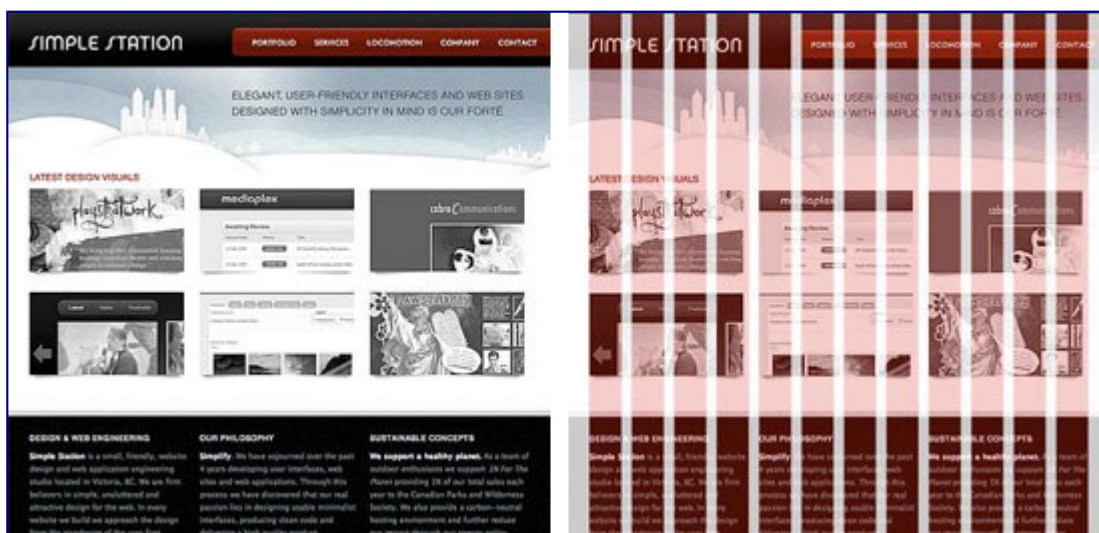


Un **sistema en grid** no es más que un **archivo CSS** que define una serie de estilos que te ayudan a **organizar el contenido de una página web**. Podemos pensar en ello como en una **plantilla o una hoja cuadriculada**. Las rejillas permiten la **estabilidad y estructuración** en un **diseño de web**, dando al diseñador una lógica para la construcción de la estética del sitio.

