma página. Esto se logra utilizando atributos `id` y `href` con un valor que comienza con `#`.

Un ejemplo de código HTML con Anclas es el siguiente:

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Enlaces con Anclas</title>
</head>
<body>
  <h1>Tabla de contenidos</h1>
  <ul>
    <li><a href="#seccion1">Sección 1</a></li>
    <li><a href="#seccion2">Sección 2</a></li>
    <li><a href="#seccion3">Sección 3</a></li>
  </ul>

  <h2 id="seccion1">Sección 1</h2>
  <p>Contenido de la sección 1.</p>

  <h2 id="seccion2">Sección 2</h2>
  <p>Contenido de la sección 2.</p>

  <h2 id="seccion3">Sección 3</h2>
  <p>Contenido de la sección 3.</p>
</body>
</html>

```

Tabla de contenidos

- [Sección 1](#)
- [Sección 2](#)
- [Sección 3](#)

Sección 1

Contenido de la sección 1.

Sección 2

Contenido de la sección 2.

Sección 3

Contenido de la sección 3.

Pie de página: Resultado del código anterior renderizado en un navegador web.

5.2.2. Vínculos.

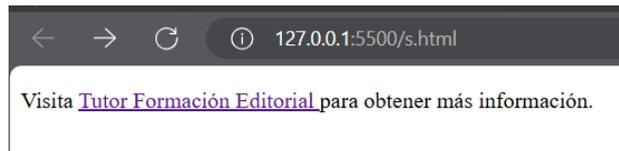
Los vínculos se utilizan para enlazar a otras páginas o sitios web externos.

Un ejemplo de código HTML con Vínculos es el siguiente:

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Enlaces Externos</title>
</head>
<body>
  <p>Visita <a href="https://tutorformacion.es/" target="_blank"> Tutor Formación Editorial </a> para obtener más información.</p>
</body>
</html>

```



Pie de página: Resultado del código anterior renderizado en un navegador web.

5.3. Los enlaces y la navegación.

Los enlaces son la base de la navegación en la web. Permiten a las personas usuarias moverse entre páginas y secciones de manera eficiente y organizada.

A continuación, se exponen algunos ejemplos prácticos de navegación con enlaces:

- Enlaces de navegación en un sitio web:

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Navegación del Sitio</title>
</head>
<body>
  <nav>
    <ul>
      <li><a href="index.html">Inicio</a></li>
      <li><a href="about.html">Acerca de</a></li>
      <li><a href="contact.html">Contacto</a></li>
    </ul>
  </nav>
  <h1>¡Hola! este es mi sitio web</h1>
  <p>Contenido de la página de inicio.</p>
</body>
</html>

```



Pie de página: Resultado del código anterior renderizado en un navegador web.

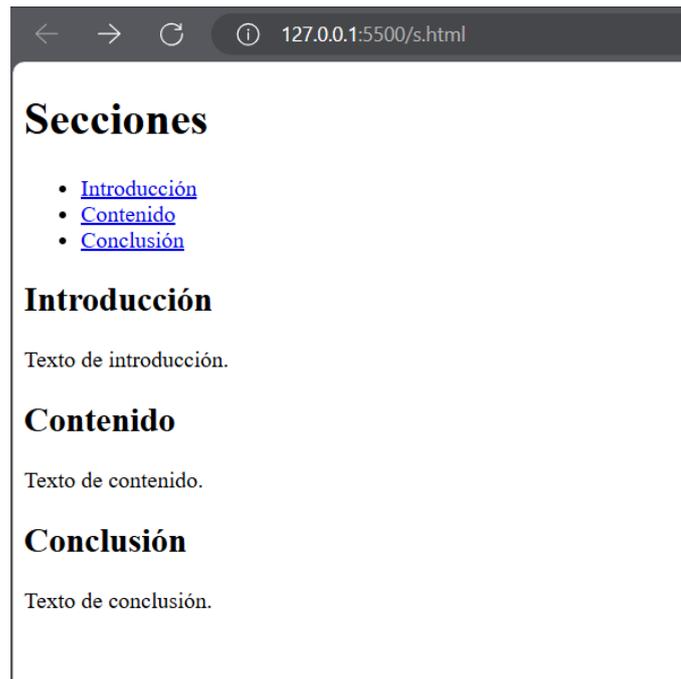
Enlaces de navegación en la misma página:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Navegación Interna</title>
</head>
<body>
  <h1>Secciones</h1>
  <ul>
    <li><a href="#introduccion">Introducción</a></li>
    <li><a href="#contenido">Contenido</a></li>
    <li><a href="#conclusion">Conclusión</a></li>
  </ul>

  <h2 id="introduccion">Introducción</h2>
  <p>Texto de introducción.</p>

  <h2 id="contenido">Contenido</h2>
  <p>Texto de contenido.</p>

  <h2 id="conclusion">Conclusión</h2>
  <p>Texto de conclusión.</p>
</body>
</html>
```



Pie de página: Resultado del código anterior renderizado en un navegador web.

