

---

# Encabezamientos, pies, columnas y plantillas

---

PID\_00151050

Ben Henick

---

Tiempo mínimo previsto de lectura y comprensión: 3 horas

---



Universitat  
Oberta  
de Catalunya



# Índice

|   |          |
|---|----------|
| <b>1. Encabezamientos, pies, columnas y plantillas .....</b>  | <b>5</b> |
| 1.1. Los pasos críticos del proceso del estilista .....   | 6        |
| 1.1.1. Recopilación de objetivos .....  | 7        |
| 1.1.2. Clasificación de contenidos .....  | 7        |
| 1.1.3. Esbozo colaborativo y creación de un compuesto.....  | 7        |
| 1.1.4. Establecimiento de una estructura entre<br>y dentro de documentos .....                                    | 8        |
| 1.2. Requisitos, clasificación e imbricación de elementos<br>con más detalle.....                                 | 8        |
| 1.2.1. Definición de objetivos empresariales .....  | 8        |
| 1.2.2. Definición y cumplimiento de los objetivos<br>de los visitantes .....                                      | 9        |
| 1.2.3. Clasificación de contenidos .....  | 11       |
| 1.2.4. Orden del código: accesibilidad<br>y otras consideraciones .....   | 12       |
| 1.2.5. Asignación de id, clases y elementos contenedores .....  | 13       |
| 1.3. Implementación de la composición a una columna .....   | 14       |
| 1.3.1. Centrado de la composición .....   | 15       |
| 1.3.2. ¿Es estrictamente necesario un contenedor<br>de toda la anchura del documento? .....                       | 15       |
| 1.4. Implementación de la composición a dos columnas .....  | 15       |
| 1.4.1. Problemas de composición en relación<br>con las disposiciones a una y dos columnas .....                   | 16       |
| 1.5. Implementación de la composición a tres columnas .....   | 18       |
| 1.6. Visión amplia de los encabezamientos y pies .....  | 20       |
| 1.6.1. Sitio personal: Cindy Li .....   | 20       |
| 1.6.2. Redes sociales: Facebook .....   | 21       |
| 1.6.3. Marketing de empresa y servicio al cliente:<br>BNSF Railway .....  | 22       |
| 1.6.4. Diseño de encabezamiento y pie:<br>los detalles de bajo nivel .....  | 23       |
| 1.7. Webs con diferentes números de columnas: hacer trampa<br>con <code>class</code> y <code>display</code> ..... | 24       |
| Resumen.....  | 25       |
| Preguntas de repaso .....   | 25       |
| Lecturas complementarias .....  | 26       |



# 1. Encabezamientos, pies, columnas y plantillas

Ben Henick

Hasta ahora, los artículos de este curso se han centrado en temas concretos, desde conceptos “blandos” como la tipografía y el color hasta una instrucción técnica “dura” sobre partes de CSS 2.1. Este artículo tiene un enfoque más amplio; su finalidad es mostrar al lector cómo utilizar el material tratado hasta ahora para construir la plantilla de una web completa.

La principal asunción de este material es que ya estáis familiarizados con las propiedades `float` (flotar), `display` (visualizar) y `position` (posición) de CSS.

Los lectores autodidactas que deseen pasar directamente al jugo del CSS están invitados (contra nuestra voluntad) a pasar directamente al subapartado 4 de este módulo, pero deberán tener en cuenta que, al hacerlo, se perderán la lección sobre cómo una planificación satisfactoria del proyecto conduce a la composición e implementación de un sitio web.

Aquellas almas impacientes que ignoren las advertencias del párrafo anterior también deberán descargar las plantillas para una, dos y tres columnas que se facilitan en este artículo y se les enlazará de nuevo a su conclusión.

**Archivo fuente:** “Composición a una columna”

[http://dev.opera.com/articles/view/38-headers-footers-columns-templates/template\\_1col.html.txt](http://dev.opera.com/articles/view/38-headers-footers-columns-templates/template_1col.html.txt)

**Archivo fuente:** “Composición a dos columnas”

[http://dev.opera.com/articles/view/38-headers-footers-columns-templates/template\\_2col.html.txt](http://dev.opera.com/articles/view/38-headers-footers-columns-templates/template_2col.html.txt)

**Archivo fuente:** “Composición a tres columnas”

[http://dev.opera.com/articles/view/38-headers-footers-columns-templates/template\\_3col.html.txt](http://dev.opera.com/articles/view/38-headers-footers-columns-templates/template_3col.html.txt)

## Nota

Podéis descargar todo el código de ejemplo en un cómodo paquete para hacer pruebas en vuestro equipo local.

**Podéis descargar el código de ejemplo en:** “layout\_template\_code.zip”

[http://dev.opera.com/articles/view/38-headers-footers-columns-templates/layout\\_template\\_code.zip](http://dev.opera.com/articles/view/38-headers-footers-columns-templates/layout_template_code.zip)

Una web construida adecuadamente suele ser el producto de un proceso deliberado, principalmente por entregas, aunque el sitio lo creen una o dos personas en vez de un equipo entero de especialistas. En la figura 1 se describe una representación bastante elaborada de este proceso y de los diez pasos que aquí se describen; este artículo se centra en cuatro, en concreto:

1) Recogida de requisitos en función de los objetivos empresariales y los consiguientes objetivos de los visitantes.

Podéis ver el subapartado 1.4 de este módulo.



- 2) Clasificación de contenidos.
- 3) Borrador colaborativo y creación de un compuesto.
- 4) Establecimiento de la estructura de los documentos que se deben utilizar en la web.

Una vez completados estos cuatro pasos, el estilista tiene la mayor parte de la información que necesita para crear las disposiciones de la web, que generalmente tendrán una, dos o tres columnas. Independientemente de la forma general de la composición, habrá diferencias de una sección de una web a otra, que a su vez influirá en cómo algunos elementos y selectores de estilo pasan a formar parte del diseño global de la web.

