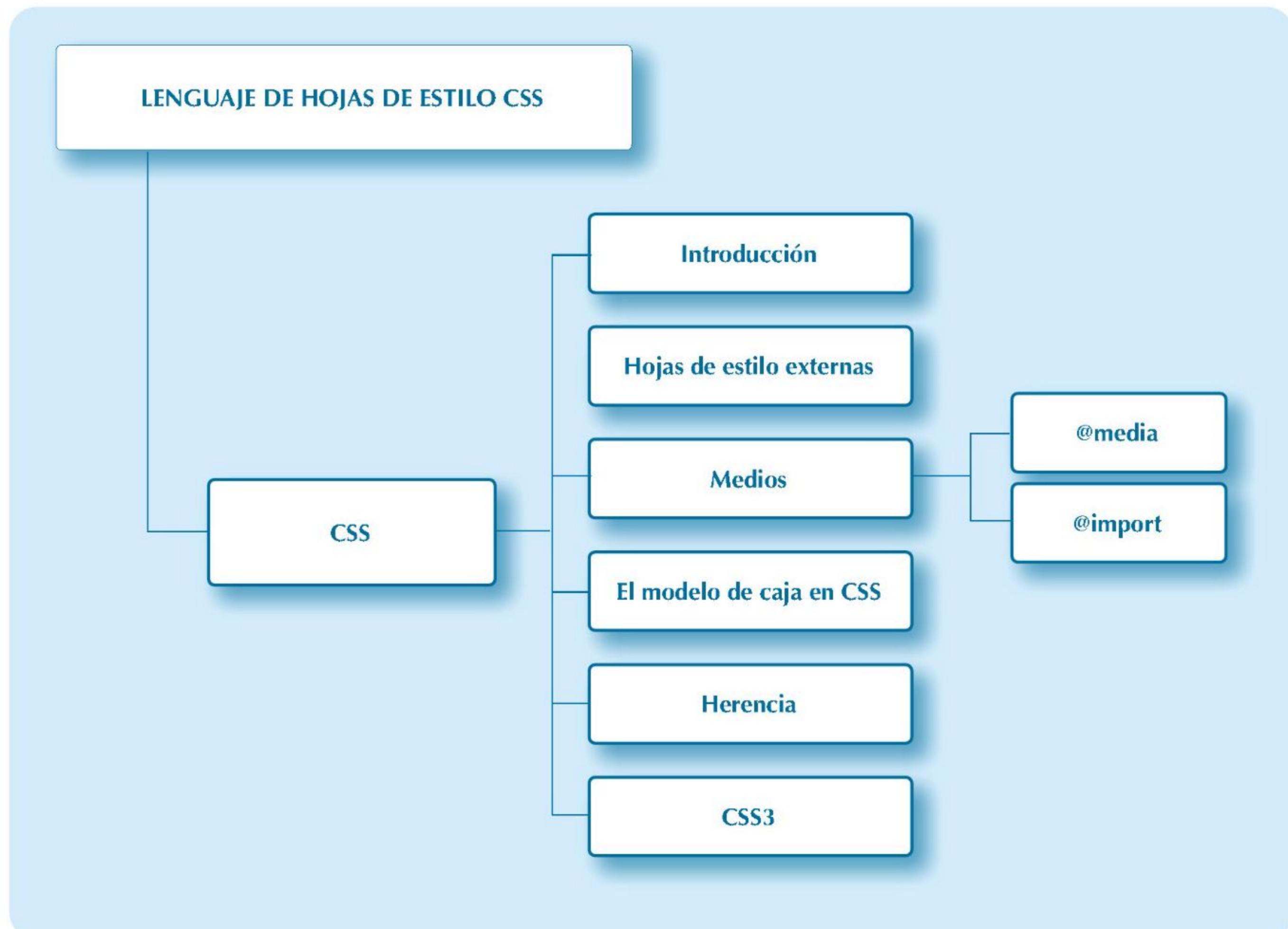


Hojas de estilo CSS

Objetivos

- ✓ Analizar las diferentes opciones para la presentación de la información en documentos web.
- ✓ Valorar la importancia de definir y aplicar una guía de estilo en el desarrollo de una aplicación web.
- ✓ Seleccionar colores y otros elementos de diseño para su visualización en pantalla.
- ✓ Crear y utilizar plantillas de diseño.
- ✓ Reconocer la posibilidad de modificar las etiquetas HTML.
- ✓ Definir estilos de forma directa.
- ✓ Asociar estilos globales en hojas externas.
- ✓ Identificar las distintas propiedades de cada elemento.
- ✓ Crear clases de estilos.

Mapa conceptual



Glosario

CSS. Del inglés, *Cascading Style Sheets*. Lenguaje de hojas de estilo en cuyos ficheros se recoge el código encargado de modificar y dar estilo al sitio web. Es el encargado de controlar el aspecto de las páginas web.

CSS3. Se trata del lenguaje de hojas de estilo en su tercera versión, el cual presenta todas las características de su predecesora y, además, aporta un mayor dinamismo en el diseño de la página web (transparencia en los elementos, esquinas redondeadas, multicolumnas...).

Herencia. Mecanismo mediante el cual es posible que ciertas propiedades de un elemento “padre” se transmitan a sus “hijos”, es decir, que tomen el mismo valor. No todas las propiedades CSS son heredables.

Medios. Canales a través de los cuales se muestra la información del sitio web: auditivos, visuales, impresos, etc. Lo habitual es que se utilice una propiedad definida para cada tipo de canal.

Modelo de cajas. Modelo que simula, a través del uso de etiquetas, lo que se pueden considerar como “cajas”, dotando a cada una de ella de un identificador, indicado por el nombre de la etiqueta. Este sistema de cajas permite, desde el fichero CSS, seleccionar cada caja y dotarle del estilo deseado.

Motor. Es la parte encargada de interpretar el código HTML y CSS. A la hora de diseñar el aspecto de un sitio web, es necesario conocer el motor de cada navegador.

4.1. Introducción a CSS

Cuando se trabaja con páginas web escritas en HTML y XHTML, es deseable que presenten un estilo concreto y un aspecto definido. La forma más eficiente para poder llevar esto a cabo es distinguir entre la parte en la que se sitúan los contenidos, los ficheros programados en HTML, y la parte del proyecto donde se recogen los aspectos relativos a la apariencia. Aquí es donde aparece CSS (*Cascading Style Sheets*), un lenguaje de hojas de estilo que será el encargado de controlar el aspecto de los sitios web (figura 4.1).

El uso de hojas de estilo CSS aporta numerosas ventajas: mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento, tanto si va a ser realizado por uno mismo como por terceros, y, además, facilita la visualización del sitio en dispositivos diferentes.

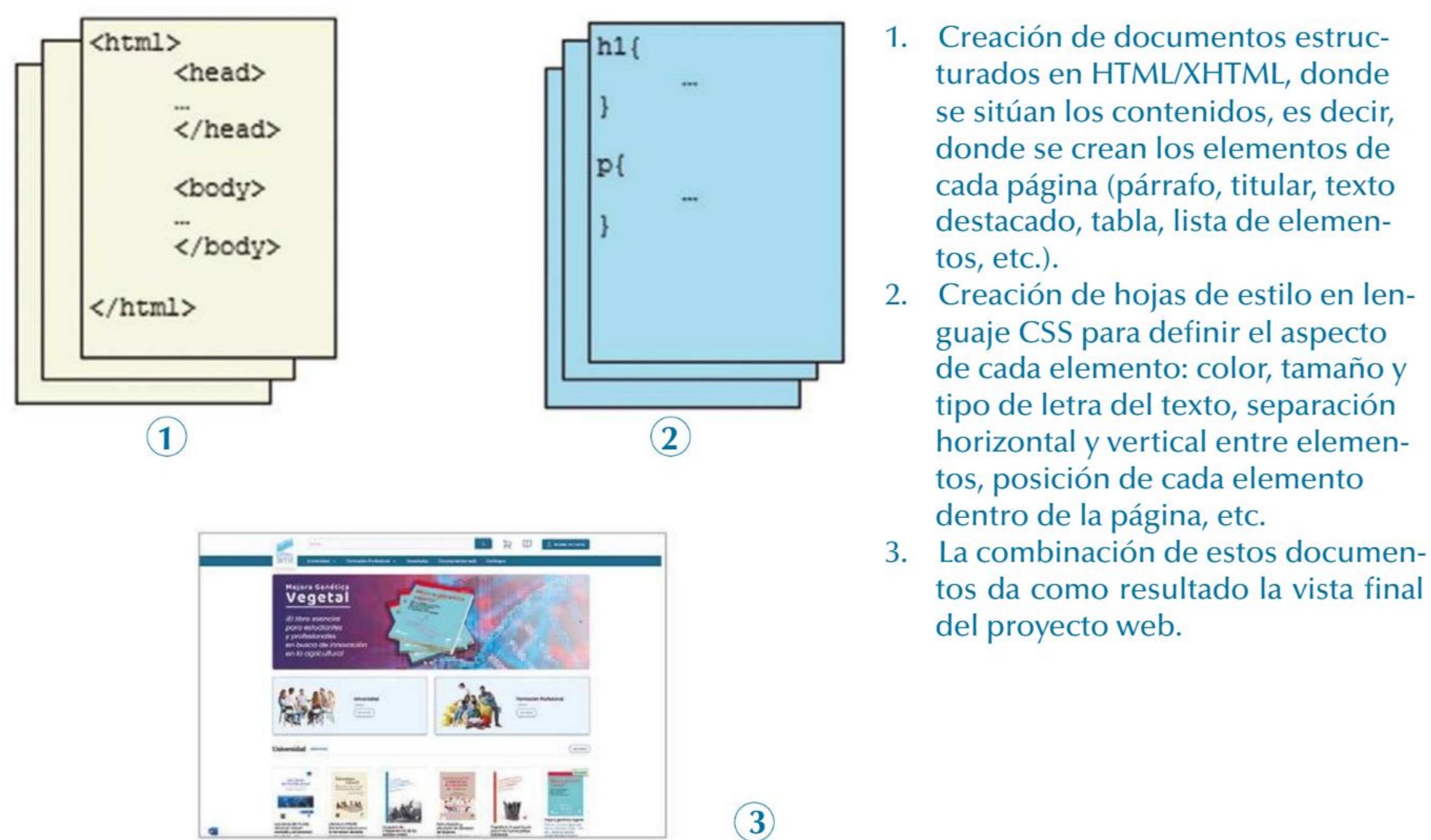


Figura 4.1
Diagrama del proceso seguido para la conexión en HTML y las hojas de estilo CSS.

No todos los navegadores ofrecen el mismo soporte a las guías de estilo. Debe conocerse el motor de cada navegador, dado que es la parte que se encarga de interpretar el código HTML y CSS. En el cuadro 4.1 se muestran los soportes de las diferentes versiones de CSS, que se estudiarán más adelante, en los principales navegadores que se utilizan en la actualidad.

CUADRO 4.1
Características de los motores en los navegadores

Navegador	CSS1	CSS2	CSS3	Motor
Firefox	Completo	Completo	Todos los selectores y pseudoclases, y ciertas propiedades	Gecko
Explorer	Completo	Completo	?	?
Chrome	Completo	Completo	Todos los selectores y pseudoclases, y muchas propiedades	Webkit
Safari	Completo	Completo	Todos los selectores y pseudoclases, y muchas propiedades	WebKit
Opera	Completo	Completo	Todos los selectores y pseudoclases, y muchas propiedades	Presto

Para añadir estilos a una página web bastaría con añadirlos en el código de la propia página web, en el sitio indicado para ello en la cabecera, utilizando las etiquetas `<style>`. Esto puede utilizarse para comenzar a familiarizarse con las hojas de estilo, pero se recomienda realizarlo en documentos aparte, como se explica en el siguiente apartado.

