

## Unidad 4. Usabilidad y diseño responsive

### Contenido

1	Usabilidad	2
1.1	Consejos para hacer una web usable	2
2	Principios usabilidad	4
3	Diseño responsive	5
3.1	Longitudes relativas	5
3.2	Media query	6
3.2.1	Tipos de medios	7
1.1	Principales expresiones a utilizar en las media queries	8
3.3	Configurar ventanas gráficas	9
3.3.1	Tamaño de contenido a la ventana gráfica	9
3.4	¿Qué es una vista de cuadrícula?	10
4	Tendencias de diseño. Mobile First	11

# 1 Usabilidad

La usabilidad se puede definir como **la facilidad con que las personas pueden utilizar una herramienta particular o cualquier otro objeto fabricado por humanos con el fin de alcanzar un objetivo concreto.**

Podemos observar que la usabilidad se compone de dos tipos de atributos:

- **Atributos cuantificables de forma objetiva:** como son la eficacia o número de errores cometidos por el usuario durante la realización de una tarea, y eficiencia o tiempo empleado por el usuario para la consecución de una tarea.
- **Atributos cuantificables de forma subjetiva:** como es la satisfacción de uso, medible a través de la interrogación al usuario.

Los usuarios no buscan usabilidad, buscan utilidad, entendida como el **provecho, beneficio e interés que produce su uso.**

Un producto será usable en la medida en que el **beneficio que se obtenga de usarlo justifique el esfuerzo necesario para su uso.** Pensemos por ejemplo en los **automóviles:** son herramientas que requieren de un gran esfuerzo de aprendizaje y adaptación por parte de sus usuarios, aunque no por ello son percibidos como artefactos poco usables.

La usabilidad **no sólo es relativa a sus usuarios, sino también a los usos.** Por ejemplo, un **microondas** puede ser usado para guardar zapatos, pero no se pensó para eso, por lo tanto, puede resultar no usable en ese contexto.

Además, **los objetivos y contextos previstos determinarán la importancia de su usabilidad.** Por ejemplo, un **teléfono móvil** puede ser usado para realizar una llamada de emergencia, o para escuchar las diferentes melodías que incorpora, un objetivo este último para el que la usabilidad puede que no sea tan relevante como para el primero.

## 1.1 Consejos para hacer una web usable

### Conocer al usuario

Es importante saber cómo afectan los elementos visuales de la pantalla al usuario, cómo piensa, cómo actúa, etc.

### Limitar posibilidades

Siempre hay que tener en cuenta que cuantas menos opciones, menos posibilidades de error tendrá el usuario.

Caduca final:

Caduca final:  /

### Orientar al usuario

Ya sea mostrándole paulatinamente las distintas opciones, o mostrándoles pequeños mensajes informativos.



### Solicitar confirmación

Antes de realizar una acción que no tiene vuelta atrás, hay que informar al usuario.

Parece que te has olvidado de adjuntar un archivo.

En tu mensaje has escrito "adjunto", pero no lleva ningún archivo adjunto. ¿Quieres enviarlo igualmente?

Cancelar Aceptar

## Evitar la pérdida de información

En caso de error y vuelta atrás, el usuario no debería volver a introducir los datos.

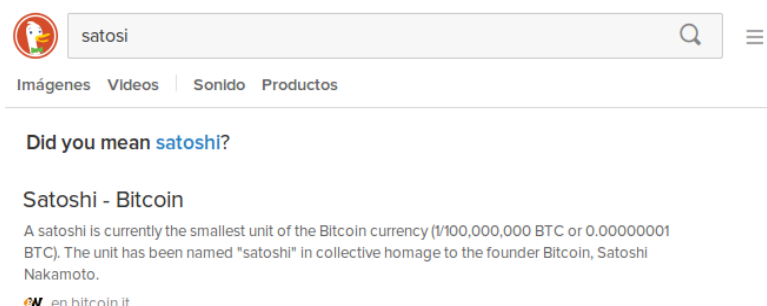
## Permitir deshacer

Siempre que sea posible, hay que ofrecer al usuario la posibilidad de volver atrás.