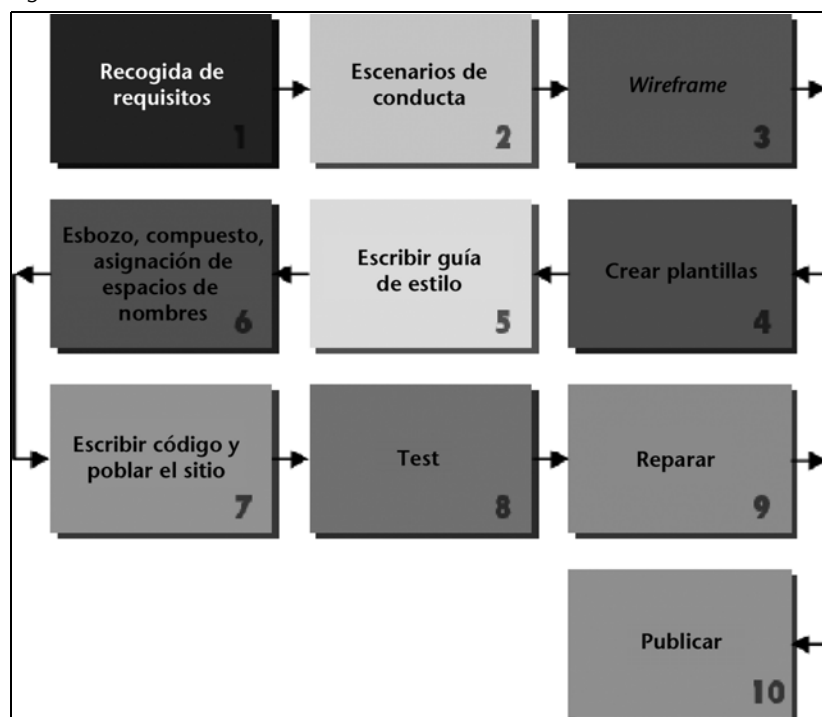
Incluso después de haber hecho las selecciones de diseño y estructurales, queda la cuestión de cómo se tratará la producción si la web se debe crear sobre un sistema de gestión de contenidos (CMS) como Wordpress o Drupal.

Este artículo subraya la importancia de los cuatro pasos mencionados anteriormente; proporciona un marco único para la clasificación de contenidos y describe cómo componer una web con todos los encabezamientos, pies y columnas.

### 1.1. Los pasos críticos del proceso del estilista

Las dos secciones siguientes se facilitan como embellecimiento de otros artículos de este currículo, con un énfasis en la planificación y el proceso más que en la implementación. El mensaje principal es: “Mirad antes de saltar”; en otras palabras, debéis comprender claramente lo que vais a implementar antes de empezar con el etiquetado, las hojas de estilos y el código.

Figura 1



Diez pasos comunes del proceso de creación de una web. Los que son especialmente relevantes para este artículo se muestran en el tramo inverso.

### 1.1.1. Recopilación de objetivos

El buen CSS se basa en una cadena de dependencias:

- el CSS se basa en una estructura para trabajar;
- la estructura se informa mediante el contenido;
- el contenido necesita un ámbito;
- el ámbito del contenido se define mediante los posibles objetivos de visitantes, y
- a su vez, los objetivos de los visitantes están relacionados con los objetivos empresariales.

La cuestión de esta cadena de requisitos es que las necesidades de vuestros visitantes señalarán directamente la estructura que debéis crear para vuestro sitio y, por lo tanto, impulsarán los selectores y las técnicas aplicadas a vuestros estilos.

Cuando el proceso de recogida de requisitos se gestiona incorrectamente, los especialistas en estilos se enfrentan a retos como los siguientes:

- Falta de activos.
- Falta de orientación respecto a la relación entre el comportamiento de la composición y la geometría de las ventanas.
- Falta de definición de los niveles de soporte de plataformas.
- Solicitudes frecuentes de cambios, ya que los requisitos se acoplan durante el proceso de implementación.
- Falta de herramientas actuales.

Podéis ver el apartado 8 del módulo "CSS".



### 1.1.2. Clasificación de contenidos

Una vez que hayáis determinado el ámbito del contenido de vuestra web, debéis definir su importancia para el sitio en conjunto y decidir qué herramientas de navegación proporcionaréis a los visitantes para encontrarlo.

También debéis tener en cuenta la cuestión de qué hacer con la publicidad, las listas de enlaces, las galerías y los comentarios.

Una vez clasificados, los contenidos se pueden asociar entonces a la navegación de la web y ponderarlos. La ponderación de contenidos se tratará aquí en un contexto primario/secundario/terciario, dado que la importancia de un bloque de contenidos la debería reflejar su posición en el orden original del documento.

### 1.1.3. Esbozo colaborativo y creación de un compuesto

Si estáis trabajando en un proyecto en el que el diseño gráfico y la implementación de estilos son tareas asignadas a diferentes personas (como suele ser el caso en los proyectos comerciales), el diseñador gráfico deberá basarse en los

diagramas de mapa del sitio y los esqueletos (si es el caso) para empezar a crear el aspecto general de la web. Puede ser útil empezar con borradores sencillos, como manera de definir elementos como el motivo general del sitio, sus características consistentes y los detalles que podrían complicar el etiquetado o la asignación de `class` e `id`.

Una vez acordados los borradores, el diseñador gráfico puede pasar entonces a los compuestos acabados, que deberían ser parecidos a capturas de pantalla que se podrían generar en el navegador de desarrollo una vez puesta la web en producción.

De ahora en adelante, en este artículo, el diseño gráfico se tratará de una manera rápida porque ya se ha explicado en los artículos sobre esqueletos y compuestos/modelos.

#### **1.1.4. Establecimiento de una estructura entre y dentro de documentos**

Con los compuestos a mano, el especialista en estilos puede empezar a escribir el etiquetado y los CSS. El primer paso de este trabajo es decidir el orden de los contenidos, el anidado de elementos, clases e `ids` que se utilizarán en la web de producción; un trabajo que sólo se puede hacer bien si el especialista en estilos comprende completamente el contenido que se debe ofrecer en la web y cómo se debe disponer.

### **1.2. Requisitos, clasificación e imbricación de elementos con más detalle**

Si se siguen las buenas prácticas modernas y se adopta el paradigma de diseño centrado en el usuario (UCD, *user centered design*) para motivar las decisiones de diseño y desarrollo, entonces los objetivos de los visitantes influyen en cada uno de los pasos del proceso de diseño.

No obstante, de hecho los objetivos del patrocinador (o editor) de la web tienen prioridad desde el punto de vista del equipo de diseño, porque sin un conocimiento de aquellos objetivos, el equipo de diseño no tiene manera de prever los objetivos de los visitantes.

#### **1.2.1. Definición de objetivos empresariales**

Los objetivos “empresariales” de la web entrarán en una o varias de las siguientes definiciones generales:

- Generación directa de ingresos (comercio electrónico).