- Ejemplo de código:

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
    <title>
      Ejemplo de estilos CSS en el propio documento
    </title>
    <style type="text/css">
      p {
        color: black; font-family: Verdana;
      }
    </style>
  </head>
  <body>
    <p>Un párrafo de texto.</p>
  </body>
</html>

```

FUNDAMENTAL

El principal inconveniente de no utilizar los documentos CSS y hacerlo desde el propio código en HTML es que, si se quiere hacer una modificación en los estilos definidos, es necesario modificar todas las páginas que incluyen el estilo que se va a modificar.

4.2. Cómo utilizar hojas de estilo externas

En la actualidad se utilizan hojas de estilo externas. En este caso, en lugar de tener el código encargado de modificar el sitio web, este código se encuentra en otro fichero. Desde el fichero principal de la página desarrollada en HTML se accede al fichero en código CSS, a través de la etiqueta `<link>`, la única particularidad que debe tenerse en cuenta es que el fichero donde se definen los estilos tendrá la extensión `.css`. Se pueden enlazar tantos ficheros CSS como se deseen (figura 4.2).

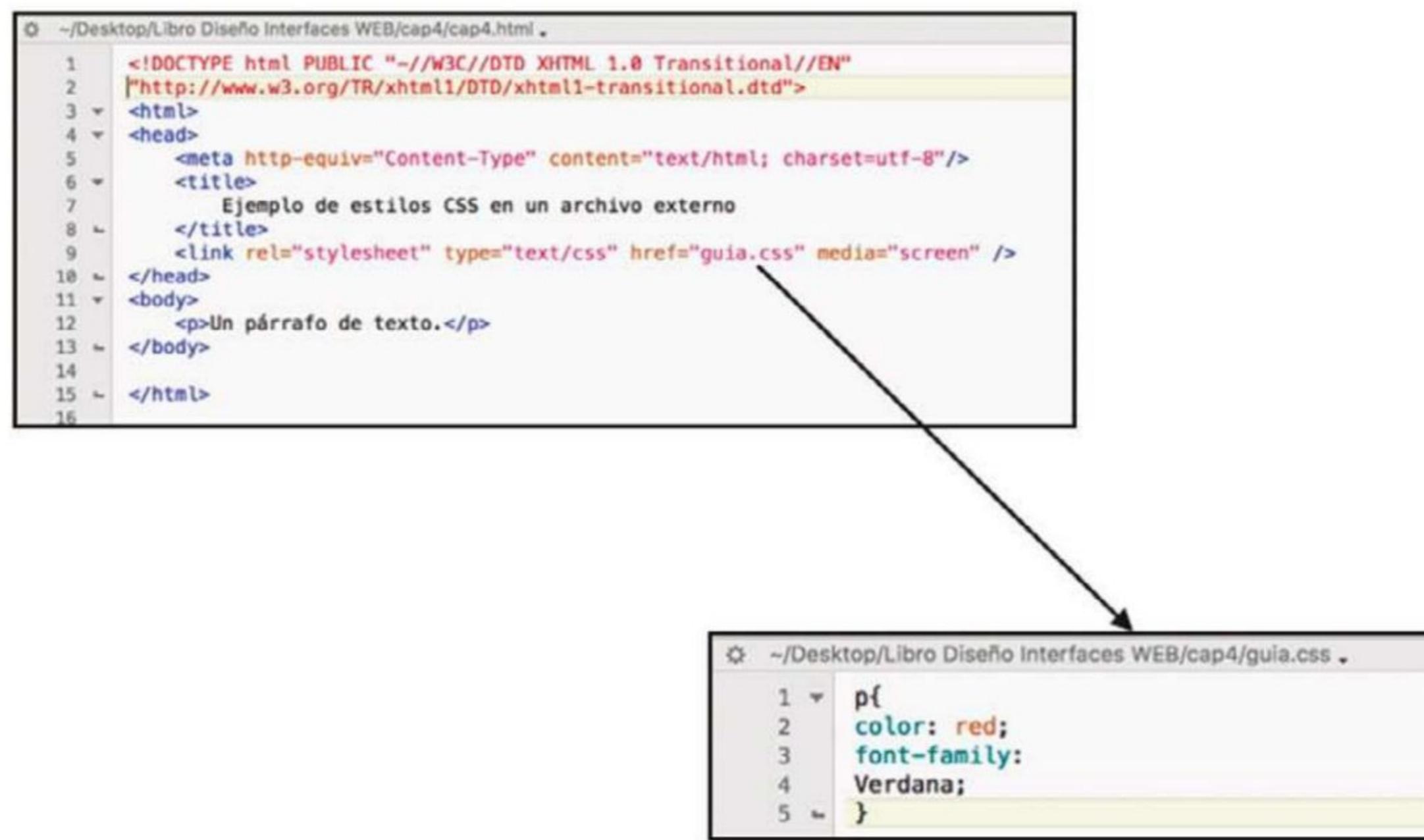


Figura 4.2
Ficheros HTML y CSS en invocación de hojas de estilo externas.

La función de cada uno de los atributos usados para enlazar el documento HTML con la hoja de estilos en CSS es:

- ✓ `rel`. Indica el tipo de relación que existe entre el recurso enlazado, el archivo CSS y la página HTML. Para los archivos CSS, el valor es ‘stylesheet’.
- ✓ `type`. Indica el tipo de recurso enlazado, en el caso de los archivos CSS su valor es ‘text/css’.

- ✓ **href**. Recoge la URL del archivo CSS, esta puede ser relativa o absoluta y puede apuntar a un recurso interno o externo al sitio web.
- ✓ **media**. Indica el tipo medio en el que se van a aplicar los estilos del archivo CSS. Más adelante se explican en detalle los medios CSS y su funcionamiento.

En la actualidad, el uso de CSS en HTML está muy extendido, puesto que a través de un solo archivo CSS es posible incluir una misma guía de estilo en todas las páginas HTML de un mismo sitio web; de esta forma se cumple uno de los principios más importantes perseguidos a la hora de realizar un correo diseño gráfico: que exista uniformidad de estilo entre todas las páginas que componen el sitio. Además, si se hace necesario realizar un cambio común en el estilo de todas las páginas, bastará con hacerlo en la hoja de estilo y se extenderá por todo el sitio web enlazado a este archivo CSS.

- Ejemplo:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>
      Ejemplo de estilos CSS en el propio documento
    </title>
    <link rel="stylesheet" type="text/css" media="screen, handheld"
          href="hojaEst.css">
  </head>
  <body>
    <p>Un párrafo de texto.</p>
  </body>
</html>
```

El código CSS (hojaEst.css):

```
p {
  color: red; font-family: Verdana;
}
```

4.3. Medios: adecuación al medio o dispositivo de visualización

Las hojas de estilo permiten especificar cómo se representa un documento, en función del tipo de medio de visualización. Se denominan *medios* a los canales a través de los cuales se muestra la información del sitio web: auditivos, visuales, impresos, etc. Lo habitual es que se utilice una propiedad definida para cada tipo de canal. En la actualidad, el medio más común es el utilizado para definir el aspecto del sitio web en pantalla, *screen*, y el que define la página para ser impresa, *print*. En el cuadro 4.2 se recogen los tipos más frecuentes.

Para especificar el medio de destino se utilizan varios mecanismos, desde la hoja de estilo en CSS o desde el propio fichero HTML del sitio web.

CUADRO 4.2

Tipos de medios o canales a través de los cuales es posible visualizar el sitio web

Medio	Descripción
<i>all</i>	Todos los medios definidos.
<i>print</i>	Impresoras.
<i>braille</i>	Dispositivos táctiles que emplean el sistema braille.
<i>embossed</i>	Impresoras braille.
<i>handheld</i>	Dispositivos de mano: móviles, PDA, etc.
<i>projection</i>	Proyectores y dispositivos para presentaciones.
<i>screen</i>	Pantallas de ordenador.
<i>speech</i>	Sintetizadores para navegadores de voz utilizados por personas discapacitadas.
<i>tv</i>	Televisores y dispositivos con resolución baja.

4.3.1. @media

La regla `@media` especifica los tipos de medios de destino. A través de esta opción es posible que, en una misma hoja de estilo, aparezcan reglas para distintos medios. Si un mismo estilo se aplica a varios medios, basta con indicar sus nombres separados por comas.

- Sintaxis de `@media`

```

}
@media nombreTipoMedio{
    selector{
        propiedad: valor;
    }
}
```

En el siguiente diagrama de código se muestra un ejemplo de uso de `@media` para indicar el tipo de medio donde se va a visualizar el sitio web. En este caso, se observan dos tipos de medio, impresión y visualización por pantalla. El tamaño de la letra se especifica para cada medio por separado, mientras que para el interlineado se va a utilizar el mismo en ambos casos; por lo tanto, se puede indicar de forma conjunta.

- Ejemplo:

```

@media print {
    h1 {
        font-size: 15pt
    }
}
```

```
}

@media screen {
    h1 {
        font-size: 15px
    }
}

@media screen, print {
    h1 {
        line-height: 1.2
    }
}
```

Actividad propuesta 4.1



Escribe un fichero HTML básico que conste de una cabecera y un cuerpo. Dentro del cuerpo, añade tres elementos de tipo párrafo, tres imágenes y dos enlaces. Luego, crea una hoja de estilo y utiliza el elemento @media para aplicar el siguiente estilo:

- Los párrafos deben tener una fuente en itálica, negrita y un tamaño de 18 px en cualquier dispositivo.
- Las imágenes deben aparecer centradas solo cuando se visualicen en una pantalla.
- Tanto los párrafos como la imagen deben estar alineados a la izquierda en todos los dispositivos, excepto cuando la imagen se muestre en una pantalla.
- Los enlaces siempre tienen que aparecer centrados.

4.3.2. @import

A través de @import es posible importar nuevas reglas desde otros ficheros CSS, e indicar el medio o medios sobre los que debe aplicarse el nuevo estilo. En este caso, a diferencia de lo que ocurría con @media, en primer lugar, aparece la URL que apunta al fichero CSS que contiene la guía de estilo, y, a continuación, se indica el medio sobre el que se va a aplicar. Es importante tener en cuenta que estas reglas deben preceder a todas las demás reglas.

Si los datos de estilo del fichero se van de aplicar a varios medios al mismo tiempo, se indicará el nombre de estos separados por comas.

- Sintaxis de @import:

```
@import url(nombreFichero.css) nombreTipoMedio;
```

- Ejemplo de código:

```
@import "ficheroEstilo1.css" print;  
@import "ficheroEstilo2.css" screen;  
@import "ficheroEstilo3.css" print, screen;  
@import "ficheroEstilo4.css"; (Por defecto se aplica a todo)
```

En el ejemplo anterior, las dos primeras reglas aplicarán los estilos contenidos en los ficheros “ficheroEstilo1.css” y “ficheroEstilo2.css”, en medio impreso y por pantalla, respectivamente. La tercera regla aplicará el estilo de “ficheroEstilo3.css” en ambos medios. Finalmente, la última regla, al no indicar ningún medio, se aplicará en todos (*all*).