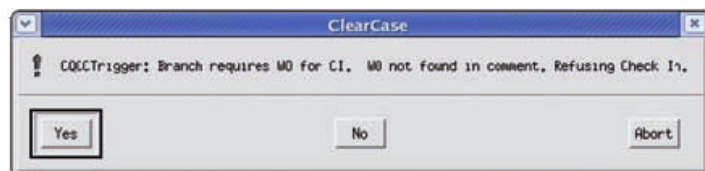
## Ofrecer solución automática a los errores

No siempre es posible, pero si es deseable que el propio sistema sea capaz de ofrecer soluciones automáticas.



## Mensajes de error para humanos

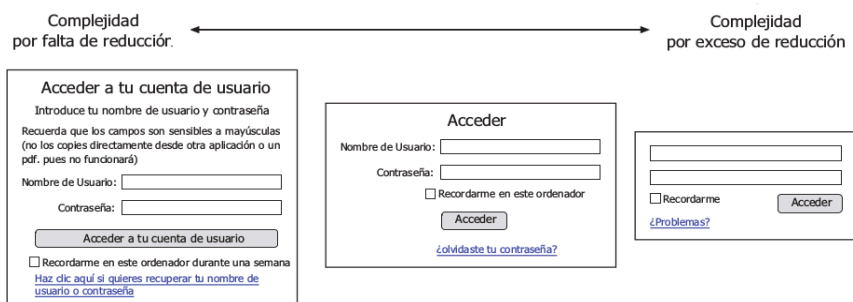
Si aparece un mensaje de error este tiene que ser entendible por el usuario.



En lo referente a la Simplicidad

## Reducción

Si prescindimos de un elemento, y el cambio no afecta a la comprensión del usuario, entonces su presencia no era necesaria.

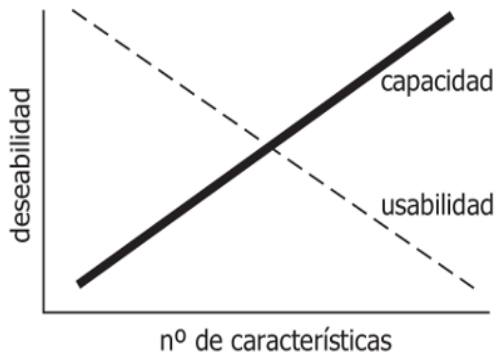


## Espacios vacíos

Al separar elementos con espacios vacíos, guiamos la atención hacia aquello que permanece y es relevante.

## Equilibrio de características

Al incrementar el número de características se observa un efecto positivo sobre las capacidades percibidas, pero un efecto negativo sobre la usabilidad percibida.



## Organizar, clasificar y ordenar

Organizar, clasificar y ordenar **son también formas de simplificar**. Por ejemplo, Google no simplifica los resultados de búsqueda por clasificación, sino por ordenación.

## 2 Crear un sitio Web usable

Un nuevo **sitio Web será usable** o podremos mejorar su usabilidad si:

- Aplicamos en su diseño los 10 principios de Jakob Nielsen también conocidos como los 10 heurísticos de Jakob Nielsen.
- Sometiendo el diseño a baterías de pruebas con usuarios representativos a quienes solicitaremos que lleven a cabo las tareas más frecuentes del sitio, analizando los problemas y fricciones que se detecten en el uso cotidiano del sitio y mejorándolo.

### 1.1 Principios de Jakob Nielsen

#### • Visibilidad estado

El sistema (o sitio web) siempre debe informar al usuario acerca de lo que está sucediendo, como, por ejemplo, cuando en una interfaz tipo webmail se adjuntan ficheros a un mensaje, el sistema debe informar del hecho mostrando un mensaje de espera.