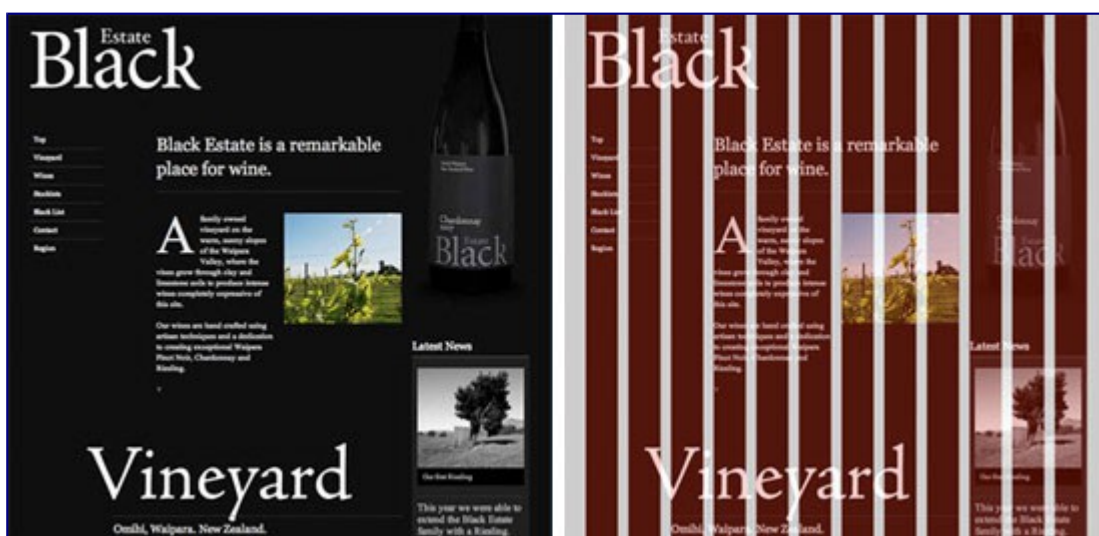
Lo básico del diseño web en grid

Los **grids** se suelen encontrar en los trabajos de impresión, pero son muy aplicables al **diseño web**. Una rejilla puede ayudar en la creación de una **estructura uniforme para iniciar un diseño web**.

Una **rejilla (grid)** es la división de un **diseño** mediante **guías verticales y horizontales** para incorporar márgenes, espacios y columnas con el fin de proporcionar un **marco para organizar el contenido**. Este marco o cuadrícula puede ser “roto” en algunos de los elementos del diseño para ayudar a crear una sensación más orgánica y fluida.



Diseño en grid de simplestation.com



Diseño en grid de blackestate

Por dónde empezar en un diseño web en grid

Para arrancar un **diseño en grid** podemos crear nuestra propia rejilla o recurrir a uno de los múltiples sistemas o [frameworks existentes](#).

960s grid sytem <http://960.gs/>

<http://unsemantic.com/>

The golden grid.

Etc

O con este generador <http://www.gridsystemgenerator.com/>

aquí tenéis algunas herramientas fáciles de utilizar para comenzar un **diseño web en grid**:

[Grid Designer](#)






The screenshot shows the 'Grid Designer' web application. It has a header with the title 'Grid Designer' and a sub-header 'as featured in Creative Review magazine!'. Below the header is a navigation bar with links: '» Columns', '» Typography', '» Export', and '» Help'. The main area contains four rows of input fields and buttons. The first row is for 'Columns' with a value of 4 and buttons for 'add', 'subtract', 'multiply', and 'divide'. The second row is for 'Pixels' with a value of 800 and buttons for 'widen' and 'narrow'. The third row is for 'Gutters' with a value of 20 and buttons for 'increase', 'decrease', 'expand', and 'collapse'. The fourth row is for 'Margins' with a value of 22 and buttons for 'increase', 'decrease', 'expand', and 'collapse'. A 'design' button is at the bottom left.

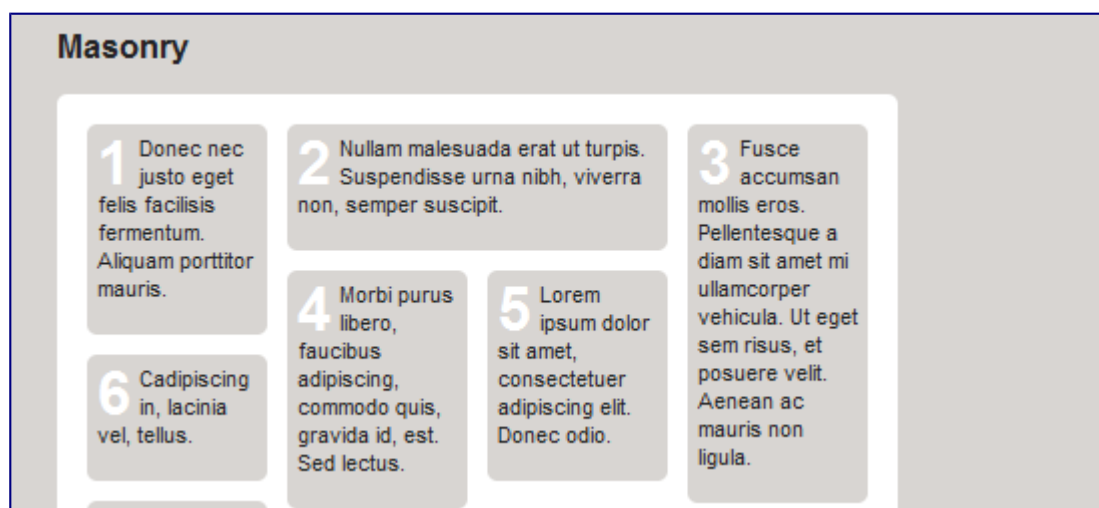
[Grid System Generator](#)