PARA SABER MÁS

Es posible encontrar las reglas anteriores combinadas entre sí en un mismo fichero escrito en HTML. Por ejemplo, anidando las siguientes líneas:

- Utilizando *link*:

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

- Utilizando *@import*:

```
@import url("estilos_seccion.css") screen;
```

- Utilizando *@media*:

```
@media print {  
    /* Estilos específicos para impresora */  
}
```



Actividad propuesta 4.2

Crea un fichero HTML básico que incluya una cabecera y un cuerpo. En el cuerpo, añade al menos dos elementos de tipo párrafo y una imagen. Luego, crea una hoja de estilo externa en CSS y utiliza la regla *@import* para vincularla al archivo HTML. En la hoja de estilo, aplica los siguientes estilos:

- Los párrafos deben tener un tamaño de fuente de 16 px y un color de texto gris.
- La imagen debe tener un borde sólido de 2 px de color negro.
- Ajusta los márgenes y el espaciado para que los elementos estén distribuidos uniformemente en la página.

4.4. El modelo de cajas de CSS

El modelo de cajas es una de las características más importantes de las hojas de estilos. Cuando se crea una página en HTML, el uso de cada etiqueta simula lo que podría considerarse una caja, dotando a cada una de ella de un identificador, indicado por el nombre de la etiqueta. Este sistema de cajas permite, desde el fichero CSS, seleccionar cada caja y dotarle del estilo deseado.

El uso de las guías de estilo con CSS permite modificar todas las características de cada una de las cajas creadas. A continuación, se analizan cada una de estas partes a través de un ejemplo que irá incorporando nuevas propiedades (figura 4.3).

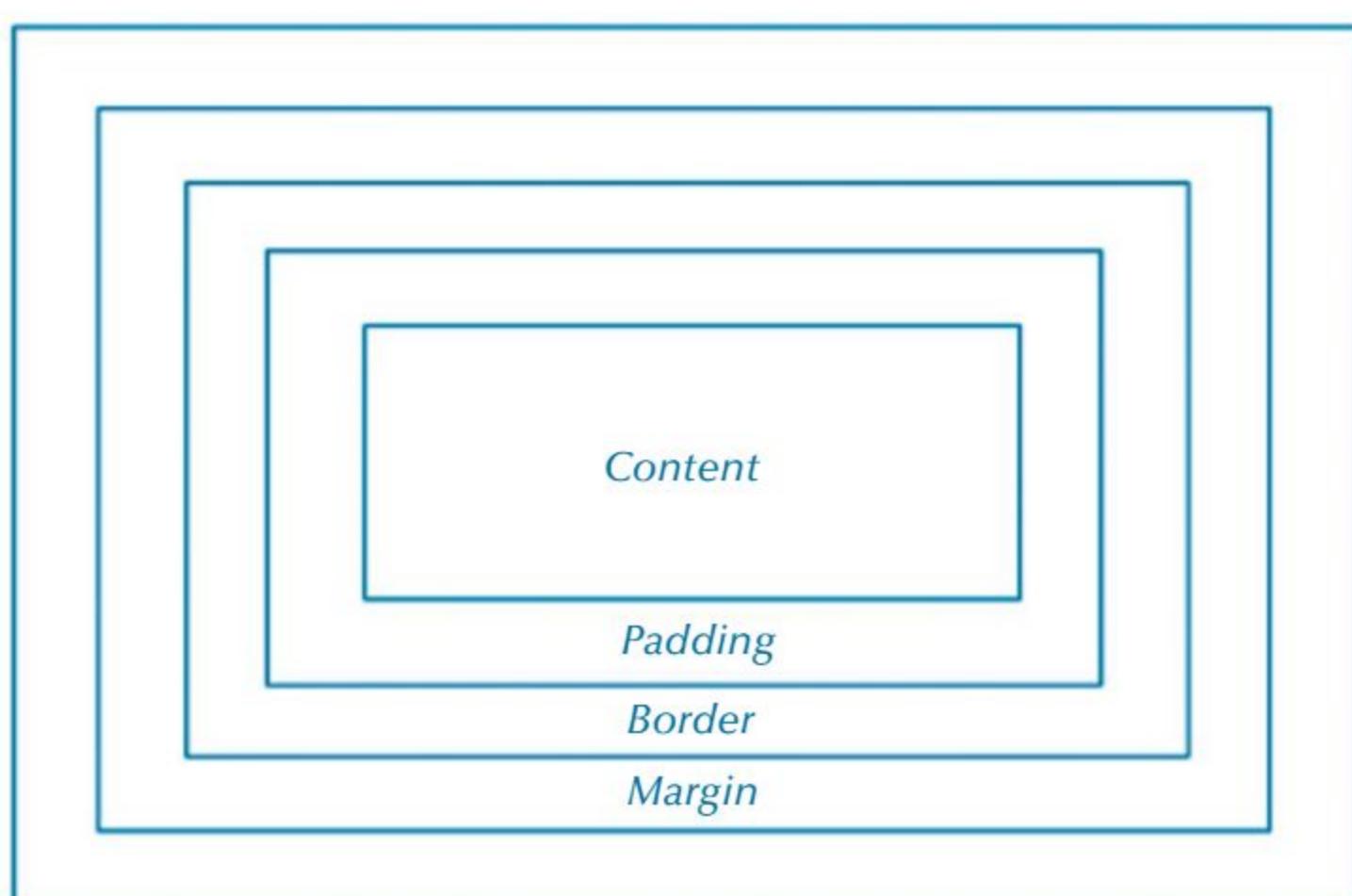


Figura 4.3
Diagrama modelo de cajas en CSS.

Las propiedades principales con las que se trabaja en el modelo de cajas son *width* y *height*, relativas a las dimensiones de la *caja de contenido* o *content*; se trata del el ancho y el alto del área donde se muestra el contenido de la caja.

Otra de las propiedades más importantes del área de contenido es el fondo de la caja. Se diferencia entre *background image* (imagen de fondo), que corresponde a la imagen que se muestra por detrás del contenido y el espacio de relleno, y *background color* (color de fondo), que es el color que se muestra por detrás del contenido y el espacio de relleno.

- Por ejemplo, si se introduce este código:

```
article {
    background: #5DEAAB;
    height: 100px;
    width: 200px;
}
```

- En pantalla, se verá como se observa en la figura 4.4.

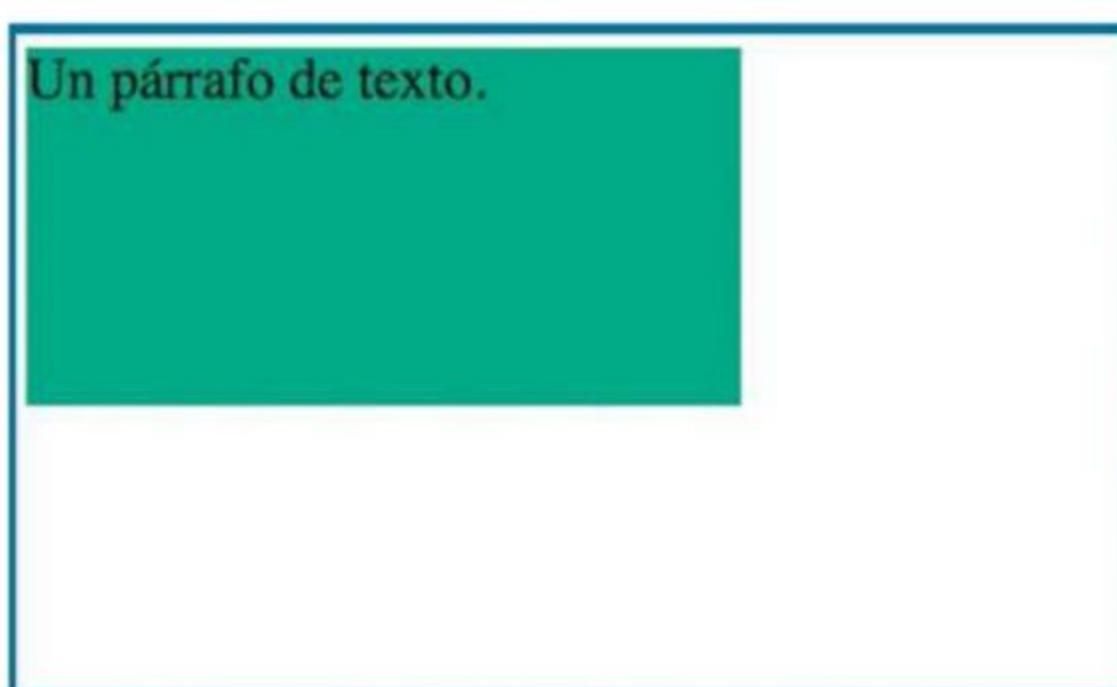


Figura 4.4
Color de fondo.

La segunda área es la de relleno, llamada *padding*. Es el espacio que existe entre la caja de contenido y el borde. Las características de esta caja que pueden modificarse son su altura y anchura. Se distinguen, por tanto, cuatro zonas posibles de acción: zona inferior, zona superior, zona izquierda y zona derecha. El nombre de las propiedades que definen estas zonas son *padding-top*, *padding-right*, *padding-left* y *padding-bottom*.

- Ejemplo:

```
article {  
    padding-top: 10px;  
    padding-right: 30px;  
    padding-bottom: 10px;  
    padding-left: 20px;  
}
```

Si solo escribimos *padding*, se aplicará el mismo formato a todos los extremos. En el siguiente ejemplo, los límites inferior, superior, izquierdo y derecho tendrían un valor de 10 px.

- Ejemplo:

```
article {  
    padding: 10px;  
}
```

Existen más tipos de empleo de esta regla que simplifican su uso, que se basan en el empleo de la palabra *padding* seguida de varias cifras separadas sin comas. Si aparecen dos valores, el primero es asignado al límite superior y al inferior, mientras que la segunda cifra será la que se aplique al relleno horizontal. En el caso de tener tres valores, el primero y el último serán los aplicados a los límites superior e inferior, respectivamente, y la segunda cifra indicará el valor de los bordes izquierdo y derecho.



PARA SABER MÁS

Existen otras propiedades que permiten el manejo refinado del tamaño de la caja de contenido para limitar los tamaños. Para ello se usan las propiedades *min-width*, *max-width*, *min-height*, y *max-height*.



Actividad propuesta 4.3

Realiza un fragmento de código en el que se pongan en uso todos los métodos de *padding* vistos en este apartado.

La siguiente área que puede verse en el diagrama de cajas es la correspondiente al borde, llamado *border*. Esta zona es la que encierra el contenido y el relleno, *content* y *padding*. Es posible modificar su grosor, estilo y color; además, al igual que ocurre con *padding*, es posible dar formato a los cuatro extremos de la caja a la vez utilizando solo la palabra *border* o hacerlo de forma individual.

- Ejemplo:

```
article {
    border: 1px solid #000000;
}
```

En aquellas ocasiones en las que se desee dar un formato diferente a cada uno de los extremos, se utilizarán algunas de las propiedades individuales destinadas a tal fin (cuadro 4.3).

CUADRO 4.3 Propiedades individuales

<i>border-top</i> <i>border-right</i> <i>border-bottom</i> <i>border-left</i>	Modifican a la vez el grosor, el estilo y el color de cada lado del borde.
<i>border-width</i> <i>border-style</i> <i>border-color</i>	Modifican de forma individual el grosor, el estilo y el color de los cuatro extremos a la vez.
<i>border-top-width</i>	Modifica el grosor del borde superior.
<i>border-top-style</i>	Modifica el estilo del borde superior.
<i>border-top-color</i>	Modifica el color del borde superior

Finalmente, el margen que envuelve al resto de elementos CSS es el denominado *margin*, y sostiene a otras cajas del diseño. Las propiedades individuales son *margin-top*, *margin-right*, *margin-bottom* y *margin-left*. El funcionamiento es el mismo que el de la propiedad *padding*.