#### • Adecuación al mundo real

El sistema debe hablar el **lenguaje del usuario**, huyendo de tecnicismos incomprensibles o mensajes crípticos.

#### • Libertad y control

El usuario debe tener el control del sistema, ser él el que decida.

Se debe ofrecer siempre una forma de "**salida de emergencia**", como por ejemplo la opción para "saltar" animaciones de introducción

#### • Consistencia y estándares

Consistencia en el **estilo** (enlaces iguales, pestañas iguales, etc.)

Seguir **estándares de diseño** ampliamente aceptados:

- usar iconos conocidos y utilizados

- icono de la empresa con enlace a la home arriba a la izquierda
- menú en la parte superior
- opciones a la izquierda o a la derecha
- pie de página con un resumen del mapa de la web
- **Prevención de errores**

Mejor que un buen mensaje de error es un **diseño que prevenga** que ocurra el error.

- **Reconocimiento antes que recuerdo**

Hacer **visibles objetos, acciones y opciones** para que el usuario no tenga por qué recordar información entre distintas secciones o partes del sitio web o aplicación. El usuario **no tiene por qué recordar** dónde se encontraba cierta información, o cómo se llegaba a determinada página.

- **Flexibilidad y eficiencia**

El sitio debe ser fácil de usar para usuarios inexpertos. Pero también proporcionar atajos o aceleradores para usuarios avanzados.

- **Diseño estético y minimalista**

Las páginas **no** deben contener información irrelevante o innecesaria. Cada información extra compite con la información relevante y disminuye su visibilidad.

- **Manejo de errores**

Los mensajes de error deben **expresar claramente cuál ha sido la causa** del problema. También deben **sugerir las posibles alternativas o soluciones**, como por ejemplo mensajes del tipo "Usted quiso decir...".

Además, se debe **guardar el contenido** introducido por el usuario para que no tenga que volver introducirlo y pueda subsanar el error.

- **Ayuda y documentación**

Aunque es mejor que un sitio web se pueda utilizar sin necesidad de ayuda o documentación, **en sitios web extensos o en procesos de interacción complejos se debe proporcionar información de ayuda** al usuario.

Video: <https://openwebinars.net/academia/aprende/introduccion-ux/11733/>

## 1.2 El análisis heurístico

El **análisis heurístico** lo realiza un consultor externo o el mismo director del proyecto, comprobando si el sitio Web o el prototipo propuesto se adhiere a **los principios de buenas prácticas** en el diseño de interfaces de usuario:

El análisis heurístico se suele realizar por más de un consultor quienes evalúan cada uno de estos puntos. Posteriormente, la puntuación final es la media de la nota dada por cada consultor. La participación de varios consultores está justificada por la implicación subjetiva inherente a la valoración que tienen que hacer.



Video: <https://openwebinars.net/academia/aprende/introduccion-ux/11745/>

### 1.3 Evaluación con usuarios

El **test con usuarios** es una prueba de usabilidad que se basa en la observación y análisis de cómo un grupo de usuarios reales de perfil similar al de nuestros clientes potenciales interactúa con nuestro diseño.

Los **procedimientos científicos de análisis del comportamiento** del usuario durante su navegación por el sitio Web como el EyeTracking, la electroencefalografía (EEG), la respuesta galvánica o el análisis facial, etc.

**Grabaciones de sesiones** como Hotjar que es una de las más utilizadas, con ella accedemos a la grabación completa de todo lo que un visitante ha hecho en el sitio Web, desde que accedió al mismo hasta que lo dejó.

Entre otras muchas cosas, Hotjar muestra mapas de calor que representan las áreas de la pantalla por donde desplazó el puntero, cuándo y hasta qué punto usó el desplazamiento vertical para ver el resto de una página, cómo introdujo datos en los formularios, qué errores cometió y cómo interactuó con el sistema.