- Ofrecimiento de publicidad, mensajes y/o servicios de almacenamiento mediante una interfaz web (como, por ejemplo, Blogger, Flickr, Scribd, YouSendIt o Basecamp).
- Comercialización de un producto o servicio.
- Facilitación de información.
- Entretenimiento del visitante.

Una vez establecidos los objetivos generales, habrá más perfeccionamiento en función de cualquiera de una serie de factores, como la demografía, los objetivos de conversión y las limitaciones de diseño impuestas por el presupuesto del proyecto o la naturaleza del propio contenido (como podría ser el caso con el vídeo Flash).

La experiencia es, de lejos, la mejor guía en esta parte del proceso de recogida de requisitos.

### 1.2.2. Definición y cumplimiento de los objetivos de los visitantes

Una vez establecidos los objetivos empresariales, es decir: “esto es lo que queremos que vean y hagan nuestros visitantes”, el paso siguiente es atraer y conducir a los visitantes a aquellos destinos y tipos de funcionalidades que les resulten más adecuados.

La principal asunción de esta parte del proceso de diseño es que “los visitantes odian los obstáculos”. Las mejores maneras de tener en cuenta este supuesto son las siguientes:

- Definir y diseñar una navegación que se adapte de la mejor manera posible a las estrategias de búsqueda más probables del visitante.  
Estas estrategias pueden incluir búsqueda de texto completo, listas tradicionales taxonómicas de enlaces o “etiquetado” (ya sea moderado o definido por el usuario).

#### Nota

Es conveniente que tengáis en cuenta que una web puede contemplar más de una estrategia de búsqueda.

- Ofrecer siempre indicaciones para que el visitante sepa dónde se encuentra en la web en conjunto. Entre los medios habituales de hacerlo están los “rastros de migas de pan” o “breadcrumbs”, enlaces a contenido relacionado e indicaciones visuales como, por ejemplo, el diseño de enlaces de navegación dependientes del contexto.
- Reforzar el estilo visual y escrito de la manera más rígida posible, sin caer en una consistencia ridícula. Esta práctica está íntimamente relacionada

con la necesidad de dar al visitante pistas consistentes sobre la ubicación; la diferencia es que mientras el color y la disposición de la navegación ofrecen al usuario orientación por toda la web, una presentación consistente a lo largo de las páginas es *esencial* para orientarse en una única página.

- Indicar siempre en lenguaje sencillo las consecuencias de seguir un enlace o enviar un formulario. A veces esto es tan sencillo como etiquetar un botón de enviar con la palabra “Búsqueda”, pero en ocasiones es posible que debáis añadir una nota para dar más instrucciones a los visitantes de la web.
- Proporcionar una distinción clara entre los elementos de diseño que responderán a la interacción del usuario y cualquier otra cosa. Los estilos que hacen que los enlaces sean casi indistinguibles del texto normal, el diseño inconsistente de los botones y los estilos de `cursor` atípicos son muy comunes y confusos. Los colores con un contraste elevado, el uso meditado de `padding` (para ampliar el espacio que ocupan los elementos de diseño interactivos), y los `title` informativos suelen ser más eficaces.
- Minimizar la cantidad de interacción del usuario (especialmente el número de clics en enlaces o botones) necesaria para alcanzar un objetivo común, como una compra o el servicio de recursos populares. En la práctica, esto generalmente requiere ponerse en la piel del visitante y analizar sus opciones desde su punto de vista. A no ser que estas tareas se puedan completar y hasta que esto sea posible, el atajo para cumplir esta directriz es seguir el espíritu del principio KISS\*.

\* <http://dev.opera.com/articles/view/opera-web-standards-curriculum-glossary/#kiss>

Cuando se trata de etiquetado y hojas de estilos, hay ciertas técnicas muy sencillas que hacen que estas reglas sean fáciles de seguir:

- **A la hora de elaborar vuestra hoja de estilos, debéis asignar las máximas propiedades posibles con reglas mediante selectores de elementos simples.**

Las `id`, por definición, son únicas y, para ser útiles, las clases se deben reservar preferentemente para casos poco comunes (o para requisitos de presentación que los navegadores antiguos no aceptarían correctamente de otra forma). No obstante, moderando su uso, los especialistas en estilos más atentos están al tanto de las ocasiones en las que el diseñador gráfico insiste en niveles de detalle potencialmente caros en sus diseños.

- **Debéis asignar una `id` al `body` de cada página.**

Si se asignan etiquetas de hojas de estilos a documentos determinados (y no sólo en las secciones de una web), los casos de presentación únicos son más fáciles de solucionar. Otra ventaja de colocar una `id` en el texto principal de cada página es que cuando se utiliza conjuntamente con elementos de navegación ponderados de manera similar, el estilista puede

proporcionar indicaciones visuales en la navegación primaria para elementos como la sección o web visualizada actualmente, sin necesidad de utilizar lógica verbosa en el lado del servidor.

- **Debéis evitar los diseños que exigen un elevado control motriz por parte del visitante.**

Una manera más directa de describir esta instrucción es “debéis evitar los menús volantes como los creados mediante las técnicas Suckerfish\*, también conocidos como menús desplegables”. Hay casos claros de uso de estos diseños y todos implican webs grandes basadas en disposiciones de una y dos columnas, pero a menudo se pueden evitar. Por otra parte, los usuarios sin experiencia y con discapacidad de control motriz a menudo encuentran problemáticos los menús desplegables:

- Para ser efectivos, estos elementos suelen requerir que los enlaces que contienen sean más pequeños que el tamaño predeterminado del tipo de letra del texto, una indicación visual contraintuitiva que hace que los elementos de menú tenga menos aspecto de enlaces que otros elementos de la página que les rodean.
- El nivel de control motriz necesario para utilizar estos elementos frustra fácilmente a los visitantes sin experiencia, ocasionales y discapacitados.
- La variedad de posibles destinos disponibles que hay en una sección determinada de la web permanece oculta hasta que el visitante interactúa con estos elementos y esto limita su capacidad de mantener un sentido de ubicación en la web hasta que alcanzan experiencia en su uso.