Actividad 5

Redacta un documento HTML que demuestre cómo crear enlaces y gestionar la navegación dentro de una página web. Incluye ejemplos de enlaces internos (anclas) y externos (vínculos). Asegúrate de utilizar correctamente las etiquetas y atributos correspondientes para cada tipo de enlace.

6. Marcos y capas.

Los marcos y capas son elementos estructurales en el desarrollo web que permiten organizar y presentar el contenido de manera eficiente. Los marcos, como los `<iframe>`, permiten incrustar contenido de otras fuentes, mientras que las capas, definidas por elementos como `<div>` y ``, facilitan la organización y el diseño del contenido dentro de una página. En esta parte, aprenderemos cómo crear y configurar marcos y capas para mejorar la estructura y usabilidad de una página web.

6.1. Marcos.

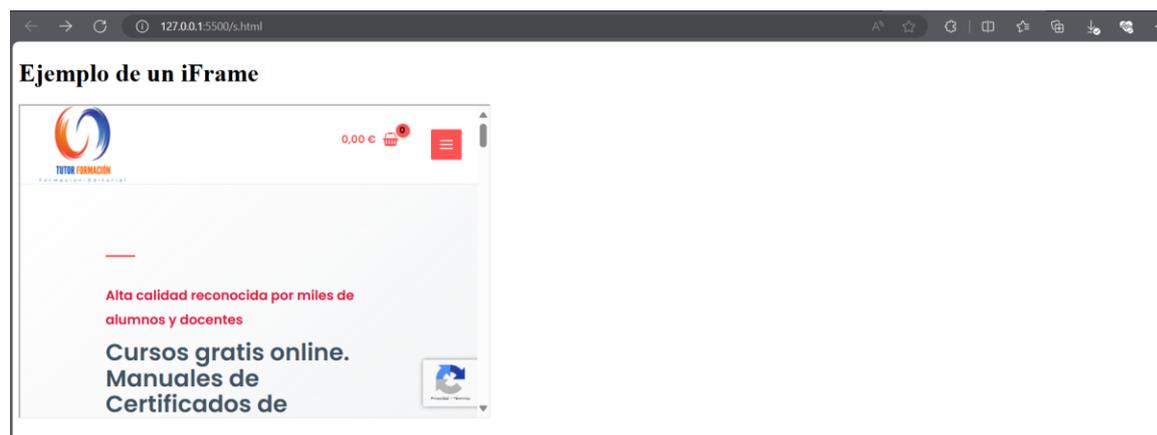
Los marcos permiten incluir documentos HTML dentro de otros documentos HTML, creando así una estructura modular.

6.1.1. Creación de marcos.

El uso de la etiqueta `<iframe>` es fundamental para crear marcos. Un `<iframe>` permite incrustar otro documento HTML dentro del documento principal.

Un ejemplo de código HTML con `<iframe>` es el siguiente:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Creación de Marcos</title>
</head>
<body>
  <h1>Ejemplo de un iFrame</h1>
  <iframe src="https://tutorformacion.es/" width="600" height="400"></iframe>
</body>
</html>
```



Pie de página: Resultado del código anterior renderizado en un navegador web.

6.1.2. Tipos de marcos.

El principal tipo de marco utilizado es el <iframe>. Esta etiqueta es ampliamente soportada y flexible.

El <iframe> permite insertar otro documento HTML en el documento actual, creando una ventana dentro de la página que puede mostrar contenido independiente.

6.1.3. Situación de los marcos.

El posicionamiento de los marcos se puede controlar utilizando CSS, lo que permite una integración más fluida en el diseño general de la página.

Un ejemplo de posicionamiento con CSS es el siguiente:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Posicionamiento de iFrame</title>
  <style>
    .iframe-container {
      position: relative;
      width: 80%;
      height: 500px;
      margin: auto;
    }
    iframe {
      position: absolute;
      top: 0;
      left: 0;
      width: 100%;
      height: 100%;
    }
  </style>
</head>
<body>
  <div class="iframe-container">
    <iframe src="https://tutorformacion.es/"></iframe>
  </div>
</body>
</html>
```