GRID SYSTEM GENERATOR			
324539 grid systems generated			
Grid system	Width	No. of columns	Margin (left)
960.gs	960	12	
Golden grid	970	12	
1kb grid	960	12	

Grid Calculator

Base font size		<input type="text" value="12"/> px
Number of columns		<input type="text" value="7"/> columns
Column width		<input type="text" value="144"/> px

jQuery Masonry



El sistema 960 Grid



960 Grid System

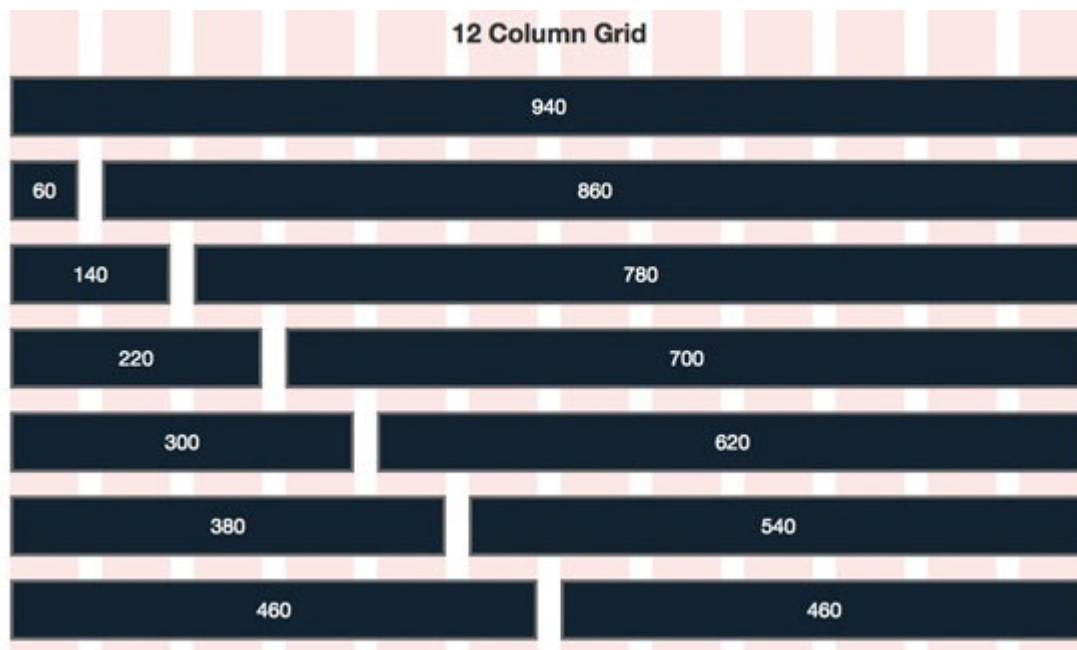
En este apartado nos centraremos en el [960 Grid System](#). La mayoría de los **sistemas de rejillas** que hay por ahí son similares pero, desde nuestro punto de vista, este es uno de los más completos.

Cualquiera de ellos, y el 960 GS en particular, nos aportan la compatibilidad entre navegadores y la optimización de los anchos de columna que vienen predefinidos.

El sistema de cuadrícula 960 es simplemente una manera de diseñar sitios web usando una rejilla de 960 píxeles de ancho. La razón es 960 píxeles de ancho se debe a que el número 960 simplifica las divisiones en columnas y los márgenes de la plantilla de diseño.

