

MAQUETACIÓN CON CSS

Elementos principales

Para maquetar una página web y crear su estructura visual (*layout*) hay que dividir la página en bloques. La estructura de una página web tendrá una serie de áreas que se utilizarán para mostrar el contenido de las distintas secciones.



Figura 1. Estructura típica de una página web

Diseño fijo, líquido, elástico

En algunas páginas web las medidas de los bloques permanecen inalterables aunque disminuya el ancho del navegador. En otras, el ancho de los elementos se adapta a la anchura de la ventana del navegador. En el primer caso estaríamos ante un **diseño fijo**. Para maquetar una página con un diseño fijo se utilizan unidades de medida fijas, normalmente en píxeles. Esta forma de diseño tiene la ventaja de que se controla totalmente el tamaño y la estructura de la página independientemente del dispositivo. Además el diseño suele ser más fácil. Por el contrario no admite bien el redimensionado del tipo de letra y se puede leer con dificultad si el tamaño o la resolución de la pantalla muy grande o muy pequeña.

Para hacer un **diseño líquido**, se utilizan unidades de medida relativas, generalmente porcentajes. Con esto se consigue que el diseño de la página se adapte a los distintos anchos o resoluciones de pantalla y se evita tener que utilizar la barra de desplazamiento horizontal si el ancho de la ventana es pequeño. En general, es preferible este tipo de diseño al fijo ya que aumenta la usabilidad y accesibilidad de la página. La principal desventaja reside en que si el ancho de la ventana es muy grande, las líneas de texto son excesivamente largas

disminuyendo la legibilidad. Lo mismo puede ocurrir si el ancho es muy pequeño ya que el tamaño de la línea de texto disminuye aumentando el número de saltos de línea. Para solucionar este problema se pueden utilizar los atributos `max-width` y `min-width`.

En lugar de porcentajes, se puede establecer como unidad de medida el tamaño de la letra. Se trata del **diseño elástico** en el que las medidas de los bloques se realizan con unidades `em`. De esta forma, al redimensionar el texto se redimensiona también el bloque.

Elementos de la estructura de la página

La figura 1 muestra un ejemplo de maquetación típico de una página web con sus elementos habituales. Cada uno de esos elementos podría dividirse en distintas secciones, por ejemplo, un menú de navegación principal, enlaces destacados, distintas noticias, enlaces a las distintas utilidades (mapa web, registro, carro de la compra, etc.), etc.

Elemento `body`

El elemento `body` será el que definirá todos los aspectos de presentación generales, como la familia de la fuente, color de fondo y primer plano, etc.

Normalmente, además en él se establece un valor de `margin` y `padding` a 0, para que el resto de los elementos se ajusten a los límites de la ventana del navegador.

Elemento contenedor

Los distintos elementos de la estructura, generalmente se incluirán dentro de un elemento `div` de nombre contenedor. El contenedor nos servirá para delimitar los límites de la página. Para los agentes basados en Mozilla, es interesante volver a marcar el margen y el relleno a 0 para que los elementos se ajusten a los límites del contenedor. Los márgenes y rellenos reales se marcarán en cada elemento.

Un ancho marcado en porcentajes, dará un aspecto flexible (líquido) al contenido. Un ancho marcado en unidades de medida absolutas (preferentemente pixel) daría un ancho fijo.

Elemento cabecera

Se utiliza para meter los elementos que se repiten en cada página (logo con el sitio, menú de navegación principal, barra de utilidades, etc.).

Elemento `pie`

Contiene los elementos del pie de página. Para no montar con los elementos superiores suele tener un atributo `clear:both`.

Áreas de estructura visual de la página (columna izquierda, contenido, columna derecha, etc.).

Según el esquema de la página tendrá una, dos, tres o más, según el número de columnas de la maquetación.



Codificación de la estructura de la página

Menos el elemento `body`, que está presente en la especificación de `html`, `html 4.0` (y por lo tanto `xhtml`) no dispone de elementos para marcar estas secciones, aunque esta situación cambiará con `html 5` que dispone de elemento `footer`, `header`, `section`, etc.