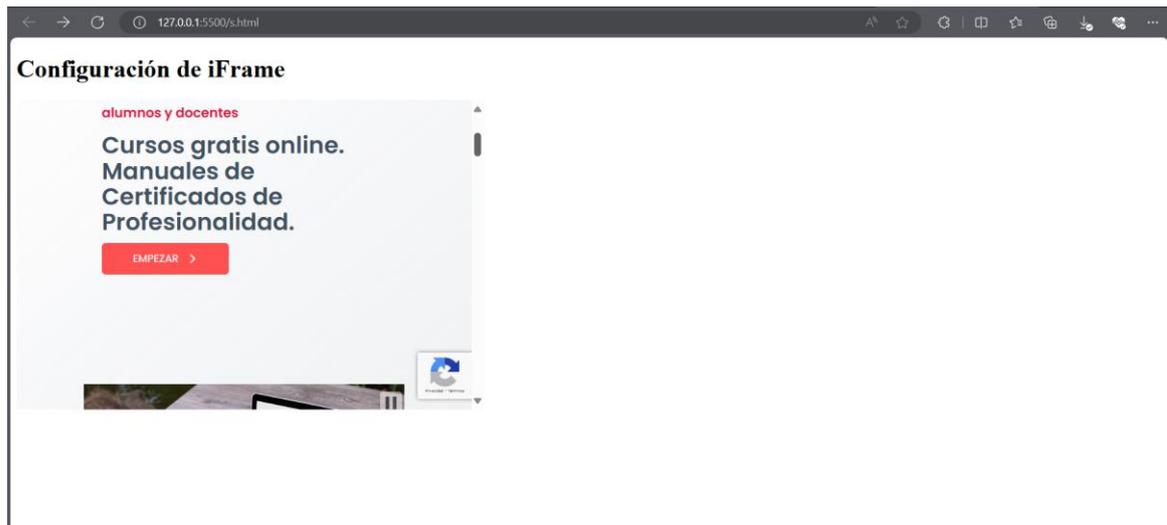
Pie de página: Resultado del código anterior renderizado en un navegador web.

6.1.4. Configuración de los marcos.

Los atributos de un `<iframe>` como `src`, `height`, `width`, `frameborder`, y `allowfullscreen` permiten configurar su apariencia y funcionalidad.

Un ejemplo de código HTML con configuración de `<iframe>` es el siguiente:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Configuración de iFrame</title>
</head>
<body>
  <h1>Configuración de iFrame</h1>
  <iframe src="https://tutor-formacion.es/" width="600" height="400" frameborder="0" allowfullscreen></iframe>
</body>
</html>
```



Pie de página: Resultado del código anterior renderizado en un navegador web.

6.2. Capas.

Las capas permiten organizar y estilizar contenido de forma efectiva, utilizando etiquetas como `<div>` y `` junto con CSS.

6.2.1. Definición de capas.

Las capas en HTML se definen usando elementos como `<div>` para secciones de bloque y `` para elementos en línea. Estos elementos se combinan con CSS para crear diseños complejos.

El elemento `<div>` es un contenedor de bloque que se utiliza para agrupar otros elementos y aplicar estilos o scripts. El `` es un contenedor en línea utilizado para aplicar estilos a una parte del texto o a un grupo pequeño de elementos.

A continuación, se expone un ejemplo de código HTML con `<div>` y ``:

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Capas en HTML</title>
  <style>
    .container {
      border: 1px solid black;
      padding: 10px;
    }
    .highlight {
      color: red;
      font-weight: bold;
    }
  </style>
</head>
<body>
  <div class="container">
    Este es un contenedor <span class="highlight">resaltado</span>.
  </div>
</body>
</html>

```



Pie de página: Resultado del código anterior renderizado en un navegador web.

6.2.2. Creación de capas.

La creación de capas implica usar elementos `<div>` y `` y aplicarles estilos CSS.