\* Podéis ver “Suckerfish Dropdowns”, de Patrick Griffiths y Dan Webby.  
<http://www.alistapart.com/articles/dropdowns>  
Podéis ver “Son of Suckerfish Dropdowns”, de Patrick Griffiths y Dan Webby.  
<http://www.htmldog.com/articles/suckerfish/dropdowns/>

### 1.2.3. Clasificación de contenidos

Una vez calculado el alcance de los contenidos que se deben presentar en la web, podéis crearle una arquitectura. La arquitectura de un sitio se puede elaborar de varias maneras (podéis ver algunos ejemplos en el artículo 23).

Generalmente, podréis asignar prioridades a vuestros contenidos, que informarán de las composiciones posibles:

- 1) Los **contenidos primarios** son la materia en torno a la que se construyen los documentos de destino individuales, como artículos, álbumes de fotos o conjuntos de datos.
- 2) Los **contenidos secundarios** incluyen metadatos legibles para las personas que están adjuntas a los contenidos primarios, y también contenidos de barra

lateral (por ejemplo, texto de exposición; extractos de críticas; enlaces a artículos relacionados en el sitio; listas de gráficos, mapas o tablas).

3) Los **contenidos terciarios** comprenden los enlaces salientes a materiales relacionados (como la lista de enlaces amigos de un blog), instantáneas persistentes de contenidos de otras fuentes como webs de redes sociales o entradas de comentarios y publicidad.

Además de los contenidos, vuestro formato casi seguro incluirá dos secciones más:

- El **encabezamiento** incluirá el título de la web (que a menudo enlaza con la portada de la web), la navegación principal, enlaces a metadatos de la cuenta de usuario (en una aplicación) y, por último, el formulario de búsqueda de contenidos principales (si está implementado).
- El **pie** contiene la declaración de *copyright*, como mínimo. Los enlaces a documentos compuestos de metadatos (como canales RSS, *sitemaps* y la parte de metacontenidos de la web de la información de contacto) también encuentran su camino en la navegación secundaria de una web, que suele formar parte del pie.

La navegación principal y el título de una web casi siempre forman parte de (o están al mismo nivel que) el encabezamiento en un contexto visual; cada implementador puede decidir si deberán formar parte del encabezamiento de la web a nivel de etiquetado.

#### 1.2.4. Orden del código: accesibilidad y otras consideraciones

Uno de los primeros pasos del diseño de plantillas web es decidir el orden de sus contenidos en el código, que debería ser consistente a lo largo de la web en conjunto.

El orden del código del documento de manera que se pueda leer correctamente sin la ventaja de las hojas de estilo es esencial por motivos de accesibilidad y soporte entre medios. En el primer caso, los usuarios con discapacidad visual pueden utilizar lo que se denomina lectores de pantallas: aplicaciones de software que leen los contenidos en voz alta al visitante, contenidos que no tienen ningún sentido si están ordenados de cualquier manera por motivos de presentación.

...Y de la misma manera que la información dispuesta en el orden adecuado para la visualización en pantalla probablemente exige claridad para ser leída en voz alta, aplicar estilos puede resultar imposible en otros medios como vi-

sualizaciones impresas o móviles. En este caso, el resultado suele ser la duplicidad de contenidos, lo que presenta una serie de inconvenientes:

- Como mínimo, se debe implementar lógica adicional en la capa de vista para tener en cuenta el hecho de que la salida de un registro individual de una base de datos se puede ofrecer de más de una manera.
- En el peor de los casos, los contenidos se duplican no sólo en el entorno en el que se encuentra el visitante, sino también en la base de datos o en el sistema de archivos del sistema principal. El resultado será la duplicidad de algunos gastos de mantenimiento.

Por lo tanto, lo más habitual es disponer las secciones más exteriores en el orden siguiente:

#### 1) Encabezamiento

- Título [preferiblemente enlazado con la página de inicio]
- Navegación principal

#### 2) Contenido principal

- Título del documento
- Texto central

#### 3) Contenido secundario

#### 4) Contenido terciario

#### 5) Pie

- Navegación secundaria
- Partes adicionales (por ejemplo, aviso de *copyright*)

La manera como se imbricarán estas secciones dependerá de una serie de requisitos variables, de los que el más habitual es el número de columnas estáticas en la disposición de la web.

### 1.2.5. Asignación de *id*, clases y elementos contenedores

Cuestiones de taxonomía al margen, podemos asumir que cualquier web dada cubrirá una serie de temas relacionados dentro de un ámbito entendido, ya sean operaciones y productos de una empresa, tipos específicos de acontecimientos, o tipos específicos de entretenimiento, por nombrar algunos ejemplos con los que suelen contar para atraer el interés de públicos no técnicos.

Por lo tanto, el estilista se encontrará probablemente asociando las etiquetas de hojas de estilos tanto con elementos estructurales de los contenidos de la

web (por ejemplo, fragmentos de navegación, encabezamiento, texto principal) como con abanicos de contenidos, más o menos amplios.

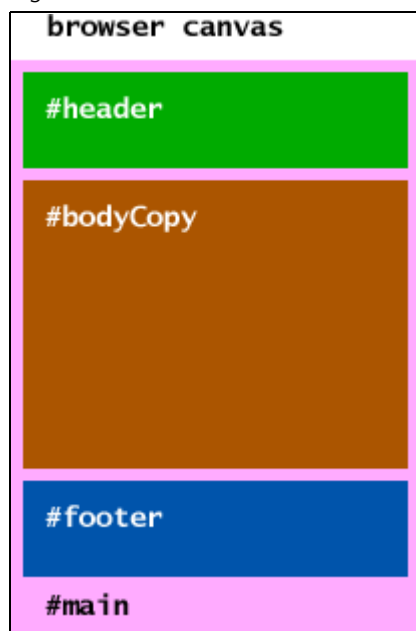
Las prácticas varían, pero el autor generalmente asigna las siguientes etiquetas de formato a sus plantillas, etiquetas que verán su uso en este módulo.

- `#main`: lienzo de contenidos
- `h1` (único): título de la web
- `ul#nav`: código de navegación del sitio
- `#breadcrumb`: rastro de “migas de pan” (si se utiliza)
- `#bodyCopy`: artículo principal
- `#bodyCopy>h2` (único): título de documento principal
- `#sidebar`: contenido secundario
- `#footer`: código del pie
- `ul#secondaryNav`: navegación secundaria

Además de éstos y mucho más importante, se añade una `id` al `body` de cada página (como se ha mencionado anteriormente) que da alguna indicación del ámbito del contenido primario asociado con todo el documento. Algunos proyectos también generan un requisito para la asignación de `classes` a los elementos `body`.

