

DISEÑO DE INTERFACES WEB
TÉCNICO EN DESARROLLO DE APLICACIONES WEB

Uso avanzado de hojas de estilo

ÍNDICE

/ 1. Introducción y contextualización práctica	4
/ 2. Introducción a CSS	5
2.1. Hojas de estilo externas	5
/ 3. Selectores	6
3.1. Selectores basados en etiquetas	6
3.2. Selectores de clase	6
3.3. Identificadores	7
3.4. Pseudoclases	7
/ 4. Propiedades	7
4.1. Propiedades de texto	7
4.2. Propiedades de fuente	9
4.3. Flotar y posicionar	10
4.4. Propiedades de color y fondo	10
/ 5. Caso práctico 1: “Diseño de una hoja de estilo desde cero”	11
/ 6. Medios	12
6.1. @media	12
6.2. @import	13

ÍNDICE

/ 7. Páginas fijas, líquidas, elásticas y sensibles	13
7.1. Páginas fijas	13
7.2. Páginas líquidas o fluidas	13
7.3. Páginas elásticas	14
7.4. Páginas sensibles. Responsive web design	14
 / 8. Caso práctico 2: “Diseño de una página responsive”	15
 / 9. Resumen y resolución del caso práctico de la unidad	16
 / 10. Webgrafía	17

OBJETIVOS

Reconocer la posibilidad de modificar las etiquetas HTML.

Definir estilos de forma directa.

Definir y asociar estilos globales en hojas externas.

Identificar las distintas propiedades de cada elemento.

Crear clases de estilos.

Analizar y utilizar selectores básicos y avanzados.

Identificar las propiedades de cada elemento.

Crear clases e identificadores de estilo.

Utilizar y actualizar las guías de estilo.

/ 1. Introducción y contextualización práctica

Para diseñar un sitio web es deseable que además del contenido y las relaciones establecidas entre los elementos, el diseño del sitio sea atractivo y consistente, es decir, si escogemos un tipo de tipografía que va a caracterizar nuestra página web, es conveniente que esta se mantenga en todo el sitio.

Si cada título presentara un tamaño, diseño, tipo de letra... uniformes, puede provocar en el usuario, y cliente del sitio, la sensación de estar navegando en una página cuidada, lo que le puede generar la misma impresión sobre los productos que se muestran en el sitio.

En este tema analizaremos en detalle las hojas de estilo, cómo crear hojas en CSS desde cero, así como enlazarlas con el sitio web que se está desarrollando.

El conjunto de propiedades sobre las que se podrá definir el formato del sitio es muy amplio, a lo largo de este tema veremos y practicaremos con algunas de las más importantes, como las propiedades de texto, de fuente o de fondo.

Además, CSS permite aplicar el estilo a diferentes bloques de contenido. Esta selección se lleva a cabo a través de los llamados selectores (básicos, de clase o identificadores).

En el siguiente audio, se propone el diseño del prototipo para el caso visto en el tema anterior, y constituirá el caso práctico de este tema.

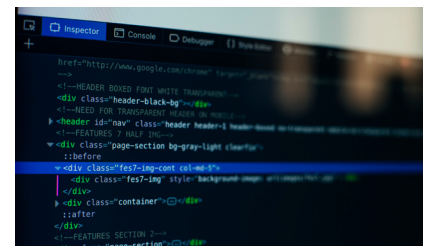


Fig. 1. Pantalla con fichero HTML de ejemplo, se subraya el uso del selector class.



Audio intro. "¿Es necesario crear ficheros específicos de hoja de estilo CSS?"