Un ejemplo de creación de capas con CSS:

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Creación de Capas</title>
  <style>
    .header {
      background-color: #f2f2f2;
      padding: 20px;
      text-align: center;
    }
    .content {
      padding: 20px;
    }
    .footer {
      background-color: #f2f2f2;
      padding: 10px;
      text-align: center;
    }
  </style>
</head>
<body>
  <div class="header">
    <h1>Encabezado</h1>
  </div>
  <div class="content">
    <p>Contenido principal de la página.</p>
  </div>
  <div class="footer">
    <p>Pie de página.</p>
  </div>
</body>
</html>

```



Pie de página: Resultado del código anterior renderizado en un navegador web.

6.2.3. Utilización de las capas.

Las capas se utilizan para organizar el diseño de la página, permitiendo una estructura clara y estilos personalizados.

Un ejemplo práctico de Layout con capas es el siguiente:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Layout con Capas</title>
  <style>
    .layout {
      display: flex;
    }
    .sidebar {
      width: 25%;
      background-color: #e0e0e0;
      padding: 10px;
    }
    .main-content {
      width: 75%;
      padding: 10px;
    }
  </style>
</head>
<body>
  <div class="layout">
    <div class="sidebar">
      <p>Barra lateral</p>
    </div>
    <div class="main-content">
      <p>Contenido principal</p>
    </div>
  </div>
</body>
</html>
```



Pie de página: Resultado del código anterior renderizado en un navegador web.

A continuación, se expone el código de un documento HTML que combina los diferentes aspectos que se han estudiado hasta ahora y una hoja de diseño CSS orientada a una web ficticia sobre rock español:

Código HTML del documento index.html:

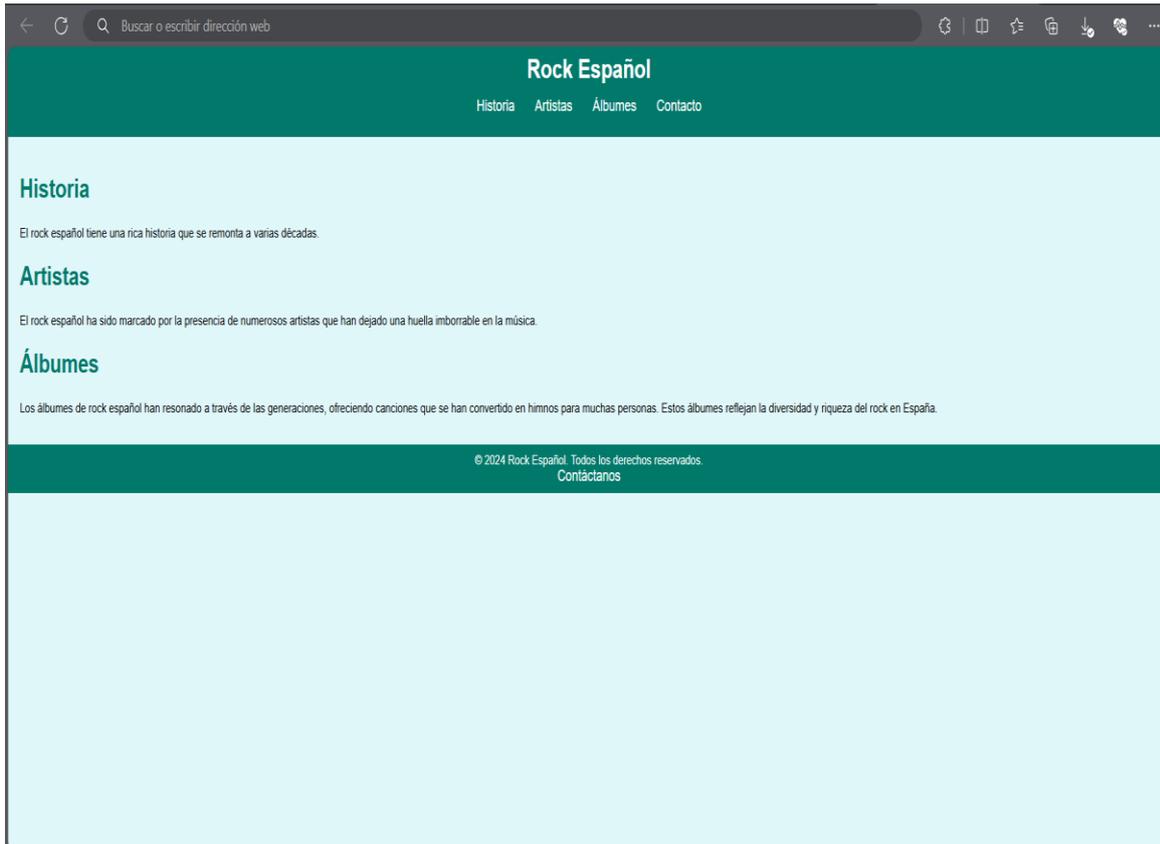
Ejemplo final

```
index.html
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Rock Español</title>
7   <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10  <header>
11    <h1>Rock Español</h1>
12    <nav>
13      <ul>
14        <li><a href="#historia">Historia</a></li>
15        <li><a href="#artistas">Artistas</a></li>
16        <li><a href="#albumes">Álbumes</a></li>
17        <li><a href="#contacto">Contacto</a></li>
18      </ul>
19    </nav>
20  </header>
21  <main>
22    <section id="historia">
23      <h2>Historia</h2>
24      <p>El rock español tiene una rica historia que se remonta a varias décadas.</p>
25    </section>
26    <section id="artistas">
27      <h2>Artistas</h2>
28      <p>El rock español ha sido marcado por la presencia de numerosos artistas que han dejado una huella imborrable en la música.</p>
29    </section>
30    <section id="albumes">
31      <h2>Álbumes</h2>
32      <p>Los álbumes de rock español han resonado a través de las generaciones, ofreciendo canciones que se han convertido en himnos para muchas personas. Estos álbumes reflejan la diversidad y riqueza del rock en España.</p>
33    </section>
34  </main>
35  <footer>
36    <p>&copy; 2024 Rock Español. Todos los derechos reservados.</p>
37    <p><a href="#contacto">Contáctanos</a></p>
38  </footer>
39 </body>
40 </html>
```

