

Creación de un sencillo blog personal

INTRODUCCIÓN TEÓRICA. HERENCIA Y CASCADA EN CSS3

Ambos conceptos están relacionados, pero son diferentes. Los definimos:

- La **herencia** permite declarar propiedades en elementos de nivel superior, y que estas propiedades se transmitan a todos los elementos descendientes. Sólo algunas propiedades se heredan por defecto, pero la herencia puede forzarse mediante la palabra clave **inherit**.
- La **cascada** permite solucionar los conflictos cuando varias declaraciones afectan a un elemento concreto. Las declaraciones importantes anulan a las que no son tan importantes. Entre declaraciones de igual nivel, la **especificidad** de la regla controla cuál se aplica. Y, si todas las demás son iguales, el orden de las fuentes supone la distinción definitiva.

En resumen, podemos decir que la herencia está relacionada con la forma en que los elementos HTML heredan propiedades de sus elementos padres (aquellos que los contienen) y los transmiten a sus hijos, mientras que la cascada está relacionada con las declaraciones CSS que se aplican a un documento y cómo las reglas contradictorias se anulan o no entre ellas.

Todos los elementos de un documento HTML heredan todas las propiedades heredables de su padre, excepto el elemento raíz (html), que no tiene padre. Por ejemplo, si escribimos en CSS:

```
html { font: 80% Verdana,sans-serif; }
```

Esta regla se aplica solo al elemento html. No se ha indicado ninguna regla para los títulos o los párrafos. Por ese motivo, ahora todo el texto se muestra en Verdana y al 80% del tamaño por defecto, precisamente por la herencia.

Veamos los tres conceptos que controlan el **orden en el que se aplican las declaraciones CSS** en la cascada:

1. **Importancia.**
2. **Especificidad.**
3. **Orden en las fuentes.**

El prioritario es la **importancia**. Si dos declaraciones tienen la misma importancia, la **especificidad** de las reglas decidirá cuál se debe aplicar. Si las reglas tienen la misma especificidad, el **orden de las fuentes** controlará el resultado.

La importancia de una declaración de CSS depende de dónde se ha especificado. Las declaraciones contrapuestas se aplicarán siempre de forma que las nuevas anularán a las más antiguas.

La **especificidad** puede considerarse una medida de cuán específico es el selector de una regla. Un selector de especificidad baja puede dar como resultado muchos elementos. Por ejemplo, el selector universal "*", que da como resultado todos los elementos del documento. En cambio, un selector muy específico puede que solo dé como resultado un único elemento de una página. Por ejemplo #nav, que sólo da como resultado el elemento con una ID de nav.

Si dos o más declaraciones entran en conflicto por un elemento determinado, y todas las ellas tienen la misma importancia, la de la regla con el selector más específico será la que «gana».

Si dos declaraciones afectan al mismo elemento, tienen la misma importancia y la misma especificidad, la señal distintiva final es el orden en las fuentes. La declaración que se ve más adelante en las hojas de estilo «ganará» a las anteriores.

Es decir, si se trabaja con solo una única hoja de estilo externa, las declaraciones al final del archivo CSS anulan a las que aparecen antes en el mismo archivo —si hay conflicto entre ellas—.

En el caso de que las declaraciones contrapuestas se produzcan en diferentes archivos CSS para un documento, entonces el orden en el que se enlazan, se incluyen o se importan las hojas de estilo determina qué declaración se aplica. Es decir, si se tienen dos hojas de estilo enlazadas en el head de un documento, cuando haya conflicto la enlazada al último anulará a la enlazada al primero.

ENUNCIADO

Se propone la creación de un sitio web (tipo blog) dedicado a mostrar noticias de interés personal. El menú de navegación deberá tener diferentes secciones y un contacto. Dará opción a descargas de artículos de interés.

La captura siguiente muestra el aspecto que se sugiere que tenga dicha página web. Se pueden realizar aportaciones, pero básicamente debe contener, al menos, los elementos que se indican en ella.



El estilo de la página irá como archivo externo.

Tiempo de resolución: 1h 30' aprox.