Por lo tanto habrá que utilizar elementos `div` para marcar cada una de estas secciones. Cada uno de estos `div` se identificarán mediante el atributo `id` de forma que podamos seleccionarlos fácilmente en las reglas `CSS` y darles la estructura que se desee.

Es importante recalcar que, puesto que estos elementos serán únicos para una página se utilizará al atributo `id` en lugar de `class`. Aunque ambos atributos se utilizan para identificar fragmentos de contenido `html`, cada elemento sólo puede tener un único `id` que **no se puede repetir** en la misma página. Por el contrario la misma clase se puede repetir en la misma página y un elemento puede pertenecer a varias clases.

```
<body>
<div id="contenedor">
  <div id="cabecera">
    <p>Contenido de la cabecera</p>
  </div>
  <div id="col-izq">
    <p>Contenido de la columna izquierda</p>
  </div>
  <div id="contenido">
    <p>Contenido principal</p>
  </div>
  <div id="col-der">
    <p>Contenido de la columna derecha</p>
  </div>
  <div id="pie">
    <p>Contenido del pie de página</p>
  </div>
</div>
</body>
```

Dos columnas con diseño líquido (ancho variable)

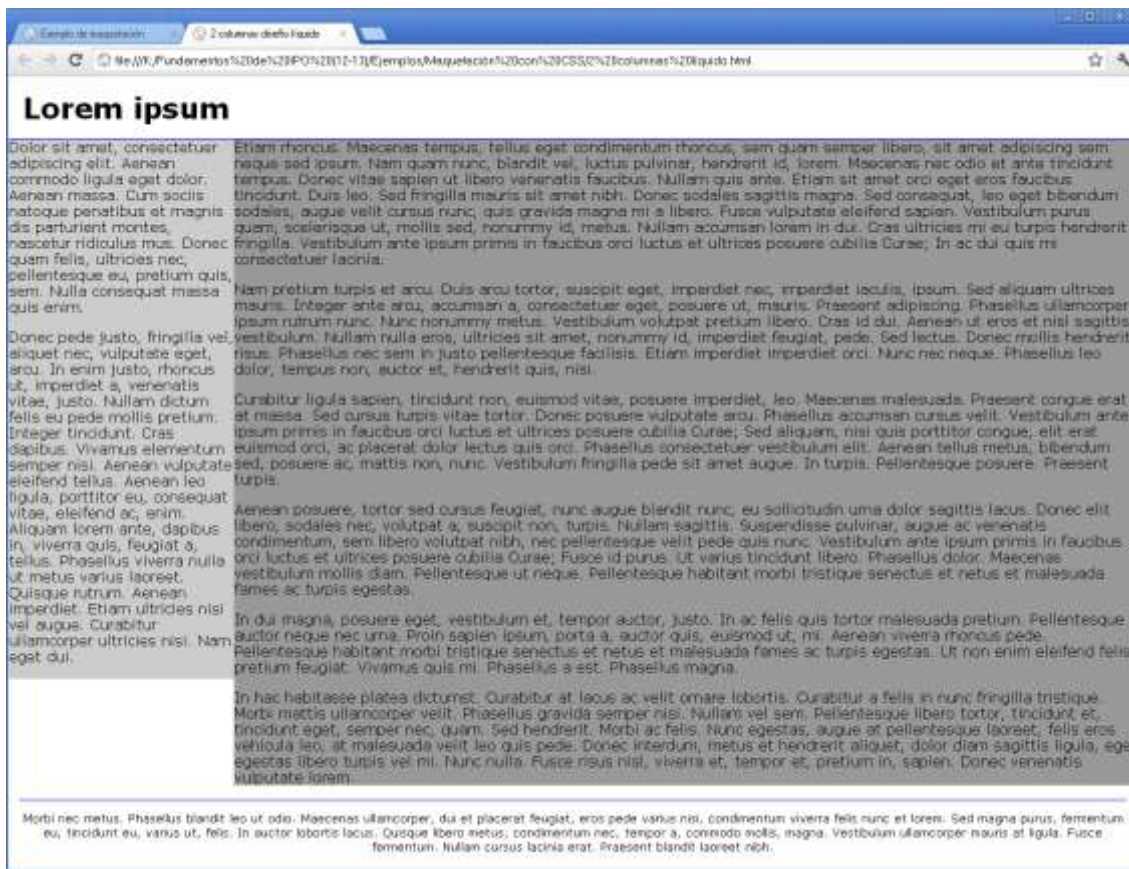


Figura 2. Una página a dos columnas con diseño líquido

Estilos para la estructura de la página

El problema principal de la maquetación es crear la estructura de las distintas secciones. En los siguientes ejemplos de layout de página los elementos `body`, `cabecera` y `pie` no cambian.

```
body {margin:0;
padding:0;
font-family: Verdana, Arial, Helvetica, sans-serif;
}

#cabecera{margin:0;
padding:0;
border-bottom:#0000ff solid thin;
}

#pie {clear:both;
margin:1em;
padding:1em 0 0 0;
text-align:center;
border-top:#0000ff solid thin;
font-size:80%;
}
```

En el caso del `pie`, para que los elementos no floten y el pie de página ocupe todo el ancho de la ventana se establece el atributo `clear` a `both`. El margen de `1em` permite separar un poco