Videos: <https://openwebinars.net/academia/aprende/introduccion-ux/11746/>

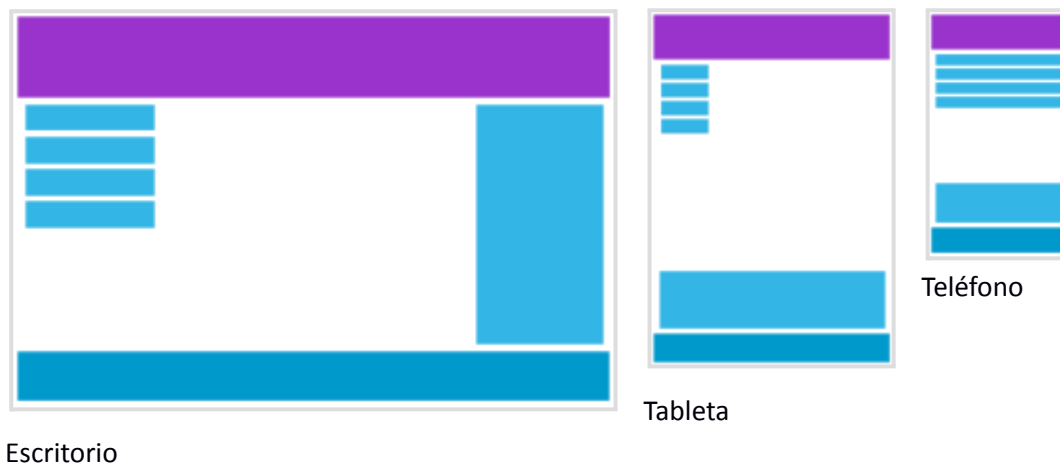
<https://openwebinars.net/academia/aprende/introduccion-ux/11748/>

## 3 Diseño responsive

Las páginas web pueden verse usando muchos dispositivos diferentes: ordenadores de escritorio, tabletas y teléfonos. Su página web debe verse bien y ser fácil de usar, independientemente del dispositivo.

El **diseño web receptivo** trata de crear sitios web que se ajustan automáticamente para verse bien en todos los dispositivos, desde teléfonos pequeños hasta grandes escritorios.

Las páginas web no deben omitir la información para adaptarse a dispositivos más pequeños, sino adaptar su contenido para adaptarse a cualquier dispositivo:



Se denomina diseño web receptivo cuando utiliza CSS y HTML para cambiar el tamaño, ocultar, reducir, ampliar o mover el contenido para que se vea bien en cualquier pantalla.

### 3.1 Longitudes relativas

Recuerda que las unidades de longitud relativa especifican una longitud relativa a otra propiedad de longitud. Las unidades de longitud relativa escalan mejor entre diferentes medios de representación.

Unit	Description
em	Relative to the font-size of the element (2em means 2 times the size of the current font)
ex	Relative to the x-height of the current font (rarely used)
ch	Relative to width of the "0" (zero)
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimensión*
vmax	Relative to 1% of viewport's* larger dimensión*
%	

Las unidades em y rem son prácticas para crear un diseño perfectamente escalable

\* Viewport = el tamaño de la ventana del navegador. Si la ventana gráfica tiene 50 cm de ancho, 1vw = 0.5 cm.

### 3.2 Media query

Una media query consiste en un tipo de medio y al menos una consulta que limita las hojas de estilo utilizando características del medio como ancho, alto y color. Se entiende como un módulo CSS3 que permite adaptar la representación del contenido a características del dispositivo.

Añadido en CSS3, las media queries dejan que la presentación del contenido se adapte a un rango específico de dispositivos de salida sin tener que cambiar el contenido en sí.

Estas consultas se pueden usar para verificar muchas cosas, tales como:

- ancho y alto de la ventana gráfica
- ancho y alto del dispositivo
- orientación (¿la tableta / teléfono está en modo horizontal o vertical?)
- resolución

Una consulta de medios consiste en un tipo de medio y puede contener una o más expresiones, que se resuelven en verdadero o falso.