- Ejemplo:

```
article {
    margin-top: 10px;
    margin-right: 10px;
    margin-bottom: 10px;
    margin-left: 10px;
}
```

TOMA NOTA



Los márgenes tienen un comportamiento peculiar llamado *margin collapsing*. Cuando dos cajas se tocan, la distancia entre ellas es el valor del margen más grande, y no la suma de ambos.

4.5. Propiedades CSS heredables

La herencia en CSS es el mecanismo mediante el cual determinadas propiedades de un elemento “padre” se transmiten a sus “hijos”. No todas las propiedades CSS son heredables; los márgenes no se heredan porque es poco probable que un elemento “hijo” necesite los mismos márgenes que su “padre”. Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad.

- Ejemplo:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
    <title>Ejemplo de herencia de estilos</title>
    <style type="text/css">
      body {color: blue; }
    </style>
  </head>

  <body>
    <h1>Titular de la página</h1>
    <p>Un párrafo de texto no muy largo.</p>
  </body>
</html>
```

En el ejemplo, el selector *body* solo establece el color de la letra para el elemento *<body>*. Esta propiedad se hereda de forma automática, ¿qué significa esto? Que los elementos descendientes muestran ese mismo color. Ahora bien, la herencia de estilos se aplica automáticamente, pero su efecto puede ser anulado estableciendo de forma explícita otro valor para la propiedad que se hereda.

En cambio, existen otras propiedades cuyo valor no se hereda, como es el caso de los bordes (*border*); es decir, si se define un estilo relativo al borde para el cuerpo, este no será aplicado al resto de elementos HTML. En el cuadro 4.4 se resumen algunas de las propiedades que sí son heredadas por defecto en los elementos HTML.

CUADRO 4.4**Resumen de las propiedades heredadas por defecto en HTML**

color	text-align
font	visibility
letter-spacing	white-space
direction	border-collapse
word-spacing	quotes
line-height	list-style-image

Ahora bien, como ya se ha dicho, existen ciertas propiedades que no son heredables “de padres a hijos”, pero existe un mecanismo mediante el cual es posible indicar que un elemento herede el valor de una propiedad. Para ello se utilizan los valores: *inherit*, *initial* y *unset*, que modifican el valor de la propiedad respecto a la herencia.

- a) *inherit*. Aplica el valor de la propiedad del elemento “padre”.
 - b) *initial*. Aplica el valor que presenta inicialmente la propiedad.
 - c) *unset*. Hereda el valor de la propiedad del elemento “padre”; si este no existe, aplica el valor inicial.
- Ejemplo:

```
body {
    border-width: 10px;
    border-color: blue;
    color:red;
}

h1 {
    border:inherit;
}
```

En este ejemplo, el elemento *h1* hereda todas las propiedades establecidas en el *body*. El color se hace forma automática, puesto es una de las propiedades que se heredan, y al no establecerse otro valor para el color en *h1* quedará de color rojo. Ahora bien, en este ejemplo las propiedades de borde también serán heredadas puesto que así es indicado por el valor *inherit*.

Actividad propuesta 4.4

Reescribe el código anterior para que el elemento ‘h1’ sea de color verde. El resto de los elementos heredarán su valor de *body*.

4.6. CSS3

CSS3 (*Cascading Style Sheets 3*) es la tercera versión del lenguaje de hojas de estilo CSS. En este apartado se verán algunas de las novedades que presenta frente a su versión anterior, CSS2.1. Esta nueva actualización presenta todas las características de su predecesora y, además, aporta un mayor dinamismo en el diseño de la página web. En cuanto a los aspectos relativos a la programación, esta no varía; se sigue empleando la estructura de propiedades y selectores vista a lo largo del capítulo.

A) Múltiples imágenes en el fondo

En esta nueva versión es posible poner varias imágenes de fondo en un mismo elemento. En el siguiente ejemplo, se puede ver cómo se utilizan varias imágenes para el fondo y, a continuación, de la URL de cada una de ellas indica las posiciones del *background*.

- Ejemplo:

```
#contenidoPrincipal {  
    width: 500px; height: 500px;  
    background:  
        url(imagen1.gif) bottom right no-repeat,  
        url(imagen2.jpg) center center no-repeat,  
        url(imagen3.jpg) repeat;  
}
```

B) Nuevas propiedades para el fondo de los elementos

En CSS3 se incorporan nuevas propiedades que definen las características del fondo de los elementos. Estas son *background-origin*, *background-clip* y *background-size*.

- *Background-origin* especifica la posición de origen de un fondo o imagen.
- *Background-clip* permite definir el comportamiento de un fondo o imagen de fondo, con respecto al borde, si debe mostrarse por debajo de este o no. Esta propiedad se aplica a todas las cajas contenedoras conocidas: *border-box*, *padding-box* y *content-box*.
- *Background-size* permite definir el tamaño de las imágenes de fondo.

C) Esquinas redondeadas

Esta nueva propiedad permite redondear las esquinas de las áreas de la página, en concreto, a través del atributo ‘border-radius’. Cabe tener en cuenta que, en función del navegador, el nombre del atributo varía: para Chrome y Safari se utiliza: ‘-webkit-border-radius’, y para Mozilla: ‘-moz-border-radius’.

- Ejemplo:

```
div {
    border: 2px solid #ffffff;
    -moz-border-radius: 9px;
    -webkit-border-radius: 9px;
}
```

D) Sombras

Esta versión incorpora una propiedad para crear sombras en los elementos, incluso en los textos. El atributo que se utiliza es ‘box-shadow’ y está formado por cuatro valores: desplazamiento horizontal de la sombra, desplazamiento vertical de la sombra, difuminado y color de la sombra; estos valores aparecen siempre en el mismo orden.

- Ejemplo:

```
div {
    box-shadow: 5px -9px 3px #000000;
}
```

E) Transparencias de color

Otra de las nuevas incorporaciones en CSS3 es la posibilidad de añadir transparencia en el color; desde un color totalmente sólido hasta el transparente. La propiedad que define la transparencia se denomina ‘opacity’, y toma un valor decimal entre 0.0 y 1.0, donde 0.0 es la máxima transparencia y 1.0 la máxima opacidad.

- Ejemplo:

```
div {
    opacity: 0.5;
}
```

F) Texto en varias columnas

En CSS3 se incorpora la opción de maquetar el texto en varias columnas a través de varios atributos: ‘colum-width’ (define el ancho de las columnas), ‘column-gap’ (define el espacio en blanco entre columnas) y ‘column-rule’ (crea una línea entre las columnas). Aunque en la actualidad aún no son soportadas por ningún navegador, de manera experimental sí son implementadas por Safari, Google Chrome y Firefox. Al igual que en el caso de ‘border-radius’, se utiliza un prefijo para su uso en función del navegador en el que se usan: ‘-moz-’ en Firefox y ‘-webkit-’ en Safari y Chrome.

- Ejemplo:

```
div{  
    -moz-column-width: 20em;  
    -moz-column-gap: 10px;  
    -webkit-column-width: 20em;  
    -webkit-column-gap: 10px;  
    -webkit-column-rule: 2px solid #ffffff;  
    -moz-column-rule: 2px solid #ffffff;  
}
```

G) Animaciones

Unas de las novedades más importantes es la creación de animaciones. Estas se crean directamente desde CSS3 sin la necesidad de utilizar otros lenguajes; además, con este tipo de implementación ya no será necesario tener instaladores *plugins*, que antes eran inevitables para la correcta visualización de las animaciones.



Actividad propuesta 4.5

Diseña y escribe un programa en HTML y CSS3 que tenga tres columnas del mismo tamaño. Además, el color de fondo de la página debe ser azul con una transparencia del 30%.

4.7. Bootstrap 4. Diseño ágil y atractivo

Bootstrap, creado por Twitter, constituye uno de los *frameworks* CSS más populares en la actualidad; esto es, plantillas CSS con gran capacidad de adaptación a distintos tipos de plataforma web mediante el uso de librerías CSS, que incluyen un amplio repositorio de tipográficas, botones y cuadros, entre otros. La versión actual es Bootstrap 4.x.

Recursos web

www

Bootstrap está en código abierto y se encuentra disponible en el primer enlace. Puedes acceder a las plantillas a través del segundo enlace.

1. <http://getbootstrap.com/getting-started/>
2. <https://themes.getbootstrap.com/>

Esta herramienta permite crear un ilimitado conjunto de interfaces combinando todos los recursos que este *framework* pone al alcance de los usuarios. Bootstrap es compatible con la

mayor parte de los navegadores más extendidos hoy en día: Google Chrome, Safari, Mozilla Firefox, Internet Explorer y Opera.

Existen distintas plantillas sobre las que se pueden realizar los desarrollos y también se puede comenzar desde cero combinando todos los elementos que Bootstrap pone a disposición. Junto al código disponible, pueden encontrarse diversas plantillas como las mostradas en la figura 4.5. Esta herramienta facilita el uso de este código, puesto que basta con tomar los elementos que se desean e incorporarlos después en la página web.

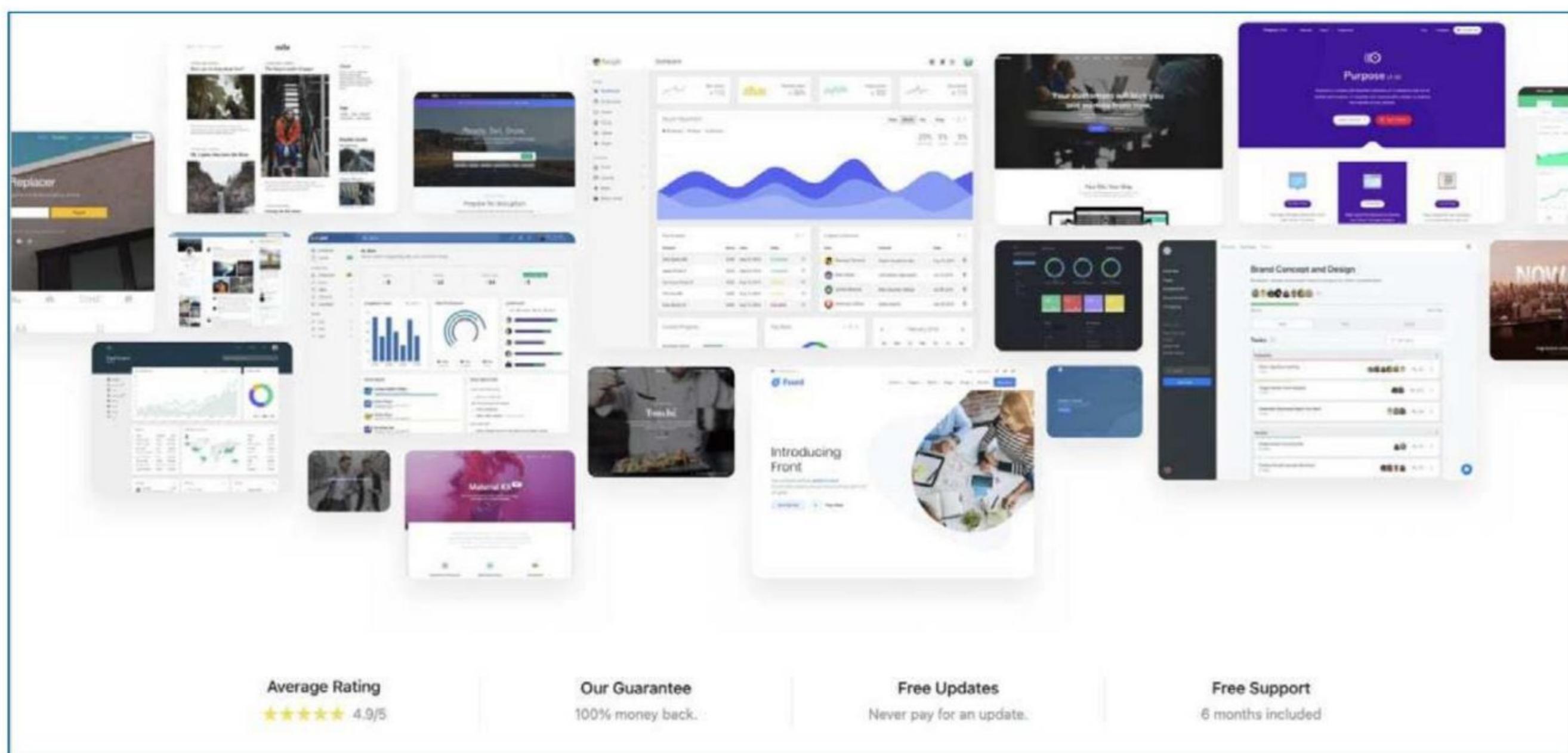


Figura 4.5
Sitio web Startbootstrap, con plantillas Bootstrap 4.

Por lo tanto, se puede resumir que Bootstrap es una hoja de estilo CSS, como las vistas hasta ahora, que puede aplicarse al documento de HTML. Para enlazar el código de un sitio web con el archivo contenedor de las reglas de estilo, 'bootstrap.min.css', se hace la misma forma que en el resto de los casos.

- Ejemplo:

```
<html>
  <head>
    <title>Prueba de Bootstrap</title>
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <div class="container" style="background-color: #cccccc">
      <p>Esto es una prueba de bootstrap 4.</p>
    </div>
  </body>
</html>
```

Utilizar Bootstrap en el diseño de un sitio web es muy sencillo. En primer lugar, desde su portal web se selecciona aquella plantilla que más adecúe el sitio que se está diseñando. Existen

multitud de opciones de plantillas gratuitas; una vez que se ha encontrado la apropiada, se descarga en el equipo.

A continuación, se descarga el código CSS y JavaScript compilado del código fuente de la plantilla, que contiene todos los archivos Less, y JavaScript originales de Bootstrap. Tras descargar y descomprimir la estructura de ficheros es la siguiente:

```
bootstrap/
└── css/
    ├── bootstrap.css
    ├── bootstrap.min.css
    ├── bootstrap-theme.css
    └── bootstrap-theme.min.css
└── js/
    ├── bootstrap.js
    └── bootstrap.min.js
└── fonts/
    ├──glyphicon-halflings-regular.eot
    ├──glyphicon-halflings-regular.svg
    ├──glyphicon-halflings-regular.ttf
    └──glyphicon-halflings-regular.woff
```

Normalmente, para cada archivo se incluyen dos opciones: los archivos compilados o los archivos compilados y comprimidos. En las carpetas se diferencian los archivos Bootstrap.css, los cuales añaden estilos a formularios, tablas o imágenes; y archivos Bootstrap.js, que incluyen el código JavaScript para la creación de ventanas emergentes, alertas, efectos de carrusel, etc.

Tras enlazar el código del documento en HTML a la hoja de estilo CSS, como se ha explicado al inicio de este apartado, utilizando las etiquetas de cada uno de los elementos contenidos en la plantilla Bootstrap, será posible dar el formato deseado al sitio web, bien utilizando todos los elementos de la guía de estilo sin modificaciones, bien realizando una versión propia.



Actividad propuesta 4.6

¿Cuáles son las características más importantes de Bootstrap 4? ¿Qué versión se utiliza en la actualidad? ¿Se usa solo para el diseño de CSS, o también incluye funciones para añadir interactividad? ¿Qué lenguaje de programación se emplea para esto último? ¿Por qué crees que cada vez está más presente su aplicación en el diseño de interfaces web?

4.7.1. Container

Bootstrap proporciona una serie de herramientas y componentes esenciales para el desarrollo de sitios y aplicaciones web. Los fundamentos de Bootstrap son la base sobre la cual puedes construir tu diseño y funcionalidad.

Para controlar el diseño y la estructura de una página web, Bootstrap ofrece tres tipos principales de containers: `.container`, `.container-fluid`, y `.container-sm`, diseñados para adaptarse a diferentes escenarios de diseño y tamaños de pantalla.

El elemento `container` es el tipo de contenedor más común en Bootstrap. Está ideado para crear un diseño fijo y centrado en una página web. Se adapta a tamaños de pantalla más grandes, y se utiliza cuando se desea que el contenido se mantenga en un ancho máximo predefinido.

A continuación, se muestra un ejemplo sencillo, en el que se crea un elemento `<div>` bajo la clase `container-fluid` y, dentro de esta, un nuevo elemento `<div>` de tipo de `container`, dentro del que pondremos el texto.

```
<div class="container-fluid bg-primary">
  <div class="container">
    <p> Lorem Ipsum ..... </p>
  </div>
</div>
```

Como se observa en la salida de la imagen (figura 4.6), al recoger bajo la clase `container` el elemento `<p>` el texto presentará los márgenes definidos de este tipo de componentes, mientras que `container-fluid` ocupará todo el navegador.

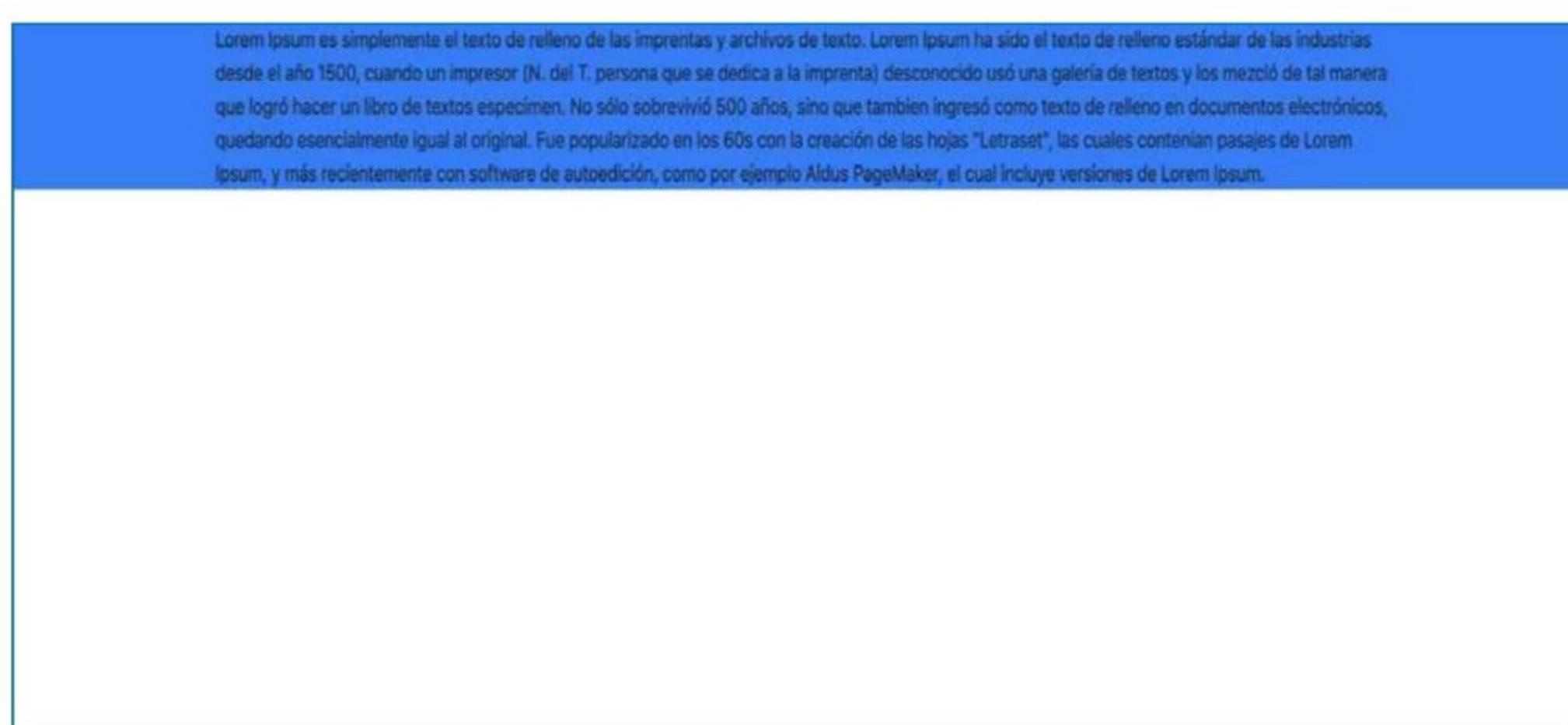


Figura 4.6
Página resultante código 7. Bootstrap 4.

4.7.2. Grid. Sistema de columnas en Bootstrap

El sistema de columnas de Bootstrap se basa en la división del contenido en 12 columnas, pero esto no implica que la creación de columnas sea fija, ni que solo se puedan crear 12. Este sistema nos permite crear diferentes agrupaciones, utilizando como referencia las 12 porciones iniciales.

En primer lugar, será necesario crear un elemento `fila class="row"` y, a continuación, con base en el sistema de doce columnas, se indicarán las columnas y el número de porciones sobre 12 que ocupa cada una de ellas.

Para crear las columnas, se utilizará el elemento `col`, que permite especificar el tamaño de las columnas creadas:

- `col`: crea columnas de igual tamaño, distribuye el espacio total entre el número de elementos `col` creados.
- `col-x`: donde *x* indica el tamaño de cada columna, por lo que se puede establecer el tamaño de cada columna con respecto al sistema de doce columnas de la cuadrícula. Es posible combinar los dos tipos de elementos; por ejemplo, una columna `col-6`, y dos columnas `col`, por lo que estas últimas tendrán un tamaño de 3, ya que reparten el espacio restante sobre el sistema de 12 columnas, que deja libre la primera.
- `col-x-auto`: se utiliza para crear columnas que se ajustan automáticamente al contenido que tienen, sin un ancho fijo. Esto resulta útil cuando se necesita que una columna se expanda según su contenido, en lugar de ocupar un ancho específico predefinido.

Por ejemplo, si se quieren formar tres columnas, sería necesario crear tres bloques de cuatro (`col-4`). En el caso de que fueran dos columnas, se podrían crear dos bloques de seis (`col-6`). Se muestra a continuación, en la figura 4.7, la salida correspondiente a este ejemplo.



Figura 4.7
Página resultante código 8. Bootstrap 4.

El código utilizado para su implementación se muestra a continuación. Para usarlo, basta con copiarlo dentro del `<body>` de un fichero HTML.