el pie de página del elemento superior y que la línea que lo delimita no llegue hasta los bordes de la ventana. El relleno superior de 1em deja un poco de aire entre la línea y el texto.

```
#contenedor {width:100%;
              margin:0;
              padding:0;
              min-width:35em;
            }
```

En el elemento `contenedor`, en un diseño líquido, para que ocupe toda la ventana se establece un ancho del 100% del tamaño de la ventana del navegador (ocupa todo el ancho puesto que el margen y el relleno del bloque que lo contiene, el elemento `body`, están a 0). También se ha añadido el atributo `min-width`, para controlar el ancho mínimo del contenedor. La medida se ha establecido para que el ancho de la ventana no se reduzca tanto como para que se superpongan los textos de la columna izquierda con los del área de contenido. Si el usuario hace el ancho menor, aparecerán las barras de desplazamiento horizontal para que se pueda acceder a la parte derecha de la página.

La medida en porcentajes también se tiene que mantener para los bloques de la columna izquierda y el área de contenido. En la imagen, para distinguir claramente ambas áreas se les ha puesto un color de fondo diferente.

```
#col-izq {width:20%;
          float:left;
          background-color:#CCCCCC;
        }
```

En el elemento `col-izq`, el atributo `width` determina el ancho que va a ocupar el bloque. Para que el siguiente bloque (el contenido) flote sobre el borde derecho de la columna, se utiliza el atributo `float`. Con `float` igual a `left` el elemento se alinea a la izquierda y el siguiente bloque flota por su lado derecho.

```
#contenidos{background-color:#999999;
            margin-left:20%;
          }
```

El bloque de `contenidos` se colocará a la derecha de la columna izquierda. Se utiliza el atributo `margin-left` para que el margen se mantenga en toda la longitud de la página. Si no se hiciera, cuando el contenido sobrepasara el texto de la columna izquierda el texto la rodearía por debajo como aparece en la figura 3.