<https://bit.ly/3etS4ID>





/ 2. Introducción a CSS

Las hojas de estilo CSS (Cascading Style Sheets) permiten definir un conjunto de reglas de estilo que son aplicadas sobre los elementos de una página web en HTML. A través de CSS es posible establecer un mismo estilo para los mismos bloques, aportando así uniformidad al diseño del sitio y simplificando su programación y posterior mantenimiento. En los ficheros de hojas de estilo se recoge todo el código encargado de dar formato a un mismo sitio web, por lo tanto, podemos decir que en estas se define todo lo relativo al aspecto de una página web: colores, tipografías, etc. Para incluir el código CSS, es posible utilizar un fichero externo, como se verá en el apartado siguiente, pero también puede colocarse el fragmento de código de formato dentro del propio archivo HTML. Concretamente, se incluye en la zona **<head>**, bajo una nueva etiqueta, **<style type="text/css">**. A continuación, se indicaría el nombre del **elemento sobre el que se va a aplicar** el estilo, en este caso sobre **<body>**:

```
<style type="text/css">

    body {

        color: purple;

        background-color: #d8da3d

    }

</style>
```

Código 1. Ejemplo de código CSS dentro del fichero HTML.



Audio 1. "XLS. Hojas de estilo extensibles"

<https://bit.ly/3dsSxCM>



2.1. Hojas de estilo externas

En la actualidad, resulta más aconsejable el uso de hojas de estilo elaboradas en un documento diferente, es decir, hojas de estilo externas. En este caso, el código encargado de dar formato a un sitio web se encuentra en un fichero externo. Mientras que en el apartado anterior se incluía el código de formato en el mismo HTML, ahora se incluye un vínculo a través de la etiqueta **<link>** que apunta al fichero CSS. Es posible enlazar con tantas hojas de estilo como se deseen.



Fig. 2. Llamada HTML a CSS externa.



Ejemplo de la sintaxis de la llamada:

```
<link rel="" type="text/css" href="nombre.css" media="">
```

- **rel:** Tipo de relación entre el recurso enlazado (archivo CSS) y el fichero HTML. Para los archivos CSS, el valor es stylesheet.
- **type:** Tipo de recurso enlazado en los archivos CSS, el valor es text/css.
- **href:** Ruta del archivo CSS, esta puede ser relativa o absoluta.
- **media:** Tipo de medio en el que se van a aplicar los estilos del archivo CSS.

/ 3. Selectores

Las reglas de estilo se aplican sobre los bloques definidos con etiquetas HTML. Para seleccionar sobre cuáles aplicarlos exactamente, encontramos varias formas, que se describen a continuación.

3.1. Selectores basados en etiquetas

Los selectores basados en etiquetas son los más simples, puesto que basta con **indicar el tipo de elemento sobre el que se va a aplicar el estilo**. Podemos decir que actúa de forma genérica dando estilo a todos los elementos de un mismo tipo.

```
p{  
  
    propiedad:valor;  
  
}
```

Código 2. Selector de etiqueta. Ejemplo sobre elemento <p>.

3.2. Selectores de clase

Este tipo de selectores aplican su estilo a aquellos **bloques que aparecen identificados en HTML** con un determinado valor para el atributo class. En el siguiente fragmento de código, el valor de la propiedad se aplica sobre el primer elemento tipo p, pero no sobre el segundo.

```
.clase{  
  
    propiedad:valor;  
  
}  
  
<p class="clase"></p>  
  
<p class="clase2"></p>
```

Código 3. Selector de clase. Ejemplo sobre elemento <p>.



3.3. Identificadores

De forma similar a los selectores de clase, los de tipo identificador **utilizan el atributo id para asignar un identificador único** a los bloques en HTML.

```
#identificador{  
    propiedad:valor;  
}  
  
<p id="identificador"></p>  
  
<p id="identificador2"></p>
```

Código 4. Uso de identificadores. Ejemplo sobre elemento <p>.

3.4. Pseudoclases

En algunos casos, un mismo elemento puede presentar varios estados y es deseable que para cada uno de estos se muestre un formato diferente. Uno de los casos más habituales es el del **elemento enlace <a>** que presenta diferentes estados como, por ejemplo:

- **a:link:** Estado para enlaces que no han sido visitados.
- **a:visited:** Estado para enlaces que ya han sido visitados.
- **a:hover:** Estado destinado a detectar si un elemento apuntador (como el puntero de un ratón) ha pasado por encima del enlace.