### 1.3. Implementación de la composición a una columna

Figura 2



Elementos de la composición de una columna; probablemente el etiquetado se imbricará como se muestra aquí.

La figura 2 muestra que `#main` está inmediatamente dentro de `body`, y que a la vez contiene todo el `#header`, `#bodyCopy` y `#footer`.

### 1.3.1. Centrado de la composición

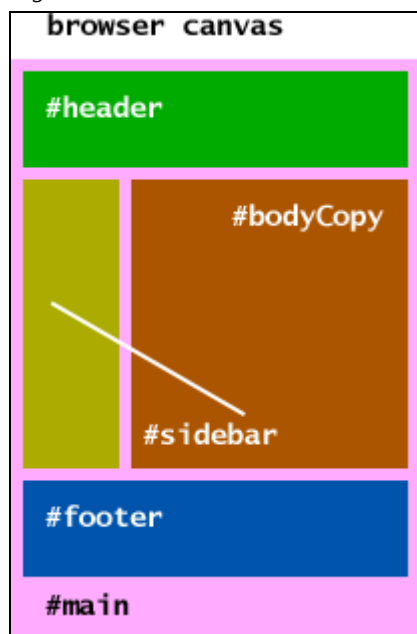
El centrado del lienzo de contenidos es una cuestión de insertar (en este caso) `#main {width: 960px; margin: auto; }` en vuestra hoja de estilos (el valor `width` elegido es arbitrario).

### 1.3.2. ¿Es estrictamente necesario un contenedor de toda la anchura del documento?

En una web que se basa completamente en composiciones de una sola columna, no es absolutamente necesario incluir `#main`; se podrían aplicar igual de fácilmente los mismos pares de propiedad/valor de composición utilizados anteriormente en `#header`, `#bodyCopy` y `#footer` de manera combinada. No obstante, no hay nada semánticamente incorrecto en incluir `#main` y su inclusión ofrece al especialista en estilos más flexibilidad en relación con elementos como las reglas, los canales, las imágenes de fondo y la construcción de la prominencia de determinados elementos en la estructura de la plantilla.

## 1.4. Implementación de la composición a dos columnas

Figura 3



Elementos de una composición en dos columnas; probablemente el etiquetado estará anidado como se muestra aquí, con la condición de que `#sidebar` seguirá de hecho a `#bodyCopy` en el orden de origen.

La diferencia entre las composiciones a una y dos columnas es el añadido de un segundo elemento contenedor para el contenido secundario (`#sidebar`, que de hecho sigue a `#bodyCopy` en el orden de origen), y algunos cambios en la hoja de estilos que probablemente se utilizarán para una disposición a una columna.

### 1.4.1. Problemas de composición en relación con las disposiciones a una y dos columnas

Los cambios en el etiquetado necesarios para convertir una plantilla de una columna en una plantilla de dos columnas son sencillos, pero en muchos casos las nuevas reglas de estilo necesarias no lo son.

#### Colocación de `#sidebar` a la izquierda en vez de en su orden original

Hay dos procedimientos para hacerlo; uno funcionará independientemente de la longitud de cualquiera de las columnas, mientras que el otro requiere que `#bodyCopy` sea más largo que `#sidebar`.

El planteamiento principal y más común es utilizar `float`:

- 1) Asignar un valor `width` a `#bodyCopy`.
- 2) Añadir a esta regla un valor `float` de `right`.
- 3) Asignar un `width` apropiado a `#sidebar`.
- 4) Asignar valores de `margin` (margen) y `padding` (relleno) según sea necesario en `#bodyCopy` (texto central) y/o `#footer` (pie) para garantizar que exista el canal deseado entre los dos elementos.
- 5) Asignar `clear: both;` a `#footer`.

Las dos columnas tienen los valores `width` (anchura) especificados de manera que tengan unos márgenes consistentes.

En el caso en el que `#sidebar` queda en el margen derecho de la composición, los pasos ya descritos se deberían continuar siguiendo, aunque el valor de `float` de `#bodyCopy` se debería establecer en `left`. El valor `width` asignado a `#sidebar` se debería sustituir por un valor `margin-left` (margen izquierdo) apropiado.

También es posible asignar un valor `margin-left` o `margin-right` (margen derecho) grande (según sea necesario) al elemento del que se ha eliminado `float`, en vez de `width`.

Un segundo planteamiento que probablemente provocará menos errores en Internet Explorer 6 es asignar un valor `margin-left` o `margin-right` grande a `#bodyCopy` según convenga, y posicionar `#sidebar` de manera absoluta. No obstante, este planteamiento es menos flexible porque los casos en los que `#sidebar` sea más largo que `#bodyCopy` harán que el elemento anterior sangre hacia `#footer`.



## Columnas falsas: uso de una imagen de fondo para alinear las longitudes de las columnas cuando la longitud de su contenido es diferente

Un examen más minucioso de las implementaciones de columnas utilizando la propiedad `float` revela que cuando se quieren diferentes colores de fondo o reglas verticales entre columnas, no se puede contar con ello para establecer la longitud del área de contenidos principal cuando se aplica mediante propiedades `background-color` o `border`.

La solución más fácil a este problema es crear y especificar una imagen de fondo (generalmente de un píxel de altura) a uno de los elementos antepasados de las columnas que, a causa de su asignación a este elemento antepasado, siempre se repetirá desde la parte superior a la parte inferior de la columna más alta relacionada. Por lo tanto:

```
#main {  
  background-image: url(images/bg_2column.gif);  
  background-repeat: repeat-y;  
}
```

Si `bg_2column.gif` está formado por dos bandas de colores muy contrastados que corresponden con más o menos precisión a las anchuras asignadas en las columnas de contenido, el resultado serán dos columnas que parecerán tener la misma altura... aunque realmente no la tengan, como se descubriría si a continuación se insertaran las reglas siguientes:

```
#bodyCopy {  
  background-color: #ccc;  
}  
  
#sidebar {  
  background-color: #999;  
}
```

Realizar este paso no elimina necesariamente la necesidad de aplicar propiedades `color` o `background-color` a una columna determinada; si el color del texto predeterminado no es legible sobre una o ambas columnas, sus colores de fondo y primer plano se deberían especificar explícitamente en la hoja de estilos como salvaguarda contra la posibilidad de que el navegador no cargue las imágenes de fondo.