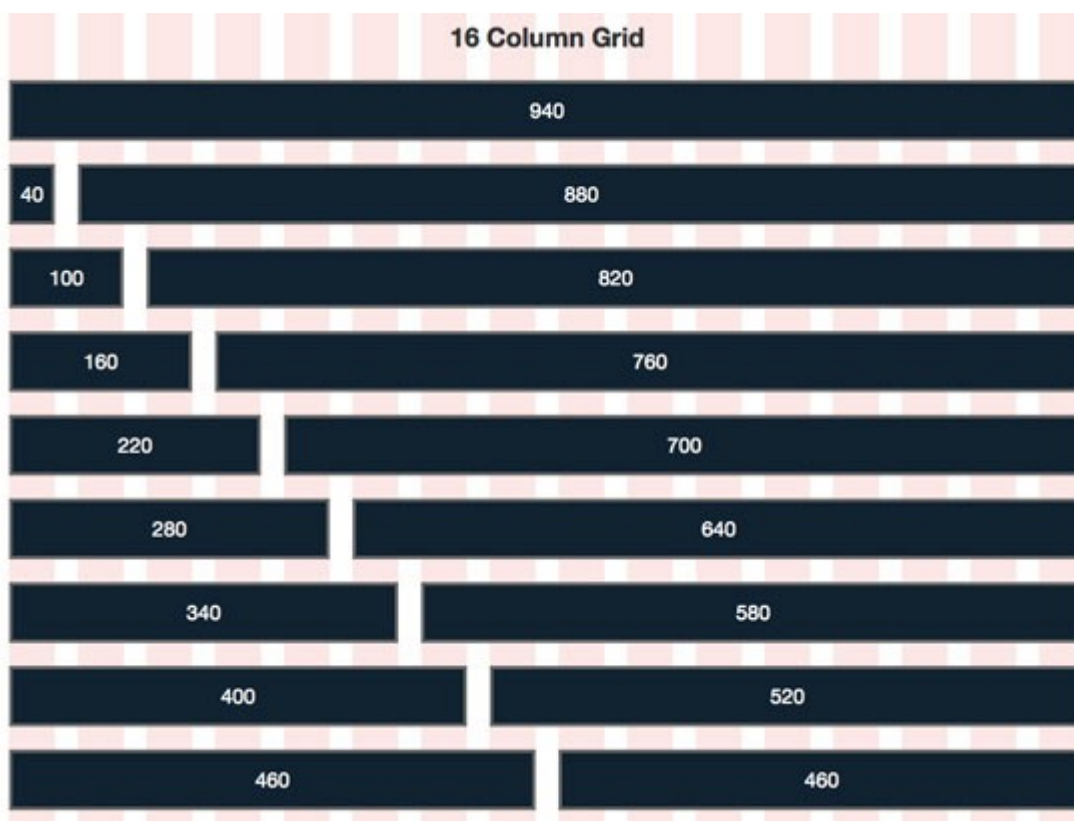
El 960 GS viene en dos variantes principales: una cuadrícula de 12 columnas y una cuadrícula de 16 columnas. También incluye una versión de 24 columnas para diseños web en los que necesitemos incluir una mayor densidad de contenidos.

En la versión de 12 columnas, la columna más estrecha es de 60 píxeles de ancho, incrementándose después en 80 píxeles. De esta forma, los anchos de las columnas disponibles son: 60, 140, 220, 300, 380, 460, 620, 700, 780, 860 y 940.