CLAVES DE RESOLUCIÓN

1. En la cabecera (`header`, no `head`) hay que incluir los textos de encabezamiento.
2. Establecemos el tamaño de la página (`Page`) en 1300 px.
3. En el menú de navegación (`nav`) se establecerán los colores de fondo, tipo y color del texto, justificación del texto, así como las distancias a respetar respecto del marco (`padding`).
4. Para los enlaces del menú de navegación se pueden cambiar algunas propiedades. Por ejemplo, cuando pasa el puntero del ratón sobre un enlace.
5. Los estilos para el contenido y, dentro de él, para los diferentes enlaces —un artículo por cada noticia enlazada— se dejan de libre elección (siempre que la estructura sea similar a la presentada en la captura).
6. Para el lateral de la página (`aside`), indicar las dos zonas: para publicidad y para descargas.
7. Se debe establecer un estilo del pie (`footer`) similar al mostrado, y añadir los enlaces al aviso legal y al sitio en obras.

SOLUCIÓN

Utilizando Visual Studio Code, introducir el código HTML siguiente (archivo [UD2_CPE1.html](#)) y la hoja de estilos correspondiente (archivo [UD2_CPE1_estilos.css](#)).

Archivo UD2_CPE1.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Sitio web</title>
  <meta name="autor" content="Nombre_alumno">
  <link rel="stylesheet" href="css/UD2_CPE1_estilos.css">
</head>
<body>
<!-- Definimos la pagina -->
<div id="page">
  <!-- HEADER - definimos la cabecera de la pagina -->
  <div id="header">
    <h1>Título del sitio web</h1>
  </div>
  <!-- NAV establecemos el menu de navegación de la pagina -->
  <div id="nav">
    <ul>
      <li><a href="CPE1_seccion_1.html">Sección 1</a></li>
      <li><a href="CPE1_seccion_2.html">Sección 2</a></li>
```

```

        <li><a href="CPE1_contacto.html">Contacto</a></li>
    </ul>
</div>
<!-- fin de NAV -->
<!-- CONTENT - parte central de la pagina -->
<div id="content">
    <div class="article">
        <h2>Contenido principal</h2>
        <!-- Primera noticia -->
        <div class="article_header">
            <h3>Noticia sobre los incendios forestales</h3>
        </div><br><br>
        <p>Controlado un incendio forestal en el parque natural de Serra Gelada
            en Benidorm ...</p>
        <div class="figure">
            
            <p class="figcaption">Imagen de un incendio</p>
        </div>
        <div class="article_footer">
            <p>comentarios (1)</p>
        </div>
        <!-- Segunda noticia -->
        <div class="article_header">
            <h3>Historia de los vuelos espaciales</h3>
        </div><br><br>
        <p>Los vuelos espaciales pasaron a formar parte de los logros de la
            humanidad en el siglo XX, después de los avances teóricos y
            prácticos ...</p>
        <div class="figure">
            
            <p class="figcaption">Imagen de un viaje espacial</p>
        </div>
        <div class="article_footer">
            <p>comentarios (0)</p>
        </div>
    </div>
</div>
<!-- fin de CONTENT -->
<!-- ASIDE - lateral de la pagina -->
<div id="aside">
    <h5>Zona de publicidad</h5>
    <a href="http://dominio-externo.com">
    </a>
    </br> </br></br> </br>
    <h5>Zona de descargas</h5></br>
    <ul>
        <li><a href="#">Articulo 1</a></li>
        <li><a href="#">Articulo 2</a></li>
    </ul>
</div>

```

```

<!-- fin de ASIDE -->
<!-- FOOTER 1 y 2 - parte inferior de nuestra pagina -->
<div id="footer1">
    <a href="CPE1_aviso_legal.html">Aviso Legal</a>
</div>
<div id="footer2">
    <h4>Sitio web en obras!</h4>
    
</div>
</div>
</body>
</html>

```

Archivo UD2_CPE1_estilos.css

```

* { margin: 0px; padding: 0px;}

h1 { font: bold 2em verdana, sans-serif;}
h2 { font: bold 1.5em verdana, sans-serif;}
h3 { font: bold 1,2em verdana, sans-serif;}
h5 { font: bold 1em verdana, sans-serif;}

body { text-align: center;}

#page { width: 1300px;
        margin: 15px auto;
        text-align: left; }

#header { background: #969591;
          border: 1px solid #999999;
          padding: 20px;
          text-align: center;}

#nav { background: #CCCCCC;
       padding: 5px 15px;}

#nav li { display: inline;
          list-style: none;
          padding: 5px;
          font: bold 14px verdana, sans-serif;
          cursor: default; }

#nav a { text-decoration: none; }

#nav a:hover { color: white; }

#content { float: left;
           width: 900px;

```

```

        margin: 20px; }

#aside {    float: left;
            width: 300px;
            margin: 20px 0px ;
            padding: 20px;
            background: #CCCCCC; }