## Desplazamiento de la navegación principal a la segunda columna manteniendo el orden del código

Una vez que se ha añadido una segunda columna a la disposición, probablemente parecerá natural colocar la navegación principal de la web en la parte

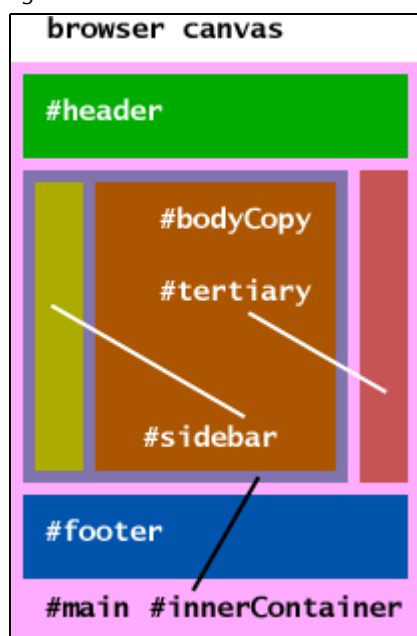
superior de aquella columna... sin embargo, ¿cómo hacerlo cuando la navegación descansa en una parte diferente de la estructura de la plantilla?

La respuesta a esta pregunta se encuentra en el posicionamiento:

- 1) Si habéis asignado `overflow: hidden;` en `#header`, (encabezamiento), haced que `#nav` sea un descendiente directo de `#main` (principal). En casi todos los casos, será posible hacerlo sin estropear el orden de origen deseado.
- 2) Asignad `position: relative;` (posición relativa) al antepasado inmediato de `#nav`, y `position: absolute;` en la propia `#nav`.
- 3) Como la posición absoluta de `#nav` lo colocará en el vértice superior izquierdo de su antepasado de manera predeterminada, ajustad los valores `left` y `top` (superior) de `#nav` según dicten las circunstancias.
- 4) Ajustad el valor `margin-top` o `padding-top` de `#sidebar` (según convenga) para reflejar la altura prevista de `#nav`. En los casos en los que `#nav` cambia de una página o sección a la siguiente, será necesario escribir algunas reglas, quizá con varios selectores en cada una; de aquí el consejo anterior de asignar una `id` referente del ámbito del contenido (y quizá también una `class`) en el `body` de cada documento de la web.

### 1.5. Implementación de la composición a tres columnas

Figura 4



Los elementos de una composición a tres columnas; observad los dos nuevos elementos contenedores y la diferente asignación de `id`.

Las principales diferencias de etiquetado que impone añadir una tercera columna son:

- Un contenedor incluye dos columnas adyacentes, generalmente la primera y la segunda.

- La tercera columna está situada en su propio contenedor.

Una vez establecido el etiquetado, la obtención de la disposición deseada es una cuestión de ordenar correctamente los valores `float`. No debéis olvidar que los contenedores no flotantes requieren un ajuste de márgenes para que la alineación sea correcta.

#### Nota

Debéis tener en cuenta que en relación con la figura 4, es mejor asignar a los elementos de la columna dual y del contenedor terciario `id` que ofrezcan alguna indicación del contexto, más que los `id` sugeridos en el propio diagrama.

Tabla 1

| Asignación flotante en cada columna en una composición de tres columnas. |                          |            |          |            |           |
|--|--------------------------|------------|----------|------------|-----------|
| Presentación deseada   | Contenido del contenedor | Contenedor | Primario | Secundario | Terciario |
| 2-1-3  | 1+2                      | left       | right    | none       | none      |
| 2-3-1  | 2+3                      | left       | none     | left       | none      |
| 3-1-2  | 1+2                      | right      | left     | none       | none      |
| 3-2-1  | 1+2                      | right      | left     | none       | none      |
| 1-2-3  | 1+2                      | left       | left     | none       | none      |
| 1-3-2  | 2+3                      | right      | none     | right      | none      |

Valores `float` de los cuatro elementos contenedores de una composición de tres columnas por orden de aparición de izquierda a derecha

La mayor dificultad de los diseños de tres columnas y su sencilla solución

La disposición a tres columnas más flexible introduce un elemento contenedor que no tiene significado semánticamente; la alternativa es imponer las convenciones de longitud del contenido u orden del código, que crearán una carga pesada, ya sea sobre los responsables de mantenimiento (en caso de requisitos de longitud de los contenidos), o sobre los visitantes (en caso de limitaciones del orden del código).

La introducción de este contenedor “sin significado” también puede plantear problemas cuando se trata de rediseñar una web. Imaginémonos la siguiente situación:

Cuando se diseña una web desde cero, se dispone con sus columnas en orden de presentación 2-3-1, pero más adelante se rediseña con sus columnas en un orden más tradicional como 2-1-3. El elemento contenedor de dos columnas se deberá mover de manera que comprenda diferentes contenedores de columnas. Y entonces, ¿qué?

En este caso, el resultado es fácil de obtener; si el sitio funciona realmente con plantillas, entonces hay (probablemente) pocos archivos que haya que cambiar. Si, en cambio, todos los documentos de la web utilizan el mismo esquema de etiquetado, se deberá utilizar buscar y sustituir. De todos modos, esto no será difícil.

La disposición de etiquetado global de los contenedores tomará una de las dos formas siguientes:

- `<div id="#container"><div id="primary">...</div><div id="secondary">...</div></div><div id="tertiary">...</div>`
- `<div id="primary">...</div><div id="#container"><div id="secondary">...</div><div id="tertiary">...</div></div>`

En estos dos ejemplos, los fragmentos correspondientes que se deben buscar y sustituir se muestran en negrita. Algunas de estas `id` serán únicas en vuestro documento, y como la ubicación de las etiquetas de cierre doble se puede prever (ya sea al lado de una tercera etiqueta de cierre o de `#tertiary`), escribir una operación de búsqueda y sustitución para modificarlas es un cambio relativamente sencillo.

## 1.6. Visión amplia de los encabezamientos y pies

Hasta ahora hemos hablado de qué debería ir en los encabezamientos y pies: logo/logotipo, búsqueda en toda la web, enlaces con información de cuentas de usuario, metadatos de la web, etc., pero la efectividad y el atractivo del encabezamiento/pie se ha tratado poco fuera de los artículos del currículo sobre esqueletos y compuestos/bocetos.