960 Grid System: disposición en 12 columnas

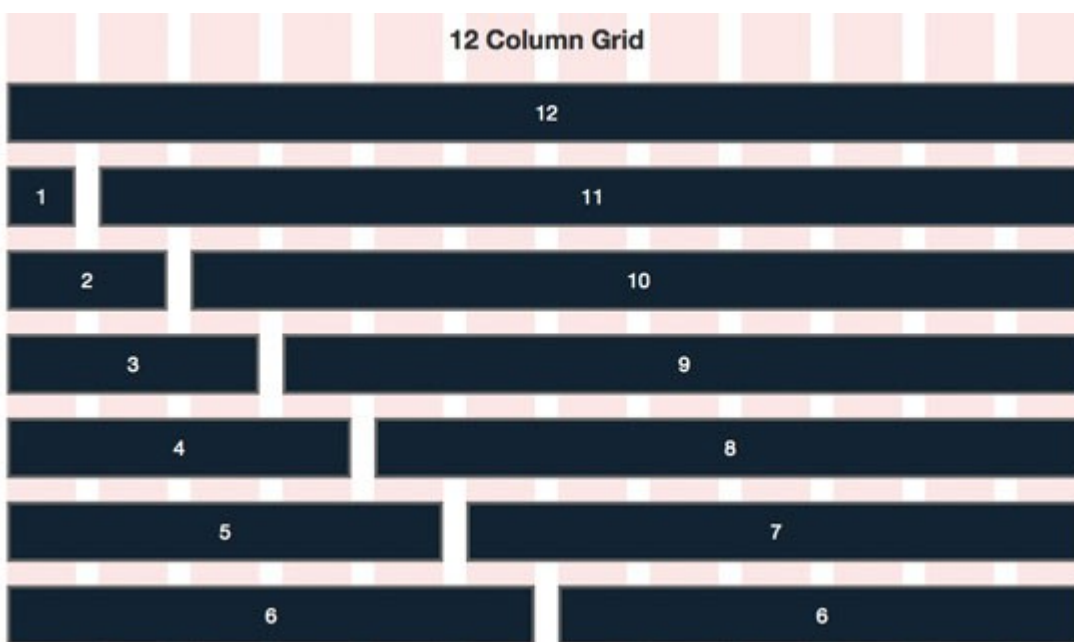
Del mismo modo, en la versión en 16 columnas, la columna más estrecha es de 40 píxeles de ancho y después se incrementa en 60 píxeles. Así que el ancho de las columnas disponibles son: 40, 100, 160, 220, 280, 340, 400, 460, 520, 580, 640, 700, 760, 820, 880 y 940.



960 Grid System: disposición en 16 columnas

Cada una de las barras horizontales podemos relacionarla con una clase CSS en el sistema 960 Grid, de forma que le asignaremos una capa (div) con el ancho adecuado.

La nomenclatura de cada uno de los elementos del CSS será acorde con con el ancho del elemento de la rejilla, de forma que la clase CSS `grid_1` representará a la columna más estrecha y la clase `grid_12` será la que aplicaremos a las columna más ancha para un sistema de 12 columnas.



Nomenclatura de las clases en 960 Grid System

Este sistema de nombres hace que sea increíblemente fácil la utilización de cada elemento, por ejemplo, si utilizamos el sistema de 12 columnas, simplemente tendremos que utilizar la clase `grid_4` para incluir uniformemente tres capas de texto que ocupen el diseño de lado a lado:



Uso de 960 GS para crear un layout de tres columnas

Reposicionando elementos en la rejilla

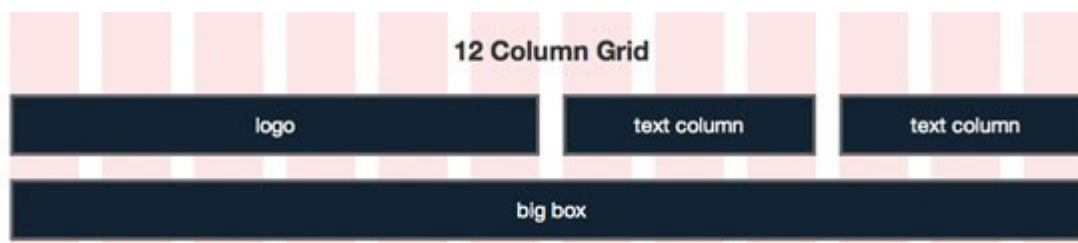
El sistema de rejilla 960 permite reposicionar los elementos de la plantilla de forma independiente. Esto se consigue “empujando (push)” o “tirando (pull)” de ellos, incluyendo ciertas características en las clases CSS. Por ejemplo:



Reposicionar elementos en el sistema de rejilla

En la primera imagen tenemos una layout básico de 4 columnas (clase `grid_3`) en un sistema grid de 12.