Audio 2. "Ampliación del estudio de selectores::pseudoelementos"
<https://bit.ly/2Bu4z2b>



/ 4. Propiedades

Las hojas de estilo en CSS permiten definir todo lo relacionado con el formato de un sitio web, es decir, actúan sobre las propiedades de cada uno de los elementos que conforman una página. A continuación, veremos algunas de las propiedades principales de elementos tales como el texto, la fuente tipográfica o las imágenes.

4.1. Propiedades de texto

Estas propiedades permiten controlar aspectos relativos al estilo del texto, en cuanto al interlineado o alineación. El formato referido a la tipografía está recogido en las propiedades de fuente.

- **Alineación del texto text-align.** Define la alineación del texto (left, right, center o justify). Si no se indica ningún valor, se utiliza la alineación a la izquierda por defecto:

```
text-align: center|justify|left|right|initial;
```



- **Interlineado del texto `line-height`:** Define el tamaño de la separación entre líneas de un mismo bloque:

`line-height`: (numero | longitud | porcentaje)

- **Decoración del texto `text-decoration`:** Permite añadir una línea al texto que hace el efecto de subrayado. Es posible determinar el lugar exacto de la ubicación de la línea (debajo, arriba, sobre el texto), así como otras características de diseño. Esta propiedad presenta varias propiedades asociadas que definen la ubicación de la línea, el color y el estilo:

`text-decoration-line` (none | underline | overline | line-through | blink | initial)

`text-decoration-color` (color | rgb() | hsl() | initial)

`text-decoration-style` (solid | double | dotted | dashed | wavy | initial)

En el siguiente ejemplo se muestra el uso conjunto de las propiedades de estilo de texto descritas.

```
@media screen{  
  ul {  
    text-align: center;  
    text-decoration: overline;  
    text-decoration-color: green;  
    text-decoration-style: dotted;  
    line-height: 3em;  
  }  
  p {  
    text-align: left;  
    text-decoration: underline;  
    text-decoration-color: green;  
    text-decoration-style: dashed;  
    line-height: 3em;  
  }  
}
```

Código 5. Ejemplo de código con propiedades de texto.



4.2. Propiedades de fuente

El diseño relativo a la tipografía, es decir, a la fuente que se va a utilizar en un sitio web es uno de los factores más importantes en el diseño, ya que va a determinar en gran medida la calidad de lectura del usuario. Para ello, tenemos:

- **Familia de fuente font-family:** Describe la familia tipográfica que va a presentar el texto. Es aconsejable indicar varios tipos de fuentes, puesto que si alguna de ellas no es reconocible por el navegador, se utilizará la siguiente. Por esta razón, es aconsejable indicar por defecto alguna de las fuentes conocidas como genéricas (courier, arial, sans-serif...).
- **Tamaño de fuente font-size:** Define el tamaño de fuente del elemento de texto sobre el que se define el formato (50%, smaller, larger, 16pt...).
- **Grosor de fuente font-weight:** Se utiliza para indicar el grosor de la letra. Es útil para simular el texto en negrita (normal, bold, lighter, bolder).
- **Estilo de fuente font-style:** Define la inclinación del texto, es decir, crea efecto de cursiva (normal, italic, oblique).
- **Formato de fuente font-variant:** Define el estilo de la fuente entre normal y small-caps.

Además de las anteriores, existe una propiedad fundamental en cuanto a las propiedades de fuente llamada font. Consiste en una propiedad combinada, es decir, a través de esta única regla es posible definir todos los rasgos de estilo descritos, siempre y cuando se respete un orden concreto. La sintaxis exacta es:

```
font: font-style font-variant font-weight font-size font-family;
```

En el siguiente ejemplo, para los elementos del listado, se establece un determinado estilo de fuente a través de la propiedad fundamental font, mientras que para los elementos de tipo párrafo se han definido de forma individual los parámetros de estilo de fuente.

```
ul {  
  text-align: center;  
  font: italic small-caps bold larger sans-serif;  
}  
p {  
  text-align: left;  
  font-style: oblique;  
  font-size: smaller;  
  font-family: arial | courier;  
}
```

Código 6. Ejemplo de código con propiedades de fuente.