**@media not|only mediatype and (expressions) {**

**CSS-Code;**

**}**

Ejemplo

Cuando el ancho del navegador esté entre 520 y 699px, aplicaremos un ícono de correo electrónico a cada enlace de correo electrónico y para anchos de navegador superiores a 1151px.

**@media screen and (max-width: 699px) and (min-width: 520px), (min-width: 1151px) {**

**ul li a {**

**padding-left: 30px;**

**background: url(email-icon.png) left center no-repeat;**

**}**

**}**

También se puede tener diferentes hojas de estilo para diferentes medios:

`<link rel="stylesheet" media="mediatype and|not|only (expressions)" href="print.css">`

### 3.2.1 Tipos de medios

Medio	Descripción
<b>all</b>	Todos los medios definidos
<b>braille</b>	Dispositivos táctiles que emplean el sistema braille
<b>embosed</b>	Impresoras braille
<b>handheld</b>	Dispositivos de mano: móviles, PDA, etc.
<b>print</b>	Impresoras y navegadores en el modo “Vista Previa para Imprimir”
<b>projection</b>	Proyectores y dispositivos para presentaciones
<b>screen</b>	Pantallas de ordenador
<b>speech</b>	Sintetizadores para navegadores de voz utilizados por personas discapacitadas
<b>tty</b>	Dispositivos textuales limitados como teletipos y terminales de texto
<b>tv</b>	Televisores y dispositivos con resolución baja



### 3.2.2 Principales expresiones a utilizar en las media queries

Al igual que hay varios media types, las media queries tienen varias expresiones a utilizar, las más interesantes para el responsive design son:

- **orientation**: permite tomar por valores *landscape* o *portrait* lo que significa que la orientación sea horizontal o vertical, o sea, *landscape* si la ventana es más ancha que alta y *portrait* si es más alta que ancha.
- **resolution**: Permite las variantes min-resolution y max-resolution. El valor que toma es la densidad de la pantalla en valor de puntos por pulgada (dpi) o en puntos por pixel (dppx). Su uso principal es para cambiar los background de imágenes según el dispositivo sea de alta resolución (como los retina display, iphone, ...) o normal para que no se carguen más datos de los necesarios.
- **width**: El alto del dispositivo. Se permiten las variantes **min-width** y **max-width**. Es la principal expresión para el responsive design.

Atributo	Resultado
<b>min-width</b>	Reglas aplicadas a cualquier ancho de ventana de navegador que supere el valor definido en la consulta.
<b>max-width</b>	Reglas aplicadas a cualquier ancho de ventana de navegador que sea inferior al valor definido en la consulta.
<b>min-height</b>	Reglas aplicadas a cualquier alto de ventana de navegador que supere el valor definido en la consulta.
<b>max-height</b>	Reglas aplicadas a cualquier alto de ventana de navegador que sea inferior al valor definido en la consulta.
<b>orientation=portrait</b>	Reglas aplicadas a cualquier navegador cuyo alto de ventana sea superior o igual al ancho.
<b>orientation=landscape</b>	Reglas aplicadas a cualquier navegador cuyo ancho sea superior al alto.

Cuando la ventana del navegador sea más ancha que su altura, es una orientación denominada "Horizontal":

```
@media only screen and (orientation: landscape) {  
  body {  
    background-color: lightblue;  
  }  
}
```

### 3.3 Operadores lógicos para las Media Queries

Para combinar diversas condiciones podemos usar los operadores lógicos, de una manera similar a como se usan en lenguajes de programación. Los que tenemos disponibles son:

- Operador **and**: las dos condiciones deben cumplirse para que se evalúe como verdadera.
- Operador **not**: es una negación de una condición. Cuando esa condición no se cumpla se aplicarán las medias queries.
- Operador **only**: se aplican las reglas solo en el caso que se cumpla cierta circunstancia.
- Operador **or**: no existe como tal, pero puedes poner varias condiciones separadas por comas y cuando se cumpla cualquiera de ellas, se aplicarán los estilos de las medias queries.