Como no sería positivo agobiarlos con un discurso larguísimo en defensa de este hecho, os proponemos buscar tres webs que sean notables, ya sea por su popularidad o por la importancia de sus editores, y echar un vistazo a sus particularidades de diseño.

### 1.6.1. Sitio personal: Cindy Li

Figura 5



En este caso, hay dos asuntos dignos de ser comentados, la identidad y el contraste.

## Identidad

*Identidad* es un término que adquiere un significado especial en el reino de la publicidad y el marketing, donde hace referencia a las marcas comerciales y otros elementos de diseño que son particulares de la presentación de una empresa comercial y sus productos. La web de Cindy Li\* lo lleva a cabo a un nivel personal mostrando una caricatura de su perfil facial en el encabezamiento del sitio y utilizando un tipo de letra distintivo para definir el título de la web y la navegación principal.

\* <http://www.cindyli.com/>

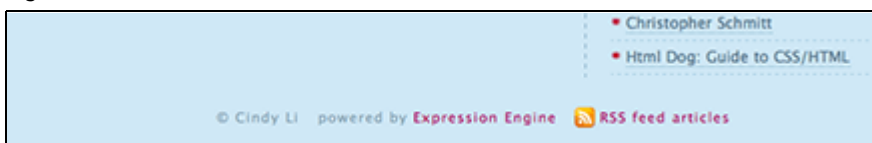
El modo como se impone la identidad en las webs de empresa se tratará con más detalle a continuación.

## Contraste

En cindyli.com, el visitante puede ver a primera vista de qué va la cosa: la identidad, el lienzo de contenidos y el contenido primario están todos colocados en huellas discretas gracias a sus colores de fondo. Además, el título de la web y la navegación tienen un contraste alto en relación con el fondo.

Y en cuanto al pie de la web:

Figura 6



El pie de Cindy Li es algo escaso, en él aparece la declaración de *copyright*, un enlace a la web de marketing de la plataforma publicitaria que utiliza (lo que probablemente es obligatorio bajo las condiciones de la licencia de uso), y un enlace a una suscripción sindicada de los artículos que publica.

Al contrario que las demás webs presentadas en este artículo, la web de Cindy no ofrece una búsqueda de texto completo, probablemente por motivos técnicos. No obstante, como la web es un blog, su diseño permite asumir que la mayoría de los lectores limitan su interés en los nuevos contenidos.

### 1.6.2. Redes sociales: Facebook

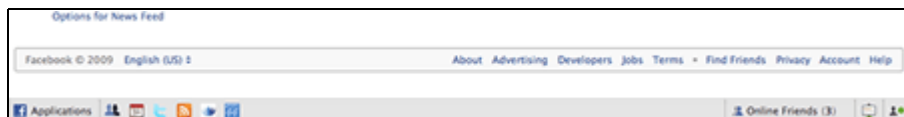
Figura 7



Facebook, como muchos destinos de redes sociales, refuerza su identidad mediante el uso de la composición y el color, ya que la propia aplicación es el destino.

Igual que muchas webs comerciales, Facebook ofrece tanto una búsqueda con texto completo como un enfoque categorizado en la navegación de la web.

Figura 8



Igual que su encabezamiento, el pie de Facebook también es pequeño, incluso si tenemos en cuenta el “widget” de aplicación persistente en la parte más inferior. Lo que tiene interés en este caso se encuentra al lado de la declaración de *copyright*: un artificio para cambiar el idioma predeterminado del usuario.

Otra práctica común ejemplificada por el pie de Facebook es que mientras los enlaces de uso en la búsqueda de usuario van en el encabezamiento, el pie contiene todos los enlaces sobre el operador de la web y de la propia web.

### 1.6.3. Marketing de empresa y servicio al cliente: BNSF Railway

Figura 9



Como la mayoría de las webs “de empresa”, el contraste más alto de las características del encabezamiento se disfruta mediante el logotipo del editor de la web... y el único tono presente es el mismo que se utiliza en el logotipo.

#### Repaso de la identidad

En cuanto a los diferentes logos, logotipos y otros recursos de diseño que refuerzan la identidad comercial, debéis tener en cuenta los siguientes, que son marcas omnipresentes que hacen negocios con Opera Software:

- Nokia



- Samsung Group y Samsung Electronics



- Google



- IBM



Aparte de estos cuatro, hay otros ejemplos similares de diseño gráfico muy conocidos: el símbolo de Nike, el “Swoosh” (de hecho, este nombre está registrado), el logotipo de “la estrella de la muerte” de AT&T, el logotipo de FedEx (y la flecha en el espacio blanco contenido por sus dos últimas letras), y el “marrrón de UPS” son ejemplos de identidad corporativa que el público general conoce casi universalmente (al menos en Estados Unidos).

Cualquier operador que se tome la molestia de desarrollar una identidad visual distintiva hará bien en integrar esta identidad en el diseño de la web porque tiene un fuerte impacto en la naturaleza del trabajo del especialista en estilos.

Y en cuanto a otros elementos del encabezamiento de BNSF aparte del logotipo, el más notable es el uso de dos modos de búsqueda (como en el caso de Facebook).

Figura 10



La web de BNSF tiene la disposición de pie más "tradicional" de estos tres sitios, donde se hace el mínimo esfuerzo por colocar la navegación secundaria en un plano visual distintivo con reglas horizontales o un color de fondo diferente.

#### 1.6.4. Diseño de encabezamiento y pie: los detalles de bajo nivel

Al adoptar una visión más amplia del diseño del encabezamiento y el pie, quedan claras los siguientes elementos comunes:

- **Los destinos de empresas y aplicaciones tienden a disponer su navegación principal en una fila a lo largo del margen superior del lienzo del navegador.** Este hecho los distingue de las webs de noticias y comercio electrónico, que a menudo utilizan un planteamiento más columnar para la composición de la navegación.
- **Igual que la navegación principal orientada horizontalmente es habitual, también lo es el uso de una navegación secundaria discreta en el pie.** Además, la división de destino y metainformación de la web descrita anteriormente en este artículo se impone bastante a menudo.
- **Donde hay presente una búsqueda de texto completo, los campos de introducción tienden a colocarse cerca del margen derecho del encabezamiento.** Esto es así sobre todo porque la búsqueda de texto completo bien implementada es una manera tan válida de navegar por el sitio como los enlaces tradicionales alimentados por una taxonomía, y es probable que un subconjunto de la población de la web abuse de ello, probablemente los usuarios con menos conocimientos técnicos.