Vídeo 1. "Creación de una hoja de estilo CSS. Propiedades básicas y catálogo de estilos"

<https://bit.ly/317XU8t>



4.3. Flotar y posicionar

La propiedad float **permite definir la posición relativa de una capa con respecto al resto**. Por defecto, cuando estas se crean lo habitual es que se sitúen de forma vertical. Pero, en ocasiones, los criterios de diseño establecen que unas se adapten a otras, ofreciendo así una interfaz más uniforme puesto que los elementos no aparecerán como entes independientes, sino que se mostrarán cohesionados. Para lograr este efecto se utiliza la propiedad de flotación, es decir, el elemento de estilo float.

```
float: left | right | none;
```

En la siguiente imagen, en el primer caso se muestra la salida si para los elementos de tipo se utiliza la opción float a la derecha. La segunda imagen corresponde a la salida estándar de posicionamiento si no se utiliza la flotación.



Fig. 3. Ejemplo de presentación usando la propiedad de flotación.

4.4. Propiedades de color y fondo

Las propiedades de estilo también permiten definir el formato de fondo de la página al completo, o de cada uno de los bloques de forma individual.

Es posible indicar que todo el cuerpo sea de un color o definir, para cada tipo de elemento, un color de fondo distinto. Las principales propiedades son:

- **Color de fondo background-color:** Define el color de fondo de un determinado bloque o elemento (color, transparent, inherit).

```
background-color: color | rgb() | #códigoHex;
```

- **Imagen de fondo background-image:** Similar a la propiedad anterior, pero en este caso se define una imagen como fondo.

```
background-image: url | none | inherit;
```

- **Posición de la imagen background-position:** Determina la posición de la imagen con respecto al elemento sobre el que se define (bottom left, right top, 30%, 70%...).

```
background-position: posición(X e Y);
```

- **Repetición de la imagen background-repeat:** Determina la posición de la imagen con respecto al elemento sobre el que se define.

```
background-repeat: repeat | repeat-x | repeat-y | no-repeat | inherit;
```



/ 5. Caso práctico 1: “Diseño de una hoja de estilo desde cero”

Planteamiento: Un centro de formación nos encarga diseñar un sitio web donde se describan los módulos para el ciclo Formativo de Grado Superior: Desarrollo de Aplicaciones Web. Se necesita algo sencillo y simple, pero teniendo la autonomía completa sobre el diseño, es decir, tenemos que realizar el trabajo de forma manual diseñando nuestro propio fichero CSS. Un ejemplo de lo que se necesitaría puede ser como el de la imagen:

Desarrollo de Aplicaciones Web		
Módulos por curso para el Ciclo Formativo de Grado Superior: Desarrollo de Aplicaciones Web		
1º CURSO	2º CURSO	
istemas informáticos	5h/semana Desarrollo web entorno cliente	6h/semar
ases de datos	5h/semana Desarrollo web entorno servidor	8h/semar
rogramación	7h/semana Despliegue de aplicaciones web	5h/semar
enguaje de marcas y sistemas de gestión de la	4h/semana Diseño de interfaces web	6h/semar
nformación	3h/semana Empresa e iniciativa	3h/semar
ntornos de desarrollo	emprendedora	

Fig. 4. Diseño de interfaz con CSS.

Nudo: En el fichero HTML, se pueden crear varios elementos:

- La barra de color azul y el nombre que pertenezcan a un div que queden identificados bajo la clase cabecera. En el código CSS se le asignará un estilo preciso a este bloque.
- Se puede crear la tabla con la forma vista en el Tema 2.
- Los elementos de la primera fila se pueden identificar con la clase título, puesto que estos presentarán un estilo diferente al resto de celdas. Deben aparecer centrados.
- Todas las celdas, excepto los títulos, se deben presentar alineadas a la izquierda.

Desenlace: El código CSS podría quedar de la siguiente forma:

```
.cabecera {
  text-align: center;
  color:white;
  text-decoration-line: underline;
  font: italic bold larger arial;
  height: 200px;
  background: blue;
  line-height: 200px;;
}

p {
  text-align: center;
  font-style: oblique;
  font-size: larger;
  font-family: arial | courier;
}

table {
  font-family: courier;
}

.titulo {
  text-align: center;
  font-family: courier;
}
```

Código 7. Diseño de hoja de estilo desde cero.