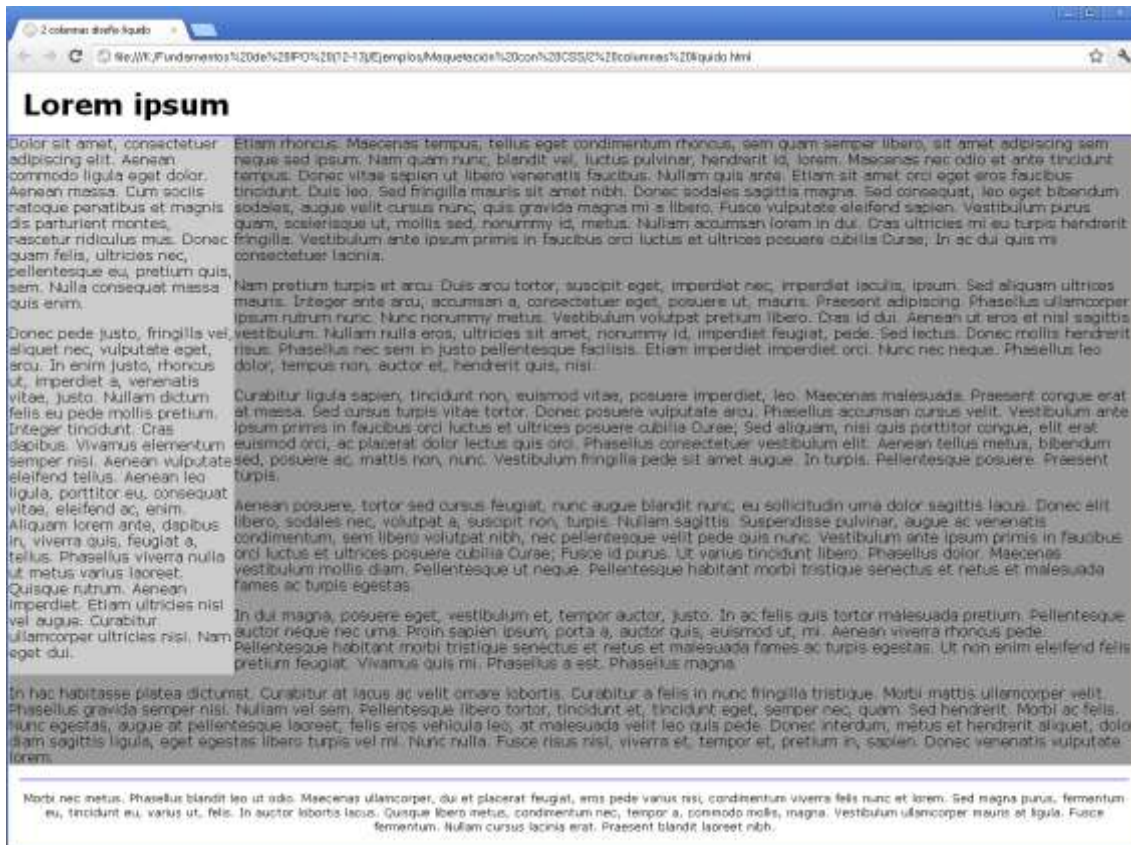


Figura 3. El texto del contenido (gris oscuro) se desborda por debajo de la columna izquierda (gris claro) si no se establece el margen izquierdo del área de contenido.

En ocasiones habrá que ajustar el ancho de la columna izquierda y el tamaño del margen izquierdo para que no ocupen exactamente el 100% (por ejemplo, que la suma de ambos sume el 98 o 99%). Esto puede ocurrir cuando los bloques que ocupan esas áreas utilizan valores para el margen, el padding o los bordes distintos de 0.

Dos columnas con diseño fijo



Figura 4. Una página a dos columnas con diseño fijo

Estilos para la estructura de la página

Como ya se ha indicado, los elementos `body`, `cabecera` y `pie` permanecen inalterables.

En el elemento `contenedor`, la anchura se establece en píxeles (en este caso 800px).

```
#contenedor {width:800px;
              margin:0 auto;
              padding:0;
            }
```

Con el atributo `margin:0 auto`, el margen superior e inferior se establece a 0, mientras que los márgenes izquierdo y derecho se establece a `auto`. Con esto se consigue que el contenido de la página se centre en la ventana del navegador, independientemente del ancho de la ventana. También se ha eliminado el atributo `min-width` que pierde su sentido en un diseño fijo.