#footer2 {  clear: both;
            text-align: center;
            padding: 20px;
            border-top: 2px solid #999999;}

#footer1 {  clear: both;
            text-align: right;
            padding: 20px;
            border-top: 2px solid #999999;}

.article { background: #b0e5f5;
            border: 1px solid #999999;
            padding: 20px;
            margin-bottom: 15px; }

.article_header { border-bottom: 1px solid #999999;}

.article_footer { text-align: right; }

.figure {   border: 1px double #999999;
            padding: 5px; }

.figure.figcaption { font: italic 0.2em verdana, sans-serif; }

```

Los archivos incluidos en el solucionario están documentados para que queden claras las aportaciones realizadas para cubrir los puntos del enunciado. También está disponible la carpeta con las imágenes.

Dichos archivos son:

UD2_CPE1.html

UD2_CPE1_estilos.css

CPE1_contacto.html

CPE1_aviso_legal.html

CPE1_seccion_1.html

CPE1_seccion_2.html

Puedes descargar un archivo comprimido que incluye todos ellos en la siguiente pantalla.

Creación de una página web

INTRODUCCIÓN TEÓRICA

Un archivo de estilos puede contener muchas reglas «normales» o reglas propiamente dichas. Cada regla normal puede contener varios selectores y declaraciones, y cada declaración puede contener otras declaraciones. El archivo de estilo puede contener muchas reglas de este tipo.

Los **selectores** se utilizan para seleccionar los elementos a los que se van a aplicar las propiedades en el código CSS.

El uso mas sencillo de los selectores es hacerlo de esta forma:

```
body { text-align: justify; }
```

Esta regla hará que todo el texto dentro de la etiqueta `body` esté con formato justificado. Pero si se quiere que el texto de los párrafos esté ajustado a la derecha, podemos hacerlo indicando:

```
p { text-align: right ;}
```

Pero si, además, algún párrafo concreto interesa que tenga color rojo se pueden utilizar dos procedimientos para especificar esta propiedad:

- **Las clases.**
- **Los identificadores.**

Podemos crear la clase (`class`) "rojo" para ese párrafo de la forma siguiente:

```
<p class="rojo">Este texto debe salir en rojo.</p>
```

Y en el archivo de estilos CSS utilizamos la propiedad 'color' para asignar el color rojo. Para indicar que es una clase, delante del nombre de la clase se pone un punto (.):

```
.rojo {color:rgb(255, 0, 0);}
```

En el caso de hacer lo mismo utilizando un identificador 'id' escribimos lo siguiente:

```
<p id="rojo">Este texto debe salir en rojo.</p>
```

Para indicar que es un identificador ID, en la hoja de estilos, delante del nombre de la clase se pone una almohadilla '#':