```
<div class="container-fluid">
  <div class="container">
    <div class="row">
      <div class="col-4 bg-primary">
        <p> Lorem Ipsum ..... </p>
      </div>
      <div class="col-4 bg-success">
        <p> Lorem Ipsum ..... </p>
      </div>
      <div class="col-4 bg-primary">
        <p> Lorem Ipsum..... </p>
      </div>
    </div>
  </div>
</div>
```

```
<div class="row">
    <div class="col-6 bg-primary">
        <p> Lorem Ipsum Lorem Ipsum ..... </p>
    </div>
    <div class="col-6 bg-sucess">
        <p> Lorem Ipsum Lorem Ipsum ..... </p>
    </div>
</div>
</div>
```

En Bootstrap, se incorporan diferentes clases para definir columnas en el sistema de rejilla (*grid*). Siguen una convención específica, y se basan en el número de columnas que se desea que una determinada sección o elemento ocupe en una fila. A continuación, se muestran algunas de las clases más comunes que se usan para definir columnas en Bootstrap:

- .col- (para dispositivos extra pequeños).
- .col-sm- (para dispositivos pequeños).
- .col-md- (para dispositivos medianos).
- .col-lg- (para dispositivos grandes).

La clase `row-cols` se utiliza en combinación con `cols` para especificar el número de columnas que se tienen que mostrar en cada fila. Permite agilizar el desarrollo, puesto que se incluyen todas las columnas de contenido, y estas son distribuidas en filas, atendiendo a las columnas indicadas en `row-cols`.

Esto se debe incluir junto al elemento de fila `row`, de la siguiente manera:

```
<div class="row row-cols-2">
    <div class="col"> Uno </div>
    <div class="col"> Dos </div>
    <div class="col"> Tres </div>
    <div class="col"> Cuatro </div>
</div>
```

En el ejemplo anterior, se crearán dos columnas en cada fila, por lo que habrá un total de dos filas para mostrar las cuatro columnas creadas (Uno, Dos, Tres, Cuatro).

En Bootstrap, se incorporan nuevas clases, que permiten controlar el orden de las columnas en el sistema de cuadrícula:

- `order-first`: esta clase se utiliza para colocar una columna en la posición inicial de la cuadrícula. Por ejemplo, se podría colocar en una columna concreta para que aparezca antes que otras incluso si no está en esa posición en el HTML.
- `order-last`: esta clase se usa para situar una columna en la última posición de la cuadrícula. Al igual que `order-first`, se puede aplicar esta clase a una columna que aparezca al final, independientemente de su posición en el HTML.
- `order-X`: donde *x* es un número del 1 al 12. Esta clase permite establecer un orden específico para cada columna. Por ejemplo, la columna con clase `.order-1` se colocará antes que las columnas con clases `.order-2` y `.order-3`.

4.7.3. Tipografías

Bootstrap incorpora reglas relativas a la tipografía. De esta forma se pueden aplicar formatos profesionales y uniformes a un nuestro sitio web, de una forma rápida y efectiva.

Los tipos de cabeceras h (h1, h2, h3...) presentan un tipo predefinido; por ejemplo, h1 se define como `semibold 2.5rem (40px)`. La clase `display` es muy útil en estos casos, puesto que aporta un diseño más sofisticado al sitio; se utiliza en combinación con las cabeceras descritas.

De la misma forma, para los elementos de tipo párrafo se utiliza la clase `lead`, que aporta un diseño más estilizado a la tipografía (figura 4.8).

Otro elemento de Bootstrap sobre tipografía es `text-muted`, mediante el cual es posible enfatizar algunas palabras del texto, que aparecerán de otro color. Además, con la etiqueta `small` se reduce el tamaño del texto marcado a un 80%.

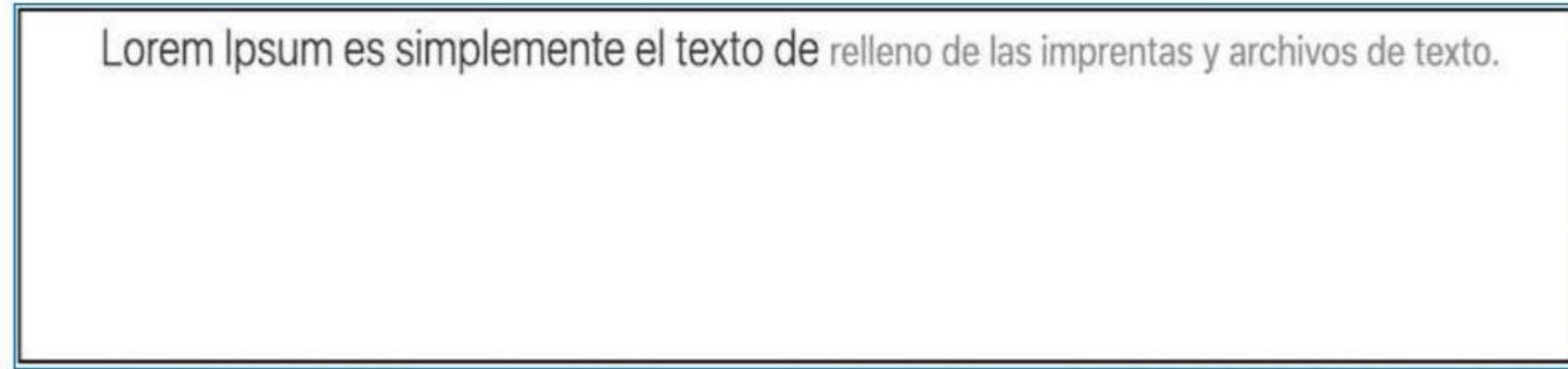


Figura 4.8
Ejemplo de texto con formato aplicado código 9. Bootstrap 4.

```
<div class="container">
  <p class="lead">
    Lorem Ipsum es simplemente el texto de
    <small class="text-muted">relleno de las
    imprentas y archivos de texto.
  </p>
</div>
```

Para alinear los bloques de texto, basta con utilizar la siguiente sentencia, indicando la alineación deseada:

`class="text-(justify|center|left|right)"`

- Alineamiento vertical (`align-items-x`): permite controlar el alineamiento vertical de los elementos dentro de un contenedor. Algunos de los valores que puede tomar son: `align-items-start` (alinea los elementos en la parte superior del contenedor), `align-items-end` (alinea los elementos en la parte inferior del contenedor), `align-items-center` (en el centro vertical del contenedor) y `align-items-baseline` (alinea los elementos de manera que las líneas de base de texto estén alineadas).
- Alineamiento horizontal (`justify-content-x`): permite controlar el alineamiento horizontal de los elementos dentro de un contenedor. Puede tomar varios valores: `justify-content-start`, `justify-content-end`, `justify-content-center`, `justify-content-around` o `justify-content-between`.

También se incorporan utilidades que permiten cambiar de forma automática las letras a mayúsculas, a minúsculas, o actuar sobre las iniciales.

```
class="text-(lowercase|uppercase|capitalize)"
```

4.7.4. Colores

Como se ha visto en algunos ejemplos de código a lo largo de este apartado, en Bootstrap la dotación de color responde a un sistema concreto de etiquetas (figura 4.9). Es posible utilizarlo con todos aquellos elementos que permiten modificar su color: *background* (`bg`), texto (`text`), etc.



Figura 4.9
Código de colores en Bootstrap 4.

4.8. YUI

El *framework* YUI (Yahoo! User Interface) es un conjunto de utilidades y buenas prácticas en CSS y JavaScript que permite la construcción de sitios web con una apariencia más dinámica.

Es frecuente que, para la descarga de estos *frameworks*, se utilice la vinculación directa a través del fragmento de código siguiente. También es posible hacerlo mediante la descarga de la librería en el equipo.

Desde el sitio web <https://yuilibrary.com>, selecciona la pestaña “Quick Start” y, a continuación, “Download YUI”. Tras su descarga, podrás descomprimir el contenido en una ubicación del ordenador conocida, puesto que luego se indicará sobre el fichero HTML para tomar los criterios de estilo.

El directorio más importante es “build”, ya que es aquí donde se colocan todos los ficheros JavaScript y CSS que se van a incluir en las páginas que utilizan el framework YUI.

Gracias a la arquitectura modular de YUI, es posible enlazar solo aquellos archivos que sean necesarios, produciendo una mejora en el rendimiento de la aplicación y evitando una sobrecarga innecesaria en el sistema. Como se analizó en capítulos anteriores, para enlazar un fichero HTML con una página de estilo en CSS se utiliza la etiqueta `<link>`.

Por ejemplo, en el siguiente fragmento de código queda enlazada con el fichero de estilo `font.css`. En el atributo `href` se ha de indicar la ruta exacta en la que se encuentra el fichero CSS.

```
<link rel="stylesheet" type="text/css" href="css/reset-fonts-grids.css">
```

El funcionamiento esencial de YUI se basa en la aplicación de los siguientes ficheros:

- `reset.css`: este fichero se encarga de neutralizar los efectos de formato por defecto que pudiera tener cualquier navegador, eliminando de esta manera formatos residuales. A continuación, siempre hay que aplicarles el nuevo formato; de lo contrario, el resultado no será adecuado.
- `base.css`: tras la aplicación del reseteo anterior, se aconseja utilizar este fichero de estilo, puesto que aporta un formato estandarizado al sitio web, consiguiendo una interfaz ya visible por parte del usuario.
- `fonts.css`: este fichero aporta ciertas pautas de estilo generales para los bloques principales en HTML.
- `grids.css`: en este último fichero se incorporan diferentes selectores, en los que YUI decide la estructura de la página (página, plantilla y rejilla).

Para practicar lo visto hasta ahora, tras la descarga de la librería copia los ficheros de formato anteriores y déjalos en una carpeta llamada “css”. A continuación, crea un fichero en HTML con el siguiente contenido, y observa lo que ocurre al ir añadiendo, uno a uno, los ficheros de estilo anteriores. En primer lugar, incorpora solo `reset.css` y observa lo que ocurre; después, añade `base.css`, y, finalmente, `fonts.css`.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Viewport Sample</title>
    <link rel="stylesheet" type="text/css" href="css/reset.css">
    <link rel="stylesheet" type="text/css" href="css/base.css">
    <link rel="stylesheet" type="text/css" href="css/fonts.css">
</head>
<body>
    <div class="box">hola</div>
    <h1> h1 </h1>
    <h2> h2 </h2>
</body>
</html>
```

Este *framework* divide internamente la estructura de cada página en tres partes, que tienen asociados sus propios selectores. En función de los que se asignen en cada bloque etiquetado en HTML, tendremos un resultado u otro (cuadro 4.5).

CUADRO 4.5
Tipos de estructuras en YUI y clases asociadas a cada una

Página	Plantilla	Rejilla
#doc, #doc2, #doc3 y #doc4	.yui-b, .yui-t1, .yui-t2, .yui-t3, .yui-t4, .yui-t5 y .yui-t6	.yui-u, .yui-g, .yui-gb, .yui-gc, .yui-gd, .yui-ge, .yui-gf
Determina el ancho total de la página	Determina el número de columnas en una página	Determina las divisiones de las columnas

Por ejemplo, para determinar la anchura total de la página se tomará el selector de página (#doc) que se adecúe a las dimensiones de diseño. De forma general, #doc, #doc2 y #doc4 colocan de forma centrada el contenido, mientras que #doc3 lo ajusta a la anchura del navegador.

Los selectores de plantilla definen el número de columnas. En este caso, debemos tener en cuenta el uso del selector .yui-b para concretar el número de columnas deseado, añadiendo sobre la principal el identificador yui-main. Para establecer el formato de columnas, es decir, el ancho en píxeles de cada una, se define sobre la etiqueta inicial de doc (<div id="doc" class="yui-t2">) el selector de plantilla adecuado (.yui-t1, .yui-t2, .yui-t3, .yui-t4, .yui-t5 y .yui-t6).

Por último, los indicadores de rejilla determinan el número de bloques en los que se divide cada una de las secciones anteriores (cuadro 4.6).

CUADRO 4.6

Clases de rejillas y descripción de tamaño

Rejilla	Descripción
.yui.g	2 unidades a partes iguales
.yui.gb	3 unidades a partes iguales
.yui.gc	2 unidades a 2/3, 1/3
.yui.gd	2 unidades a 1/3, 2/3
.yui.ge	2 unidades a 3/4, 1/4
.yui.gf	2 unidades a 3/4, 1/4

Para ver la hoja de estilo en la que se recogen todos los criterios de diseño, basta con consultar el fichero “grids.css”.