En el elemento `col-izq`, además del atributo `float`, se establece el ancho mediante píxeles.

```
#col-izq {width:150px;
          float:left;
          background-color:#CCCCCC;
        }
```

Por último, también se establece en píxeles el margen izquierdo del elemento contenidos.

```
#contenidos{background-color:#999999;
            margin-left:150px;
          }
```

Tres columnas con ancho variable

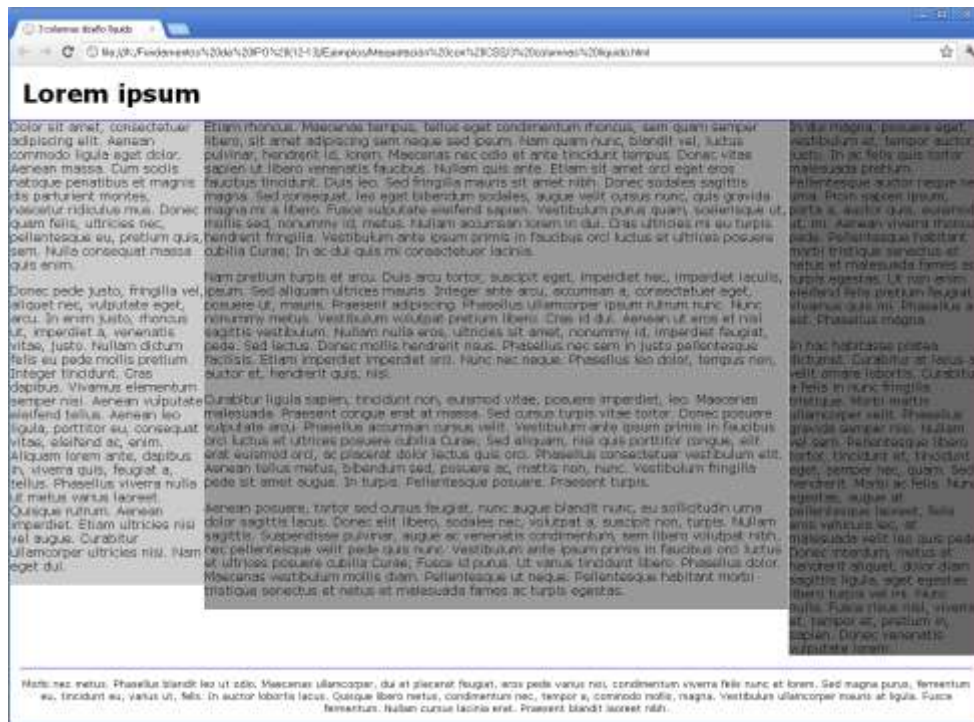


Figura 5. Una página a tres columnas con diseño líquido



Estilos para la estructura de la página

En este caso el área principal de la página tiene 3 elementos: `col-izq`, `contenidos` y `col-der`.

```
#col-izq {width:20%;  
        float:left;  
        background-color:#CCCCCC;  
}
```

El elemento `col-izq` se mantiene igual que en el caso de dos columnas. Sin embargo, el elemento `contenidos` ahora debe estar preparado para que el siguiente elemento, `col-der`, flote sobre su costado derecho. Al mismo tiempo, para que el contenido no ocupe todo el resto de la ventana, se delimitará la anchura mediante el atributo `width`.

```
#contenidos{background-color:#999999;  
            width:60%;  
            float:left;  
}
```

Por último, el elemento `col-der` deberá ocupar todo el espacio restante de la página y al mismo tiempo deberá desplazarse hacia la derecha. Por esa razón no lleva atributo `width` y se establece el valor de la propiedad `margin-left` a 80% (20% de la columna izquierda más 60% del contenido).

```
#col-der {background-color:#666666;  
         margin-left:80%;  
}
```

Tres columnas de ancho fijo



Figura 6. Una página a tres columnas con diseño fijo

Estilos para la estructura de la página

El único cambio que se produce es que las medidas de anchura tan del elemento contenedor, como de los elementos `col-izq`, `contenidos` y `col-der` se establece en píxeles en lugar de en porcentajes.

```
#contenedor {width:800px;
              margin:0 auto;
              padding:0;
            }

#col-izq {width:150px;
          float:left;
          background-color:#CCCCCC;
        }

#contenidos{background-color:#999999;
            width:500px;
            float:left;
          }

#col-der {background-color:#666666;
          margin-left:650px;
        }
```