#### Más información sobre la implementación de la composición de navegación

Antes de pasar a los detalles, deberíais consultar List-O-Matic\*, una aplicación alojada en Accessify.com que crea elementos de navegación con un estilo sencillo, preparados para ser insertados en cualquier composición de página.

\* <http://www.accessify.com/tools-and-wizards/developer-tools/list-o-matic/>

Aparte del tema de las herramientas que hacen el trabajo por vosotros, hay dos enfoques básicos en la composición de la navegación:

La navegación se integra en el encabezamiento de la web (visualmente, si no literalmente) y se orienta horizontalmente. En este caso, el valor `dis-`

`play` (visualizar) de los enlaces individuales probablemente se cambiará a `block`, y a los elementos de su lista de contenidos se les asignará un valor `float de left`.

La navegación está orientada verticalmente y está colocada a la izquierda del contenido principal, ya sea en su propia columna o justo encima del contenido no primario. En este caso, lo más probable es que veáis que se utiliza algún tipo de posicionamiento no `static`.

### 1.7. Webs con diferentes números de columnas: hacer trampa con `class` y `display`

Algunas webs pueden tener páginas con una o dos columnas; otras, dos o tres. La flexibilidad es uno de los puntos fuertes de CSS, y uno de los recursos de los que los diseñadores gráficos tiran inconscientemente, en su libidionosa cruzada por obtener un control férreo sobre la experiencia del usuario.

Normalmente, estos casos se tratan parcialmente con *includes*: scripts del sitio que permiten añadir fragmentos persistentes a una página de manera programática, en vez de copiarlos repetidamente.

Sin embargo, incluso con esas inclusiones los especialistas en estilos todavía encontrarán diferencias en la disposición; entonces, ¿cuál es la mejor manera de tratarlas?

El planteamiento más directo es añadir una `class` al `body` de cualquier página que la pueda necesitar. Esto podría asumir una naturaleza ordinal que corresponde a algunas series de disposiciones sugeridas en las directrices de identidad, o tener que ver de nuevo con el contenido y el resultado en varias reglas selectoras, como las siguientes:

```
.about #bodyCopy,  
.contact #bodyCopy,  
.privacy #bodyCopy {  
  
    float: none; width: auto;  
}  
  
.about #sidebar,  
.contact #sidebar,  
.privacy #sidebar {  
  
    display: none;  
}
```



El único inconveniente de seguir este planteamiento sin la ventaja de incluir declaraciones (una manera pobre de hacer que el contenido desaparezca o reaparezca, si queréis) es que las políticas de los operadores de los motores de búsqueda podrían reducir el peso de aquellas páginas en sus resultados o, en el caso de implementaciones espectacularmente chapuceras, incluso sacar de sus listas las webs que las utilizan. Por este motivo (entre muchos otros que se deben tener en cuenta), cualquier tipo de alojamiento que contratéis debería aceptar algún tipo de función de inclusión.

## 1.8. Resumen

Aunque es tentador –sobre todo si sois principiantes– sentarse directamente y empezar a escribir etiquetado y código, este proceso no da como resultado webs especialmente atractivas, útiles o mantenibles.

No obstante, teniendo en cuenta esmeradamente el contenido que irá en una web y el modo como se debería disponer, podéis lanzar cualquier web a la estructura derivada de sus requisitos.

Plantillas básicas:

- una columna

**Archivo fuente:** “Composición a una columna”

[http://dev.opera.com/articles/view/38-headers-footers-columns-templates/template\\_1col.html.txt](http://dev.opera.com/articles/view/38-headers-footers-columns-templates/template_1col.html.txt)

- dos columnas

**Archivo fuente:** “Composición a dos columnas”

[http://dev.opera.com/articles/view/38-headers-footers-columns-templates/template\\_2col.html.txt](http://dev.opera.com/articles/view/38-headers-footers-columns-templates/template_2col.html.txt)

- tres columnas

**Archivo fuente:** “Composición a tres columnas”

[http://dev.opera.com/articles/view/38-headers-footers-columns-templates/template\\_3col.html.txt](http://dev.opera.com/articles/view/38-headers-footers-columns-templates/template_3col.html.txt)

## Preguntas de repaso

1) Dada una lista de posibles enlaces proporcionados por vuestro instructor, divididlas en enlaces primarios (encabezamientos) y secundarios (pies). Justificad vuestras asignaciones basándoos en los ejemplos facilitados en este artículo.

2) Identificad lo siguiente en un compuesto creado por un compañero:

- a) el número de columnas que se deben aplicar al diseño;

- b) las anchuras de estas columnas,
- c) el `float/width/margin`, y
- d) la combinación `float/width/margin`, si la hay, que se debería utilizar para implementar la presentación de estas columnas.

3) Con un logotipo, una lista de requisitos y una arquitectura determinados, diseñad un encabezamiento de web.

4) Si no podéis demostrar el uso de la sección áurea de la composición del encabezamiento diseñado en el ejercicio anterior, modificad la composición de manera apropiada y evaluad subjetivamente lo atractivo que es el resultado.

5) Basándoos sólo en los resultados de los motores de búsqueda, obtened el motivo por el que las listas son preferibles a las colecciones de `div` (o de otros elementos) para estructurar los elementos de navegación. Haced referencia a las características del software de lectura de pantallas en vuestra respuesta.

6) Revisad, de memoria, uno de los archivos de la plantilla proporcionada de manera que admita un número diferente de columnas. Modificad también de manera significativa la composición de la navegación principal, comparada con lo que se encontró en el archivo de plantilla original.

## Lecturas complementarias

**BNSF Railroad.** 2006. [Fecha de consulta: 13 de enero del 2009].  
<http://www.bnsf.com/>

**Facebook.** 2008. [Fecha de consulta: 13 de enero del 2009].  
<http://www.facebook.com/home.php>

**Henick, Ben.** 2006. "Avoid edge cases by designing up front". *A List Apart*. [Fecha de consulta: 13 de enero del 2009].  
<http://www.alistapart.com/articles/avoidedgecases>

**Li, Cindy** (2008). "The Adventures of Cindy Li". [Fecha de consulta: 13 de enero del 2009].  
<http://www.cindyli.com/>

**Morville, Peter; Rosenfeld, Louis** (2006). *Information architecture for the world wide web* (3.<sup>a</sup> edición). Cambridge: O'Reilly Media.