/ 6. Medios

Las hojas de estilo permiten especificar el formato en función del medio en el que se va a mostrar la interfaz de diseño del sitio web. Es decir, permite definir estilos diferentes para mostrarla por pantalla o ser impresa, por ejemplo.

Para este fin son claves el uso de las reglas **@media** y **@import**. La primera permite definir el tipo de medio sobre el que se va a aplicar la regla de formato, mientras que **@import** permite la asociación de hojas de estilo externas a diferentes medios.

6.1. @media

Esta regla permite la asignación de diferentes estilos a un mismo bloque en función del tipo de medio en el que va a ser visualizado. Sintaxis:

```
@media TIPO_DE_MEDIO{  
    ...CÓDIGO CSS DE ESTILO...  
}  
@media TIPO_DE_MEDIO{  
    ...CÓDIGO CSS DE ESTILO...  
}
```

Por ejemplo, el siguiente código da el formato al contenido impreso, pero no lo puede mostrar a través de la pantalla.

```
@media print{  
    body {  
        color: purple;  
        background-color: #d8da3d  
    }  
}
```

Código 8. Ejemplo de código con la regla @media para canales de impresión.

En la siguiente tabla se recogen algunos de los identificadores correspondientes a los medios o canales de visualización de un sitio web.

Nombre del medio	Campo de aplicación
screen	Pantalla
print	Impresora
handheld	Dispositivos de mano
projection	Elementos de proyección

Tabla 1. Identificador de medios para visualización de contenido web



6.2. @import

La regla @import **permite importar hojas de estilo** ya implementadas y asociarlas a diferentes medios. De esta forma, podríamos crear una hoja de estilo para cada tipo de medio que se vaya a contemplar, y desde la hoja de estilo principal realizar la importación necesaria para cada uno de los casos. La sintaxis básica es la siguiente:

```
@import url_nombreFichero TIPO_DE_MEDIO;
```

Por ejemplo, en el siguiente caso quedaría asociada la hoja de estilo 'A' con la visualización en pantalla, y la hoja de estilo 'B' con la impresión y la visualización en dispositivos de mano:

```
@import "hojaEstiloA.css" screen;  
@import "hojaEstiloB.css" print handheld;
```

Código 9. Ejemplo de código con la regla @import.

/ 7. Páginas fijas, líquidas, elásticas y sensibles

En la actualidad existen multitud de dispositivos en los que visualizar una misma página web (PCs, tablets, móviles...), por lo tanto, a la hora de diseñar una interfaz es conveniente adaptar el estilo en función de las características del medio donde se van a mostrar. Existen diferentes tipos de páginas en cuanto a la adaptabilidad de sus dimensiones: fijas, líquidas o elásticas.

7.1. Páginas fijas

En las páginas con una estructura fija, las dimensiones se expresan utilizando valores absolutos. En este tipo de casos se definen unas dimensiones fijas indicando el tamaño en píxeles en la hoja de CSS:

```
div#cabecera {  
  width: 750px;  
  padding: 5px;  
  border: red solid 1px;  
}
```

Código 10. Código para dimensiones de página fija.

7.2. Páginas líquidas o fluidas

En este segundo caso, no se especifica un valor fijo para los bloques, sino que los elementos contenedores se ajustan a las dimensiones de la ventana, para ello se expresan las medidas relativas a través de porcentajes. En este caso, los bloques varían su tamaño para adaptarse a las dimensiones de la pantalla, evitando la aparición de barras de desplazamiento. Este tipo presenta buenos resultados en pantallas medianas y pequeñas:

```
div#cabecera {  
  border: red solid 1px;  
  padding: 0.67%;  
}
```

Código 11. Código para dimensiones de página líquida.