Más de tres columnas

Cuando se tienen más de tres columnas, se procede de una forma similar: las primeras áreas se colocan utilizando los atributos `width` y `float`, mientras que en la última se establece utilizando el atributo `margin-left`.

```
#col-izq {width:25%;
          float:left;
          background-color:#CCCCCC;
        }

#contenidos{background-color:#999999;
            width:25%;
            float:left;
          }

#col-der{background-color:#666666;
          width:25%;
          float:left;
        }

#ultima {background-color:#999999;
          margin-left:75%;
        }
```



Figura 7. Una página a cuatro columnas con diseño elástico

Se puede utilizar este sistema siempre que se desee crear columnas en cualquier lugar de una página web.

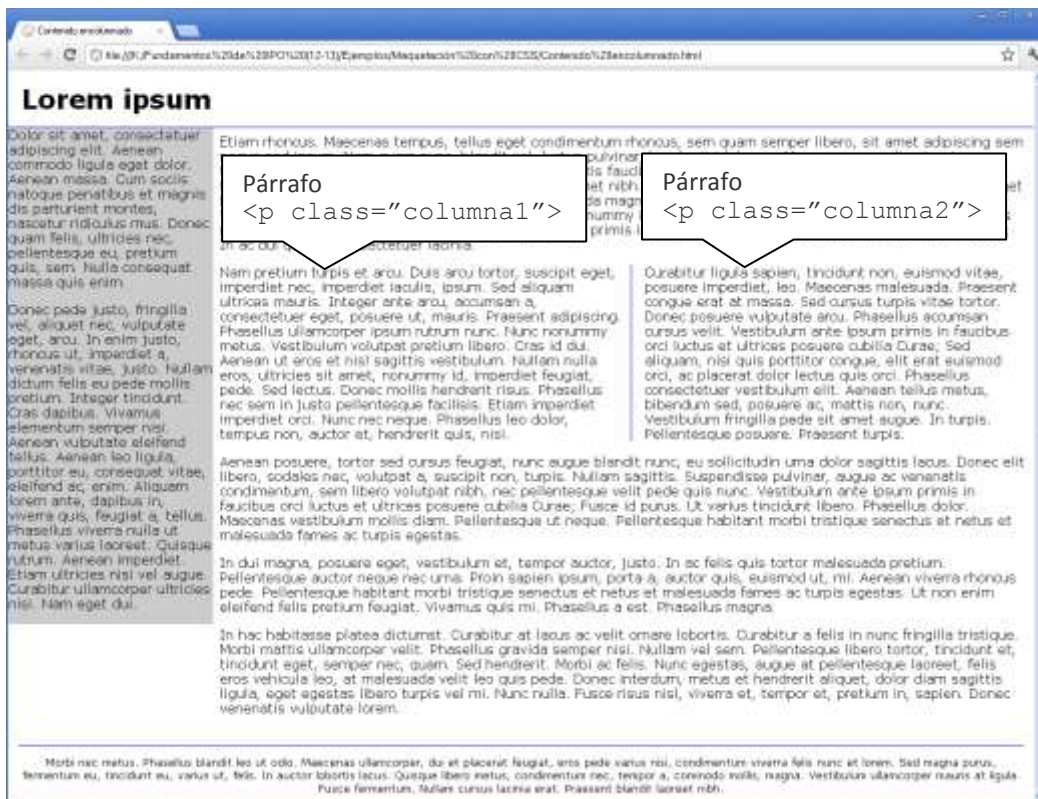


Figura 8. Dentro del elemento contenidos, aparecen dos párrafos encolumnados con el atributo `class="colimna1"` y `class="columna2"`

```
#col-izq {width:20%;
          float:left;
          background-color:#CCCCCC;
        }

#contenidos{padding:0.5em;
            margin-left:20%;
        }

p.columna1 {width:50%;
            float:left;
            }

p.columna2 {margin-left:51%;
            padding-left:1em;
            border-left: #0000ff solid thin;
            }
```