WWW

Recurso web



Las constantes innovaciones tecnológicas que se producen tanto en *software* como en *hardware* tienen un impacto directo en el cumplimiento de las metas propuestas por la ONU en la Agenda 2030, a través de los Objetivos de Desarrollo Sostenible. Puedes leer más sobre ello en la web de la Red Española por el Pacto Mundial, escaneando el siguiente código QR:



Resumen

- Cuando se trabaja con páginas web en HTML y XHTML es deseable que presenten un estilo concreto y un aspecto definido. Para lograr este fin aparece CSS (*Cascading Style Sheets*), un lenguaje de hojas de estilo que será el encargado de controlar el aspecto de los sitios web.
- Para añadir estilos a una página web basta con añadirlos en el código de la propia página web, en el sitio indicado para ello en la cabecera, y utilizando las etiquetas `<style>`.
- Normalmente, se van a utilizar hojas de estilo externas. En este caso, el código se encuentra en otro archivo. Desde la página en HTML se accede al fichero en código CSS a través de la etiqueta `<link>`.
- En la actualidad, el uso de CSS en HTML está muy extendido, puesto a través de un solo archivo CSS es posible incluir una misma guía de estilo en todas las páginas HTML de un mismo sitio web; de esta forma se cumple uno de los principios más importantes a la hora de realizar un correcto diseño gráfico: que exista uniformidad de estilo entre todas las páginas que componen el sitio.
- Las hojas de estilo permiten especificar la manera en que se representa un documento en función del tipo de medio de visualización.
- Se denominan *medios* a los canales a través de los cuales se muestra la información del sitio web: auditivos, visuales, impresos, etc.
- Existen dos reglas de selección del tipo de medio: `@media` especifica los tipos de medios de destino; a través de esta opción es posible que en la misma hoja de estilo aparezcan reglas para distintos medios. E `@import`, que permite importar nuevas reglas desde otros ficheros CSS, e indicar el medio o medios sobre los que debe aplicarse el nuevo estilo.
- El modelo de cajas es una de las características más importantes de las hojas de estilos. Cuando se crea una página en HTML, el uso de cada etiqueta simula lo que podría considerarse una “caja”, dotando a cada una de ellas de un identificador, indicado por el nombre de la etiqueta.
- Las propiedades principales con las que se trabaja son ‘width’ y ‘height’, relativas a las dimensiones de la *caja de contenido*. Es el área donde se muestra el contenido de la caja. Otra de las propiedades del área de contenido es el fondo de la caja. Se diferencia entre ‘background image’ (imagen de fondo), la imagen que se muestra detrás del contenido; y ‘background color’ (color de fondo), color que se muestra detrás del contenido y el espacio de relleno.
- Existen otras tres áreas sobre las que es posible definir su estilo: `padding`, es el espacio que existe entre el la caja de contenido y el borde; `border`, es la zona que encierra el contenido y el relleno; `content` y `padding` y `margin`.
- La herencia en CSS es el mecanismo mediante el cual ciertas propiedades de un elemento “padre” se transmiten a sus “hijos”. Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad. Hay ciertas propiedades que no son heredables “de padres a hijos”, pero existe un mecanismo mediante el cual es posible indicar que un elemento herede el valor de una propiedad. Para ello se utilizan los valores: `inherit`, `initial` y `unset`, que modifican el valor de la propiedad con respecto a la herencia.

- CSS3 (*Cascading Style Sheets 3*) es la tercera versión del lenguaje de hojas de estilo CSS. Esta nueva actualización presenta todas las características de su predecesora y, además, aporta un mayor dinamismo en el diseño de la página web.
- En cuanto a los aspectos relativos a la programación, con CSS3 no varía; se sigue empleando la estructura de propiedades y selectores. Algunas de las novedades son la posibilidad de utilizar múltiples imágenes de fondo, esquinas redondeadas, transparencias, etc.



Ejercicios propuestos

1. Detecta y corrige los errores que aparecen en el siguiente código para que funcione correctamente. Determina la salida del código.

- El código HTML:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>
      Ejercicio 1 del Capítulo 4. Estilos CSS en el documento
    </title>
    <link rel="stylesheet" type="text/css" media="screen">
  </head>
  <body>
    <h1>INICIO DEL EJERCICIO</h1>
    <div>
      <p>Un párrafo de texto.</p>
      <p>Un párrafo de texto.</p>
    </div>

    <ul>
      <li>Elemento 1</li>
      <li>Elemento 2</li>
    </ul>
  </body>
</html>
```

- El código CSS (hojaEst.css):

```
div {  
    color: red; font: Verdana;  
    opacity: 0.5;  
}  
body {  
    border-width: 10px;  
    border-color: blue;  
    color:red;  
}  
  