7.3. Páginas elásticas

Las páginas elásticas, al igual que las líquidas, adaptan su tamaño, pero lo hacen en base a las dimensiones del contenido de los bloques. Para ello se utilizan las unidades de medida relativas em, es decir, si se aumenta el tamaño de la fuente, aumenta el tamaño del bloque que lo contiene. Este tipo de diseño presenta buenos resultados en pantallas relativamente medianas y pequeñas.

```
div#cabecera {  
  
    border: red solid 0.0625em;  
  
    padding: 0.3125em;  
  
}
```

Código 12. Código para dimensiones de página elástica.

En los tres bloques de código anteriores (10, 11, 12), se han dado diferentes formatos a un mismo bloque de contenido, denotado con el identificador cabecera. Los identificadores son un tipo de selectores que se utilizan para distinguir entre bloques del mismo tipo. Es decir, en este caso, buscará entre los elementos div, aquel que esté identificado como cabecera y le aplicará el estilo.

7.4. Páginas sensibles. Responsive web design

Las conocidas como páginas responsivas adaptan su diseño al tipo de dispositivo y a las dimensiones de estos. Suelen ser de gran utilidad, sobre todo, en la actualidad, puesto que la misma página web puede mostrarse en una pantalla de ordenador o en un teléfono móvil. El no respetar las dimensiones de cada uno de estos dispositivos ofrecería resultados irregulares.

Lo habitual en estos casos es utilizar diferentes hojas de estilo adaptadas en función del dispositivo, para esto es útil el uso de la regla @import vista en apartados anteriores.

Otra opción es utilizar las Medias Queries. Estas permiten definir a través de las reglas @media qué estilos han de aplicarse en función de una serie de condiciones. Existen dos formas de aplicarlo:

- **Desde link:** se le indica la carga de la hoja de estilos solo si el dispositivo es una pantalla y el ancho máximo es de 800px. Si no se cumple, se tomarían los datos de otra hoja de estilo.

```
<link rel="stylesheet" media="only screen and (max-width: 800px)" href="estilos_1.css">
```

- **Desde la hoja de estilo:** este tipo supone una ventaja evidente con respecto a la anterior, puesto que en una sola hoja se pueden definir todos los formatos posibles en función del tipo de medio y de las dimensiones de la pantalla. Dependiendo de las dimensiones se tomará un formato u otro.

```
@media tipoDeMedio (max-width:¿?px){  
  
    <!-- Implementación de estilo-->  
  
}
```



Este tipo de reglas permite encadenar diferentes opciones para evaluar la opción de estilo que más se adecúa a cada bloque. En el siguiente ejemplo, el bloque box aparece de color rojo solo mientras las dimensiones sean menores de 600px; si estas se superan, sea cual sea el dispositivo, cambiará a color azul.

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <style type="text/css">
    .box {
      background: red;
      height: 100px;
    }
    @media all and (min-width:600px) {
      .box {
        background: blue;
      }
    }
  </style>
</head>
<body>
  <div class="box">Hola</div>
</body>
</html>
```

Código 13. Código ejemplo Responsive Web Design.



Vídeo 2. "Diseño web responsive"

<https://bit.ly/2No1140>



/ 8. Caso práctico 2: "Diseño de una página responsive"

Planteamiento: Al centro de formación le ha gustado el diseño completado en el caso práctico 1 y nos amplía el contrato para realizar las modificaciones oportunas que hagan que la interfaz del sitio presente diferentes comportamientos en función de las dimensiones y el tipo de dispositivo en el que se vaya a visualizar. Concretamente nos pide que:

- En dispositivos de mano todos los elementos aparezcan alineados a la izquierda.
- En pantalla con dimensiones de ancho superiores a 600px el diseño sea tal y como se realizó en el caso práctico 1.
- Para pantallas con dimensiones inferiores, la altura del banner azul se reduzca a la mitad y sea de color negro. Además, en este caso, la letra debe ser Courier en el encabezado.

Ojo: Ten en cuenta que si se cambia la altura de la banda azul también hay que modificar la posición del título del encabezado (line-height: 100px). Recuerda cambiar también el tamaño de la letra reduciéndolo para que quepa el título completo en la pantalla.

Nudo: En base a las especificaciones anteriores, el código CSS que se ha modificado sería el correspondiente al Código 14, donde se indica que siempre que se visualice en dispositivos de mano el texto estará alineado a la izquierda. En el Código 15 se ha mantenido el fragmento inicial para dimensiones superiores a 600px (min-width) y se ha especificado dentro de la condición de tamaño mínimo. Para dimensiones menores sí se han introducido cambios.



Desenlace:

```
@media handheld {  
  * {  
    text-align: left;  
  }  
}
```

Código 14. Alineación a la izquierda para dispositivos de mano.