EDITORIAL TUTOR FORMACIÓN

Código CSS del archivo styles.css:

```
1  body {
2      font-family: Arial, sans-serif;
3      margin: 0;
4      padding: 0;
5      background-color: #e0f7fa;
6  }
7  header, footer {
8      background-color: #00796b;
9      color: #ffffff;
10     text-align: center;
11     padding: 10px 0;
12 }
13 header h1, footer p {
14     margin: 0;
15 }
16 nav ul {
17     list-style: none;
18     padding: 0;
19     text-align: center;
20 }
21 nav ul li {
22     display: inline;
23     margin: 0 15px;
24 }
25 nav ul li a, footer a {
26     color: #ffffff;
27     text-decoration: none;
28     font-size: 1.2em;
29 }
30 main {
31     padding: 20px;
32 }
33 section h2 {
34     font-size: 2em;
35     color: #00796b;
36 }
```



Pie de página: Resultado del código anterior renderizado en un navegador web.



La etiqueta `<meta name="viewport" content="width=device-width, initial-scale=1.0">` es una etiqueta HTML que se utiliza para controlar el diseño en dispositivos móviles. Aquí se explica qué significa cada parte:

- `name="viewport"`: Este atributo le dice al navegador que se quiere controlar el área de visualización (viewport) de la página.
- `content="width=device-width"`: Este atributo establece el ancho de la página para seguir el ancho de la pantalla del dispositivo (que varía dependiendo del dispositivo).
- `initial-scale=1.0`: Este atributo controla el nivel de zoom cuando la página se carga por primera vez. El valor 1.0 no aplica ningún zoom.

7. Prueba de autoevaluación.

¿Cuál es el lenguaje de marcas más conocido utilizado para crear páginas web?

- a) XML
- b) LaTeX
- c) HTML
- d) CSS

¿Cuál de los siguientes es un ejemplo de un elemento de bloque en HTML?

- a) ``
- b) `<a>`
- c) `<div>`
- d) ``

¿Qué atributo se utiliza para definir el destino de un enlace en HTML?

- a) `src`
- b) `class`
- c) `id`
- d) `href`

¿Qué etiqueta define el título de una página web en el navegador?

- a) `<head>`
- b) `<title>`
- c) `<meta>`
- d) `<body>`

¿Qué navegador es desarrollado por Mozilla Foundation y enfocado en la privacidad y seguridad?

- a) Google Chrome
- b) Safari
- c) Microsoft Edge
- d) Mozilla Firefox

Completar:

HTML es el acrónimo de _____.

Los lenguajes de marcas como HTML y XML permiten estructurar el _____ y definir la _____ de la información en la web.

La etiqueta _____ se utiliza para definir un párrafo de texto en HTML.

En HTML, los elementos de _____ crean un bloque visible en la página y ocupan todo el ancho disponible.

Los estilos CSS pueden incorporarse en un documento HTML de tres maneras principales: CSS _____, CSS _____ y CSS _____.