En la segunda imagen, se aplican cambios de posición a la primer y última columna mediante unas variantes de la clase `grid_3` (`push_1` y `pull_1`). El alcance de esta funcionalidad se comprende mejor en este ejemplo:



Ejemplo de distribución sin reposicionamiento

Aplicando este código:

```
<div class="grid_6 push_3">
  <strong><p>logo</p></strong>
```



```

</div>

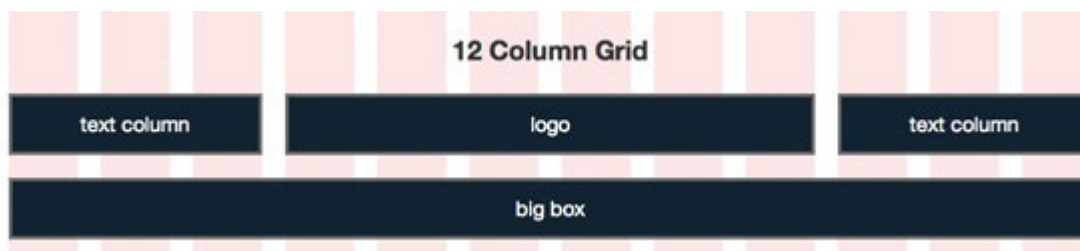
<div class="grid_3 pull_6">
    <strong><p>text column</p></strong>
</div>

<div class="grid_3">
    <strong><p>text column</p></strong>
</div>

<div class="grid_12">
    <strong><p>big box</p></strong>
</div>

```

Conseguimos reposicionar el logo en la plantilla:

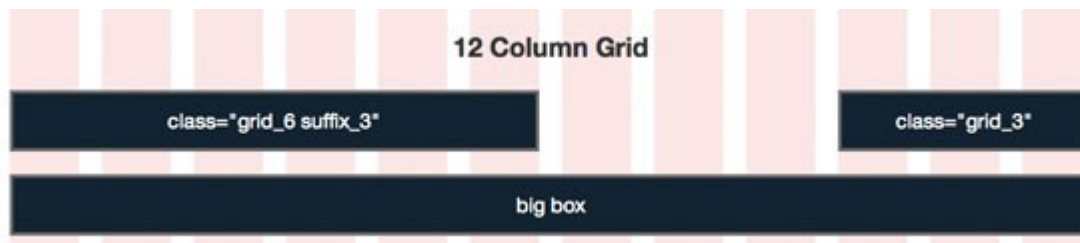


Reposicionando elementos en la rejilla

Incluyendo espacios vacios en el diseño

Ya hemos hablado en otros artículos del uso de espacios vacios en diseño web ([“Diseño web minimalista: principios, conceptos y ejemplos”](#)). Los amplios espacios vacios alrededor de los elementos generan una percepción ordenada de los conceptos esenciales del sitio web.

El sistema grid 960 facilita mucho la creación de zonas vacías alrededor de los elementos sobre los cuales queremos llamar la atención del usuario. Simplemente tenemos que utilizar, en nuestro elementos div, las clases `prefix` y `suffix`, cuyo uso es muy similar al de las clases `push` y `pull`. Por ejemplo, para incluir un espacio vacío con el ancho de una columna antes de un elemento, utilizaríamos la clase `prefix_1 class`, si lo quisiéramos incluir después del elemento utilizaríamos la clase `suffix_1`. En este ejemplo se incluye un espacio en blanco de tres columnas después de un elemento `grid_6`:



Incluir zonas vacias en la rejilla

CSS reset

El sistema grid 960 incluye también un [CSS reset](#). Mediante este recurso, se permite restablecer el estilo de todos y cada uno de los elementos de nuestro diseño a sus valores de referencia.

Se trata de un recurso opcional que aporta el sistema y que, en caso de no necesitarlo, simplemente se puede no utilizar.

Actualización: [960 grid template \(enlace a sitio de descarga de documentos en PDF con wireframes para diseños web con el sistema de rejilla 960\)](#)

Existen multitud de opciones más: <http://www.cssauthor.com/grid-systems/>