```
.cabecera {  
  text-align: center;  
  color:white;  
  text-decoration-line: underline;  
  font: italic bold smaller courier;  
  height: 100px;  
  background: black;  
  line-height: 100px;  
}  
@media all and (min-width:600px) {  
  .cabecera {  
    text-align: center;  
    color:white;  
    text-decoration-line: underline;  
    font: italic bold larger arial;  
    height: 200px;  
    background: blue;  
    line-height: 200px;  
  }  
}
```

Código 15. Ajustes y cambios según tamaño.

/ 9. Resumen y resolución del caso práctico de la unidad

Como hemos visto en este tema, el desarrollo de las hojas de estilo es una de las claves principales en la implementación del formato del diseño que van a presentar los sitios web, y hemos aprendido que es posible incorporar un código CSS dentro de los mismos ficheros en HTML, aunque para simplificar el código y que resulte más sencillo su desarrollo y posterior mantenimiento es aconsejable hacerlo en ficheros externos y enlazarlas a través de la etiqueta <link>. El siguiente fragmento debe pasar a formar parte de tus plantillas de desarrollo en HTML:

```
<link rel="stylesheet" type="text/css" href="nombre.css" media="screen">
```

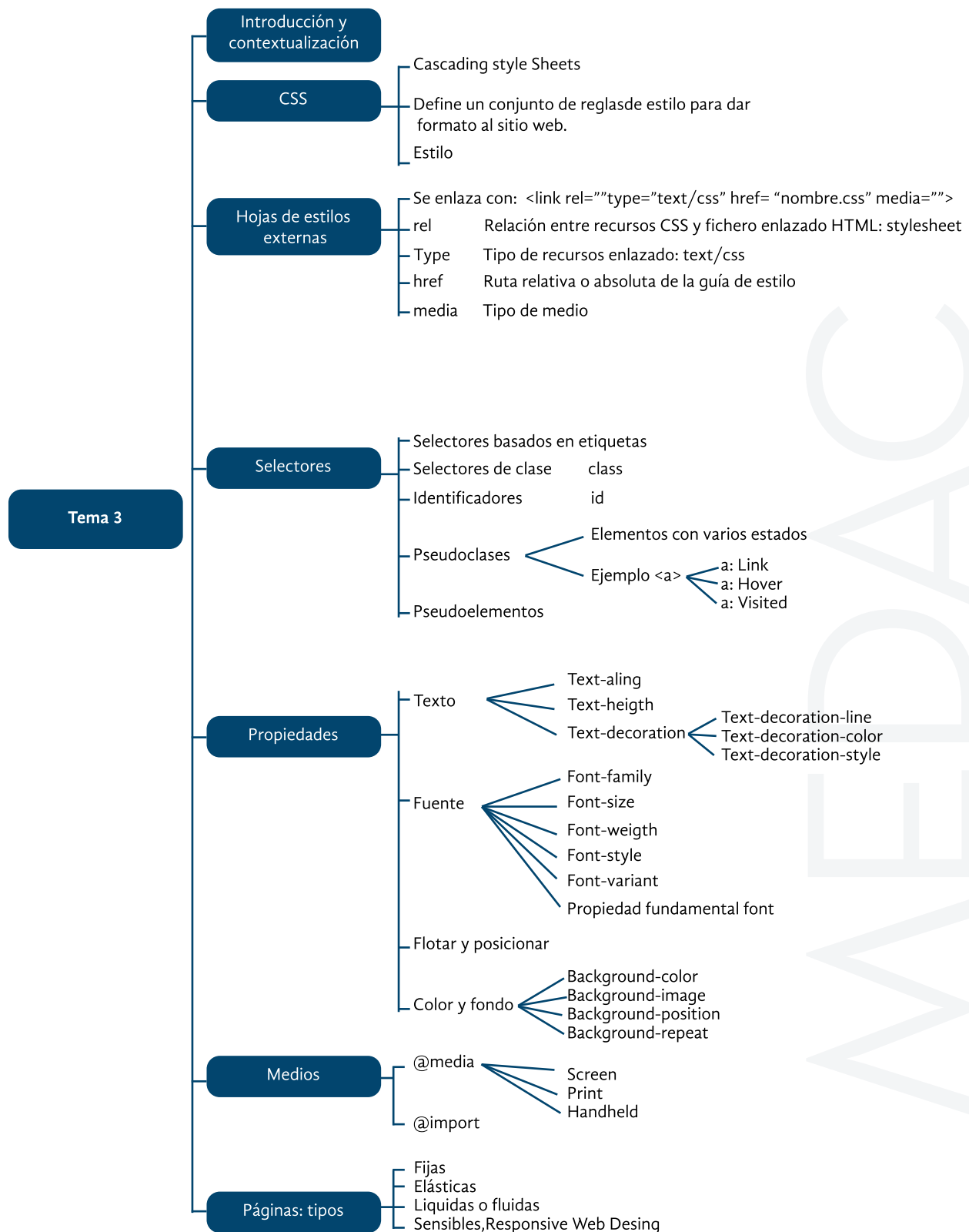
Código 16. Línea completa de código link a ficheros CSS.

Se han repasado también un amplio abanico de propiedades que permiten definir las características de estilo del sitio: de fuente, de texto, color, posición y flotación. Estas se aplican a las etiquetas que indican los selectores, ya sean basados en etiquetas, clases o identificadores.

Por otro lado, en función del tipo de dispositivo será deseable que el sitio web presente una apariencia en concreto, tanto para facilitar la navegación de los usuarios como para evitar que el sitio pierda su formato original. Para ello se utilizan las reglas @media y @import.