h1 {  
    border: inherit;  
}
```

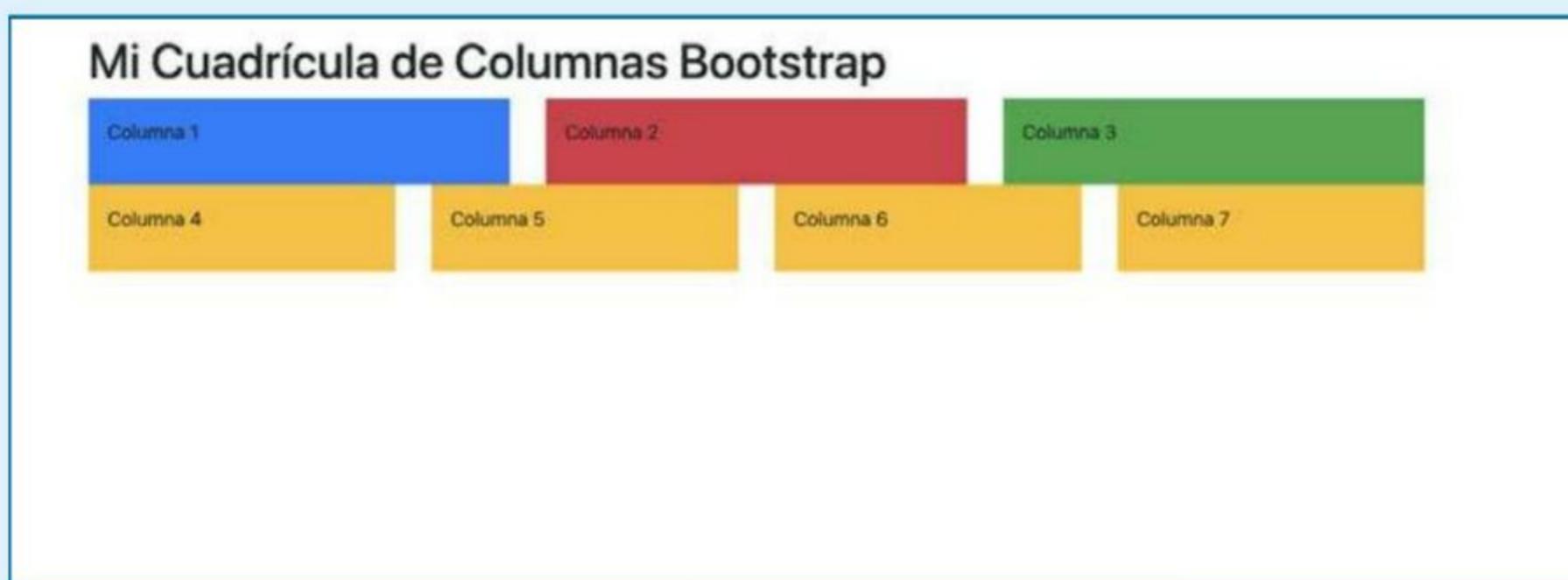
2. Un sitio web está compuesto por un conjunto de páginas: Menu.html, Noticias.html y FormularioContacto.html. Se desea que desde la página de "Menu.html" se pueda acceder directamente a las otras dos páginas, a través de dos botones. En cambio, desde Noticias.html y FormularioContacto.html solo será posible regresar a la página de inicio, Menu.html, a través de un botón. Escribe el código HTML necesario para crear este programa.
3. Añade, en el sitio web anterior, una nueva página "Fotos.html", a la cual se tendrá acceso desde "Menu.html" y "FormularioContacto.html", a través del texto: 'Galería de imágenes'. Esta nueva página va a contener, al menos, tres imágenes y un título para cada una de ellas. Las imágenes deben estar contenidas en una carpeta independiente llamada: "Fotos", que debe estar situada dentro de la carpeta central donde están los ficheros HTML. Crea la estructura de carpetas pedida, escribe el código de la página "Fotos.html" y reescribe los dos ficheros anteriores para cumplir con las nuevas especificaciones.
4. CSS3 presenta más dinamismo que su predecesora, CSS2.1. ¿En la implementación de qué nuevos elementos crees que se basa este diseño más dinámico e interactivo? ¿O es la combinación de todos ellos lo que proporciona esta característica?
5. Visita el sitio web de Bootstrap y descarga el sistema de ficheros principal. A continuación, escoge de entre todas las plantillas disponibles aquella que más se adapte al sitio web o sitios web que estás modelando a lo largo de los ejercicios de este libro. Descárgala y utiliza los elementos de la hoja de estilo para dar un nuevo formato al sitio web escogido.
6. Crea un sitio web utilizando el sistema de rejillas de Bootstrap para organizar el contenido de la siguiente manera:
 - Utiliza el sistema de rejillas de Bootstrap para dividir el contenido en cinco filas y tres columnas cada una.

- En cada una de las cinco filas, coloca contenido relevante utilizando las clases de Bootstrap para estructurarlo y diseñarlo.
- Asegúrate de que el diseño sea responsive y se adapte correctamente a diferentes tamaños de pantalla, utilizando las clases de Bootstrap para dispositivos móviles y de escritorio.
- Usa clases de Bootstrap para estilizar el contenido, como colores de fondo, espacio, alineación de texto, etc.

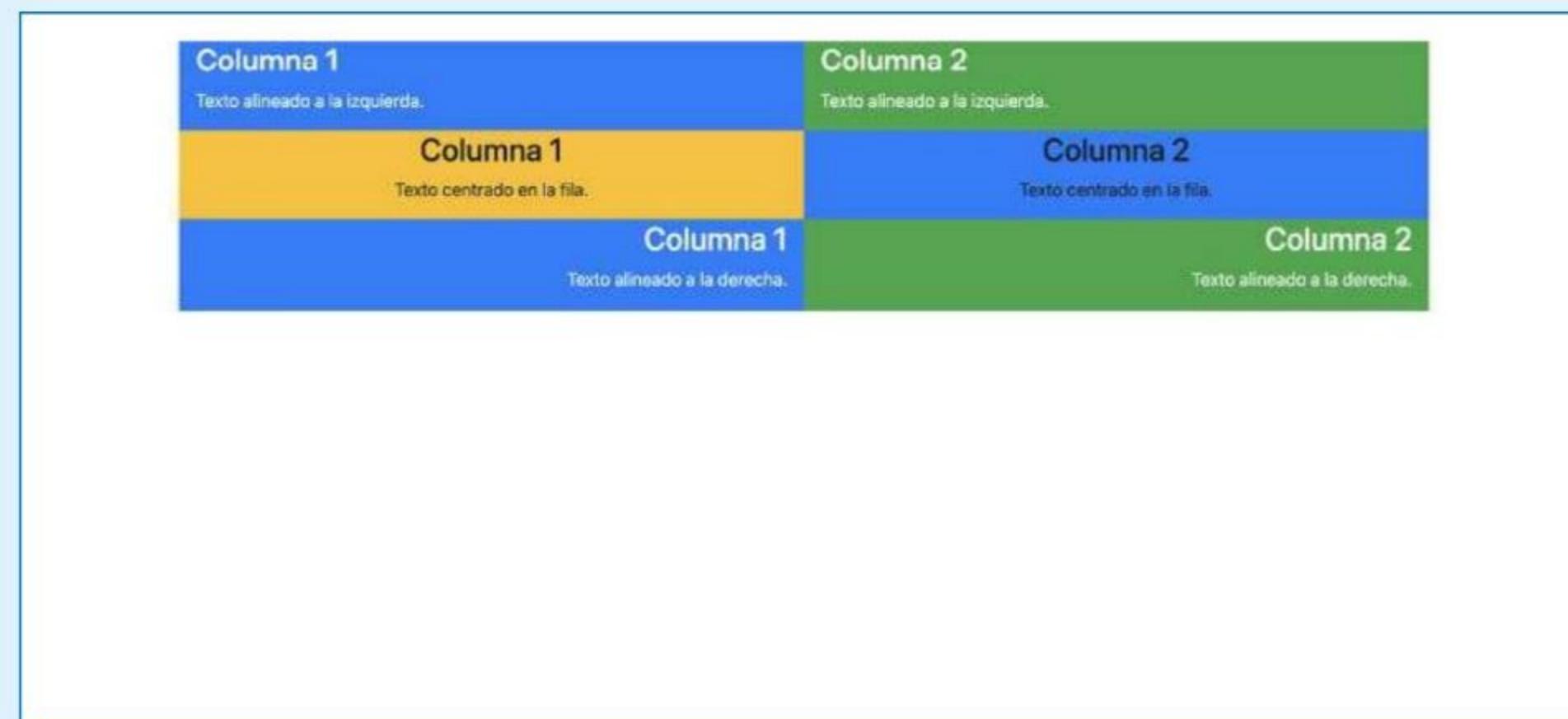
7. Implementa una interfaz web con columnas, empleando un sistema de cuadrícula Bootstrap. Imagina que se quiere crear una fila con tres columnas de diferente ancho en dispositivos pequeños y superiores.

- La primera columna ocupa 4 columnas en dispositivos extrapequeños y 6 en dispositivos pequeños y superiores.
- La segunda columna ocupa 4 columnas en dispositivos extrapequeños y 6 en dispositivos pequeños y superiores.
- La tercera columna ocupa 4 columnas en dispositivos extrapequeños y 12 en dispositivos pequeños y superiores.
- * *Nota para el ejercicio:* se crea una fila con tres columnas. En dispositivos extrapequeños (ancho de pantalla menor de 576 px), cada columna ocupa el 33,33 % del ancho total, ya que se especifica `col-4`. En dispositivos pequeños y superiores, la primera y la segunda columna ocupan el 50 % del ancho, y la tercera columna ocupa el 100 % del ancho (una columna completa), como se especifica con `col-sm-6` y `col-sm-12`, respectivamente.

8. Diseña utilizando Bootstrap la estructura que se muestra. La página debe contener dos secciones principales, cada una con filas y columnas que presenten información diferenciada. Cada sección debe estar destacada con un fondo de color. Para dar el color utiliza `bg-primary`, `bg-danger`, `bg-warning` y `bg-success`. Ejempl: `<div class="bg-warning p-3">`. Además, incluye un *padding* de al menos 3 px, `p-3`.

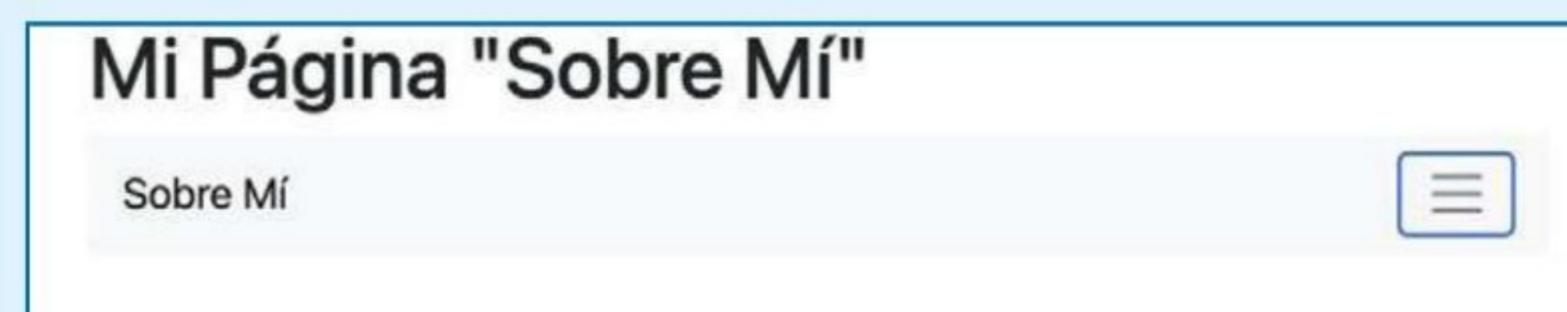


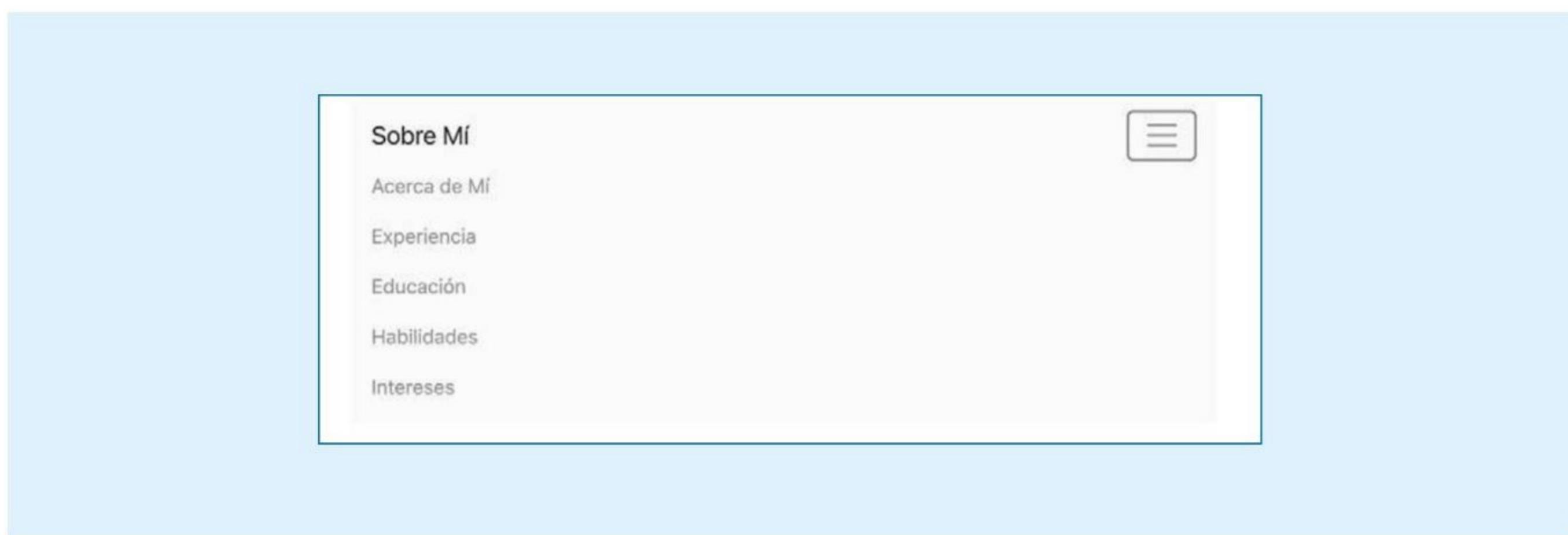
9. Diseña utilizando Bootstrap la estructura que se muestra. Utiliza las clases y propiedades de alineación.



10. Diseña la página “Sobre mí”, que se encuentra típicamente en los sitios web en los que se presenta al equipo o al autor. Se tendrán que aplicar conceptos de espaciado entre cajas, colores, diseño de *grid*, tipografías y tablas. Los requisitos de diseño son los siguientes:

- Espaciado entre cajas:
 - Debe constar de varias secciones: “Acerca de mí,” “Experiencia,” “Educación”, “Objetivo del sitio”, etc.
 - Utiliza las clases de Bootstrap para controlar el espaciado entre los elementos en cada sección.
- Colores:
 - Selecciona una paleta de colores adecuada para representar el estilo personal que se le quiera dar al sitio. Utiliza colores de fondo y de texto de manera coherente en toda la página.
- Diseño de *grid*:
 - Crea una cuadrícula de columnas utilizando Bootstrap para organizar el contenido de cada sección de tu página “Sobre mí”.
 - Utiliza columnas para mostrar imágenes de perfil, logros, proyectos personales, etc.
- Tipografías:
 - Utiliza fuentes tipográficas que reflejen la personalidad o el estilo del sitio.
 - Usa las clases de Bootstrap para aplicar estilos tipográficos, como negrita, cursiva, tamaño de fuente, etc.
- Barra de navegación: al pulsar sobre las tres líneas, se despliega el menú, como se muestra en la imagen inferior.





ACTIVIDADES DE AUTOEVALUACIÓN

1. ¿Qué función tiene CSS en el desarrollo web?

- a) Controlar el aspecto de los sitios web.
- b) Escribir el contenido de las páginas web.
- c) Manejar la interactividad de los usuarios.
- d) Ninguna de las anteriores.

2. ¿Cómo se pueden añadir estilos a una página web utilizando CSS?

- a) Mediante la etiqueta <style> en la cabecera del documento HTML.
- b) Con la etiqueta <link> en la cabecera del documento HTML.
- c) A través de la etiqueta <css> en el cuerpo del documento HTML.
- d) Ninguna de las opciones anteriores es correcta.

3. ¿Qué permite lograr el uso de hojas de estilo externas en HTML?

- a) Incluir una misma guía de estilo en todas las páginas de un sitio web.
- b) Controlar la interactividad de los usuarios en el sitio web.
- c) Mejorar la accesibilidad del sitio web.
- d) Ninguna de las opciones anteriores es correcta.

4. ¿Qué elementos pueden controlarse mediante el modelo de cajas en CSS?

- a) Anchura y altura.
- b) Fondo y color.
- c) Espacio de relleno y borde.
- d) Todos los anteriores.

5. ¿Qué significa la herencia en CSS?

- a) Transmitir propiedades de un elemento padre a sus hijos.
- b) Eliminar las propiedades de un elemento.
- c) Ocultar el contenido de un elemento.
- d) Ninguna de las opciones anteriores es correcta.

6. ¿Qué versión de CSS aporta un mayor dinamismo en el diseño de la página web?

- a) CSS1.
- b) CSS2.
- c) CSS3.
- d) Ninguna de las anteriores.

7. ¿Qué novedades trae consigo CSS3 en comparación con versiones anteriores?

- a) Uso de múltiples imágenes de fondo, esquinas redondeadas, transparencias, etc.
- b) Creación de contenido multimedia.
- c) Control de la interactividad del usuario.
- d) Ninguna de las opciones anteriores es correcta.

8. ¿Cuál es el mecanismo utilizado para indicar que un elemento hereda el valor de una propiedad en CSS?

- a) *Inherit*.
- b) *Inheritable*.
- c) *Inherit-value*.
- d) *Heredity*.

9. ¿Qué elementos se pueden controlar a través del modelo de cajas en CSS?

- a) *Padding, border, content y margin*.
- b) *Width, height, background y color*.
- c) *Font, style, color y size*.
- d) Todos los anteriores.

10. ¿Cuál es la etiqueta utilizada para importar hojas de estilo externas en HTML?

- a) `<import>`
- b) `<style>`
- c) `<link>`
- d) `<css>`

SOLUCIONES:

1. a b c d

2. a b c d

3. a b c d

4. a b c d

5. a b c d

6. a b c d

7. a b c d

8. a b c d

9. a b c d

10. a b c d