```
#rojo {color:rgb(255, 0, 0);}
```

Y comprobamos que, ambos casos, el resultado obtenido es visualizar el contenido de dicho párrafo de color rojo.

Pero aunque, aparentemente, ambos métodos —las clases y los ID— sirven para lo mismo (asignar propiedades CSS a elementos HTML), hay que tener en cuenta que se utilizan para cosas diferentes.

¿Cuándo debemos utilizar clases o identificadores?

1. **Un ID solo debe ser utilizado una vez en el documento.** Cuando se asigna un ID a un elemento,

no se puede volver a asignar a otro elemento de la misma página.

2. Una clase se puede utilizar las veces que se quiera dentro del mismo documento.

Como resumen podemos decir que, si en un documento web hay una serie de propiedades comunes en un conjunto de elementos de la misma página, se deben utilizar clases.

Si dentro de esos elementos, hay uno que se quiere destacar de alguna forma una única vez en la página, entonces se debe utilizar un ID.

ENUNCIADO

Se propone la creación de una página web personal pero de temática genérica. En nuestro caso, hemos seleccionado el tema medio ambiental. Puede realizarse utilizando otra temática de soporte.

La captura siguiente muestra el aspecto que se sugiere para dicha página web. Se pueden realizar aportaciones pero, básicamente, debe contener al menos los elementos que se indican en ella.



El estilo de la página irá como archivo externo.

Tiempo de resolución: 2 h aprox.

CLAVES DE RESOLUCIÓN

1. En la cabecera (`header`, no `head`) hay que incluir la imagen de fondo junto con los textos de encabezamiento.
2. Establecemos el tamaño de la página (`Page`) en 1000px.
3. En menú de navegación (`nav`) se establecerán los colores de fondo, tipo y color del texto, justificación del texto, así como las distancias a respetar respecto del marco (`padding`).
4. Para los enlaces se pueden cambiar algunas propiedades. Por ejemplo, cuando pasa el puntero del ratón sobre un enlace.

Los estilos para el contenido, y dentro de él para los diferentes enlaces —un artículo por cada imagen enlazada— se dejan de libre elección (siempre que la estructura sea similar a la presentada en la captura). Se debe establecer un ancho para la caja de contenido de forma que las tres imágenes quepan en los 100 px definidos en la página completa.

5. Se debe establecer un estilo del pie (footer) similar al mostrado, y añadir los enlaces a los diferentes documentos web que habrá que preparar o a los sitios web indicados.

SOLUCIÓN

Utilizando Visual Studio Code, introduce el código HTML siguiente (archivo **UD2_CPE2.html**) y la hoja de estilos correspondiente (archivo **UD2_CPE2_estilos.css**). Estos archivos los puedes encontrar dentro del fichero comprimido de la solución que puedes descargar en la siguiente pantalla.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <title>Titulo</title>
    <meta name="autor" content="Nombre_alumno">
    <link rel="stylesheet" href="css/UD2_CPE2_estilos.css">
  </head>
  <body>
    <!-- HEADER - definimos la cabecera de la página -->
    <div id="header">
      <hgroup>
        <h1>Página web personal</h1>
        <h2>Tema: medio ambiente</h2>
      </hgroup>
    </div>
    <!-- Definimos la página -->
    <div id="page">
      <!-- NAV establecemos el menú de navegación de la pagina -->
      <div id="nav">
        <ul>
          <li><a href="https://www.miteco.gob.es/es/biodiversidad/temas/espacios-protegidos/espacios-naturales-protegidos/">Espacios naturales</a></li>
          <li><a href="https://es.wikipedia.org/wiki/Cambio_clim%C3%A1tico">Cambio climático</a></li>
          <li><a href="https://www.proteccioncivil.es/coordinacion/gestion-de-riesgos/incendios-forestales">Incendios forestales</a></li>
          <li><a href="https://ayudaenaccion.org/ong/blog/sostenibilidad/tipos-contaminacion-ambiental/">Contaminación ambiental</a></li>
          <li><a href="https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/prevencion-y-gestion-residuos/">Gestión de residuos</a></li>
          <li><a href="https://www.bbc.com/mundo/topics/cjgn709jk16t">Artículos de prensa</a></li>
        </ul>
      </div>
    </div>
  </body>
</html>
```


Utilización de formularios

ENUNCIADO

Se propone la creación de una página web dedicada al registro de los usuarios a una revista (en nuestro caso hemos utilizado como ejemplo la publicación Muy Interesante).

Hay que crear un formulario de registro con los campos siguientes:

- Nombre, apellidos, teléfono de contacto.
- Dirección particular (calle, número, código postal, población y provincia).
- Un correo electrónico.
- Si se dispone de web, la URL correspondiente.
- Contraseña para acceder a la web de la revista como usuario registrado.
- Repetición de la contraseña.
- Valoración de la revista (del 1 a 10).
- Si se dispone de suscripción a la revista, indicar la fecha de inicio y un par de opciones sobre la recepción de información acerca de los productos y servicios que ofrece la empresa, así como ofertas o promociones.

El botón ENVIAR del formulario con el método POST no realizará ninguna tarea, pero se supondrá que existe un programa en JavaScript que llevará el control de los datos introducidos antes de enviarlos al servidor.

El estilo de la página irá como archivo externo. Tiempo de resolución: 2 h aprox.

CLAVES DE RESOLUCIÓN

- Los campos nombre, apellidos, etc, serán de tipo 'text'.
- El campo correo electrónico será de tipo 'email'.
- El campo para la página web será de tipo 'url'.
- El campo para la contraseña será de tipo 'password'.
- El campo para la valoración será de tipo 'range'.
- El campo para la fecha será de tipo 'date'.
- Y los campos para las ofertas y promociones serán de tipo 'checkbox'.
- Para las comprobaciones sobre los datos introducidos en todos ellos, se realizaría un script del que se incluirá la llamada, pero no el contenido.
- Respecto a los estilos aplicados, añadir al comienzo el selector universal.

```
* { margin:0; padding: 0; }
```

- Dar un estilo al formulario globalmente y luego ir concretando los estilos para cada uno de los elementos del formulario.