Además, se ha visto cómo es posible definir las propiedades de un sitio web para que estas varíen adaptando la interfaz de la página en función las dimensiones del soporte. Este tipo de páginas reciben el nombre de líquidas o elásticas, mientras que las que no varían el tamaño de sus bloques reciben el nombre de fijas. En la actualidad, están cada vez más presentes las de tipo responsive, que utilizan las Medias Queries para adaptar los valores de las propiedades al medio y a las dimensiones de este.





Resolución del caso práctico de la unidad

Como se ha visto en el tema, incorporar un fichero CSS es de gran utilidad, puesto que aquí es donde se recogen todas las directrices de estilo de un mismo sitio web, es decir, si se va a crear una página web con títulos en todas las páginas, o en todos los bloques, es deseable que todos presenten el mismo estilo, no que uno aparezca en negrita, otro en color blanco, otro subrayado de color azul...

Es necesario prestar la atención suficiente a estos aspectos para mantener la uniformidad del sitio, de lo contrario dará la impresión de que el diseño y cuidado de la página web no importa, lo que podría hacer pensar al cliente de ese sitio web, la forma de tratar los productos o servicios que aparecen en esa página, tampoco han sido demasiado cuidados.

Además, como se ha visto en el tema, existe un elemento muy útil, los selectores. Por un lado, los de tipo general son los utilizados de forma genérica para asignar a cada elemento un estilo propio (p, h1,...). Pero, además, si se desea que un conjunto de elementos del tipo que sean aparezcan de color azul, basta con asignarles un identificador o una clase que los distinga del resto y nos permitirá asociarles un estilo característico de clase.

/ 10. Webgrafía

García-Miguel, D. (2019). Diseño de Interfaces Web (1.a ed.). Madrid, España: Síntesis.

Meyer, A. (2007). CSS: The Definitive Guide (3.a ed.). O'Reilly Media, Inc.

Meyer, A. (2007). CSS Web Site Design. Peachpit Press.