El aspecto del formulario será muy sencillo. Se puede tomar como ejemplo la siguiente captura:



Muy Interesante: Registro

Nombre:

Apellidos:

Teléfono:

Dirección particular

Calle/Plaza, num.:

Localidad:

Código postal:

Provincia:

Correo electrónico:

Indica si tienes página web:

Contraseña:

(al menos 6 caracteres, sin espacios)

Repite contraseña:

En una escala de 1 a 10, indica cuál es tu valoración de la revista:

Fecha de inicio de la suscripción:

Deseo recibir información sobre los productos y servicios de ZINET MEDIA GLOBAL, S.L.

☐

Deseo recibir información sobre ofertas y promociones de nuestros Socios Comerciales.

☐

SOLUCIÓN

Los archivos con la solución ([UD2_CPE3.html](#) y [UD2_CP3_estilos.css](#)) puedes descargarlos en la siguiente pantalla:

Audio y vídeo en HTML 5 y CSS3

INTRODUCCIÓN TEÓRICA

En el apartado 3 de la Unidad 2, vimos los principales elementos de HTML 5. La tabla siguiente muestra otros elementos de HTML 5 que podemos utilizar:

Elemento	Descripción
img	<p>Inserta una imagen en una página web. Es obligatorio utilizar el atributo src para especificar el path o ubicación del archivo de imagen que se quiere insertar. Se puede escribir un texto alternativo con alt y dar dimensiones con los atributos height y width.</p> <pre></pre>
blockquote	Contiene un bloque de texto con sangrado.
q	Contiene una cita y el navegador añade las marcas de citación.
br	Inserta una línea en blanco. No tiene etiqueta de cierre.
div	Permite agrupar varios elementos de bloque (párrafos, encabezados, listas, tablas, otras divisiones, etcétera). A menos que se le dé estilo a la división, el navegador no muestra nada. No hace falta definir divisiones cuando el grupo es de un solo elemento.
video	<p>Permite incrustar vídeos en la página web. Puede reproducirse con autoplay, en bucle con loop, controlando la reproducción con controls, etc.</p> <pre><video src="video.mp4" width="640" height="480" controls></video></pre>
audio	<p>Permite insertar archivos de audio en diferentes formatos, como mp3 y ogg. También dispone de una interfaz de control sobre la reproducción del mismo con una API en JavaScript sin necesidad de plugins.</p> <pre><audio id="player" src="archivo.ogg" controls></audio></pre>
canvas	Es un nuevo elemento de HTML5 y proporciona una API para dibujar formas, imágenes, textos, crear gradientes, aplicar transformaciones, etcétera. El llamado lienzo (canvas) es también como un rectángulo en la página web en la que, con scripts de JavaScript, por ejemplo, se puede dibujar.
mark	<p>Permite representar un texto resaltado porque tiene una importancia destacada.</p> <pre><p>Algunos elementos de HTML5 como <mark>aside</mark> tienen una importancia especial por la funcionalidad que introducen.</p></pre>
embed	Facilita un punto de integración en la página para una aplicación externa o un contenido interactivo, como un plug-in.

Tabla 1. Otros elementos de HTML 5.

En este caso práctico se va a trabajar con los elementos HTML 5 de audio y vídeo. Y respecto a CSS3, se va a diferenciar entre los conceptos de transición y animación.

EL ELEMENTO VÍDEO

Integrar un elemento de vídeo en un documento HTML 5 es muy sencillo. La sintaxis utilizada es la siguiente:

```
<video src="video.mp4" width="640" height="480" controls></video>
```

En la etiqueta `<video>` se debe integrar la referencia al recurso `src`. Existen también una serie de atributos que permiten la gestión de dicho elemento. La tabla siguiente los describe:

Atributo	Función
width/height	Permiten especificar al navegador las dimensiones del vídeo. Si no se indican, se toma por defecto la información del tamaño contenida en el archivo de vídeo. El navegador ajustará el tamaño automáticamente en caso de que solo se especifique uno de los valores.
controls	Permite activar la barra de control del navegador web. Alternativamente también es posible definir sus controles propios a través de JavaScript.
poster	Permite incluir un archivo de imagen que se utilizará como vista previa del vídeo. Si no existe se utilizará el primer fotograma del vídeo.
autoplay	Indica al navegador que debe reproducir el vídeo automáticamente, una vez se finalice la carga del mismo.
loop	El vídeo se reproducirá en bucle.
muted	Este atributo se utiliza para silenciar el tono del vídeo.
preload	Indica al navegador cómo debe ser cargado el vídeo una vez finalizada la descarga de la página web. Las opciones para ello son: <ul style="list-style-type: none"> • auto (se carga completamente). • preload (el vídeo carga determinados metadatos). • none (impide la carga previa de datos de vídeo).

Tabla 2. Atributos del elemento vídeo.

EL ELEMENTO AUDIO

Integrar un elemento de audio en un documento HTML 5 es muy sencillo. La sintaxis utilizada es la siguiente:

```
<audio src="audio/burbujas.mp3" width="300" height="168" controls> </audio>
```

En la etiqueta `<audio>` se debe integrar la referencia al recurso `src` indicando la ubicación del archivo de audio. Entre la etiqueta de apertura y cierre, antes del cierre se escribe el texto que se mostrará en caso de que el navegador del usuario no soporte el audio.

Existen también una serie de atributos que permiten la gestión de dicho elemento. La tabla siguiente los describe:

Atributo	Función
width/height	<p>Permiten especificar al navegador las dimensiones del audio. Si no se indican, se toma por defecto la información del tamaño contenida en el archivo de audio.</p> <p>El navegador ajustará el tamaño automáticamente en caso de que solo se especifique uno de los valores.</p>
controls	Permite activar la barra de control reproductor de sonido en la página para el usuario interactúe. Alternativamente, también es posible definir sus controles propios a través de JavaScript.
autoplay	Indica al navegador que debe reproducir el audio automáticamente una vez se finalice la carga de la página, sin esperar al usuario.
loop	El audio se reproducirá en bucle indefinidamente.

Tabla 3. Atributos del elemento audio.

TRANSICIÓN CSS FRENTE A ANIMACIÓN

No hay que confundir transición CSS con animación CSS. Existen algunas diferencias entre ellas:

1. Las transiciones CSS requieren un disparador como, por ejemplo, hacer clic en un elemento. Las animaciones no requieren activación. De forma predeterminada, una animación inicia automáticamente su secuencia cuando se carga la página.
2. Las transiciones solo pueden pasar de un estado inicial a un estado final. No se pueden especificar puntos intermedios, como sí se puede hacer con la animación.
3. La transición solo se ejecuta una vez cuando se activa. No existe un ciclo de transición y no tienen propiedades que indiquen cuántas veces pueden ejecutarse o en qué dirección pueden ejecutarse. Las animaciones, en cambio, tienen las propiedades `animation-iteration-count` y `animation-direction`, que sí lo permiten.

En este caso práctico extendido vamos a utilizar las transiciones para hacer rotar sobre el eje Y el logo de la página cuando el cursor se pasa por encima (`hover`).

En concreto, la rotación es de 180°. A la rotación se le pueden añadir efectos de sombras o resplandor con `box-shadow`. En nuestro caso:

```
box-shadow: 0px 0px 15px 15px darkolivegreen;
```

Y que dicha sombra tenga forma, por ejemplo, redonda siendo la imagen cuadrada. En nuestro caso se consigue con:

```
border-radius:50%;
```

ENUNCIADO Y CLAVES DE RESOLUCIÓN

Retomamos el caso práctico extendido **Creación de una página web**, y vamos a completarlo.

1. En el menú de navegación, incluiremos la opción de acceso a los vídeos y audios que ofrece la página personal sobre el medio ambiente.
2. En el menú de pie de página, incluiremos una opción de Contacto, que ofrecerá la posibilidad de escribir un correo electrónico a través de un pequeño formulario.
3. El logo incluido en la cabecera de la página rotará haciendo uso de la propiedad CSS3 de la transición.
4. Las páginas accedidas desde la principal (Acerca de mi, Copyright, etc.), que en el caso práctico extendido **Creación de una página web** no guardaban el aspecto de la página principal, ahora sí lo tendrán. De esa forma, el sitio web presentará un aspecto más uniforme.

SOLUCIÓN



APOYO MULTIMEDIA

En el vídeo siguiente encontrarás claves adicionales para resolver este Caso Práctico.

Los archivos incluidos en el solucionario están documentados para que queden claras las aportaciones realizadas, con el objetivo de cubrir los puntos del enunciado.

Dichos archivos los puedes descargar en la pantalla que sigue justo a continuación del vídeo, y son los siguientes:

[UD2_CPE4.html](#)

[UD2_CPE4_videos.html](#)

[CPE4_Politica_cookies.html](#)

[CPE4_copyright.html](#)

[CPE4_Contacto.html](#)

[CPE4_Acerca_de_mi.html](#)

[UD2_CPE4_estilos.css](#)

[UD2_CPE4_contacto.css](#)