### 3.4 Configurar ventanas gráficas

Debe incluir el siguiente <meta> elemento de vista en todas sus páginas web:

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

Un <meta>elemento de ventana proporciona las instrucciones del navegador sobre cómo controlar las dimensiones y la escala de la página.

La **width=device-width**: establece el ancho de la página para seguir el ancho de pantalla del dispositivo (que variará según el dispositivo).

La **initial-scale=1**: establece el nivel de zoom inicial cuando el navegador carga la página por primera vez.

### 3.4.1 Tamaño de contenido a la ventana gráfica

Los usuarios suelen desplazarse por los sitios web verticalmente tanto en el escritorio como en dispositivos móviles, ¡pero no horizontalmente! Por lo tanto, si el usuario se ve obligado a desplazarse horizontalmente, o alejarse, para ver toda la página web, se obtiene una experiencia de usuario deficiente.

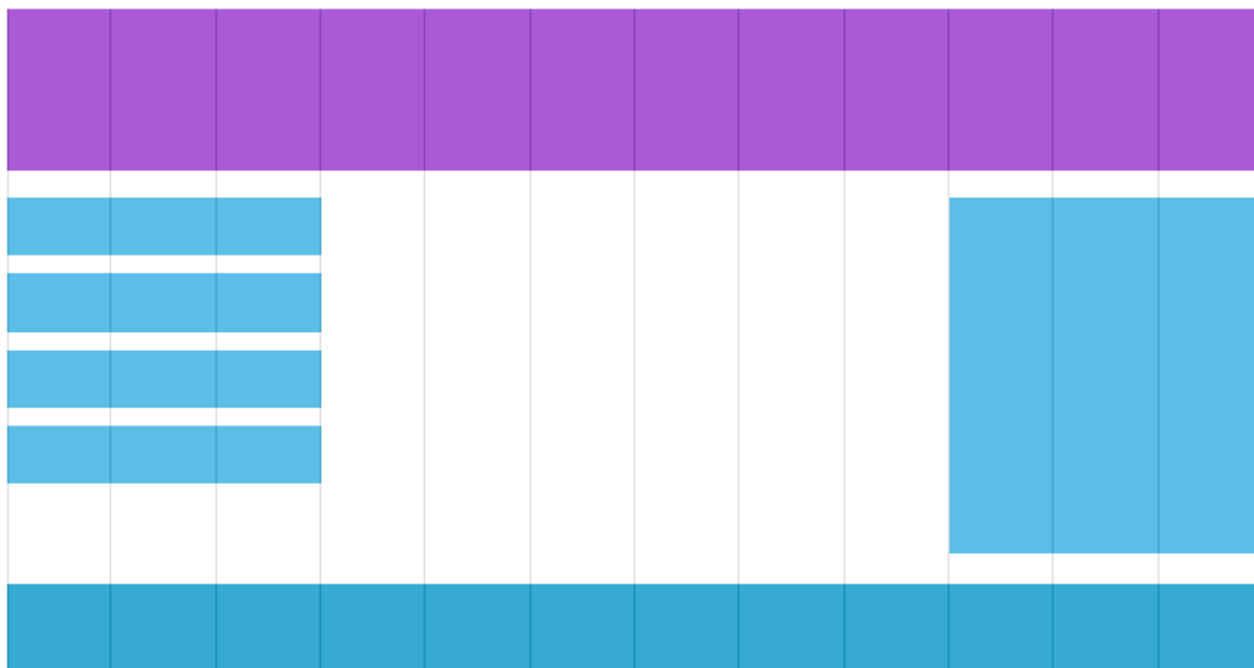
Algunas reglas adicionales que seguir:

- **NO use elementos grandes de ancho fijo:** por ejemplo, si una imagen se muestra con un ancho más ancho que la ventana gráfica, puede hacer que la ventana se desplace horizontalmente. Recuerde ajustar este contenido para que se ajuste al ancho de la ventana gráfica.
- **NO permita que el contenido dependa de un ancho de ventana particular para representarlo bien.** Dado que las dimensiones y el ancho de la pantalla en píxeles CSS varían ampliamente entre los dispositivos, el contenido no debe depender del ancho de una ventana gráfica para renderizar bien.
- **Utilice @media query de CSS para aplicar estilos diferentes para pantallas pequeñas y grandes.** Establecer anchos de CSS absolutos grandes para elementos de página hará que el elemento sea demasiado ancho para la ventana gráfica en un dispositivo más pequeño. En su lugar, considere usar valores de ancho relativo, como ancho: 100%. Además, tenga cuidado de usar valores grandes de posicionamiento absoluto. Puede hacer que el elemento caiga fuera de la ventana gráfica en dispositivos pequeños.

## 3.5 ¿Qué es una vista de cuadrícula?

Usar una vista de cuadrícula es muy útil cuando se diseñan páginas web. Hace que sea más fácil colocar elementos en la página.

Una vista de cuadrícula responsive a menudo tiene 12 columnas y tiene un ancho total de 100%, y se contraerá y expandirá a medida que cambie el tamaño de la ventana del navegador.



## Construyendo una cuadrícula sensible

Vamos a comenzar a construir una vista de cuadrícula sensible.

Primero asegúrese de que todos los elementos HTML tengan la **propiedad box-sizing configurada a border-box**. Esto asegura que el relleno y el borde estén incluidos en el ancho y la altura total de los elementos.

```
* {  
  box-sizing: border-box;  
}
```

Queremos utilizar una vista de cuadrícula receptiva con 12 columnas para tener más control sobre la página web. Primero debemos calcular el porcentaje para una columna:  $100\% / 12 \text{ columnas} = 8.33\%$ .

Luego **hacemos una clase para cada una de las 12 columnas**, `class="col-"` y un número que define cuántas columnas debe abarcar la sección:

```
.col-1 {width: 8.33%;}  
.col-2 {width: 16.66%;}  
.col-3 {width: 25%;}  
.col-4 {width: 33.33%;}  
.col-5 {width: 41.66%;}  
.col-6 {width: 50%;}  
.col-7 {width: 58.33%;}  
.col-8 {width: 66.66%;}  
.col-9 {width: 75%;}  
.col-10 {width: 83.33%;}  
.col-11 {width: 91.66%;}  
.col-12 {width: 100%;}
```

Todas estas **columnas deben estar flotando a la izquierda** y tienen un relleno de 15 píxeles:

```
[class*="col-"] {  
  float: left;  
  padding: 15px;  
}
```

Cada fila debe estar envuelta en a `<div>`. El número de columnas dentro de una fila siempre debe **sumar hasta 12**:

```
<div class="row">  
  <div class="col-3">...</div> <!-- 25% -->  
  <div class="col-9">...</div> <!-- 75% -->  
</div>
```

Las columnas dentro de una fila flotan hacia la izquierda y, por lo tanto, se sacan del flujo de la página y otros elementos se colocarán como si las columnas no existieran. Para evitar esto, **agregaremos un estilo que borre el flujo**:

```
.row::after {  
  content: "";
```

```
clear: both;
display: table;
}
```

## 4 Tendencias de diseño. Mobile First

La filosofía *Mobile First*, significa **diseñar** (para responsive design webs) **primero para móviles**, obligando así a centrarse únicamente en los elementos y las acciones más importantes de un sitio web, creando una experiencia y una buena usabilidad para el usuario. Es mucho más fácil rellenar que limpiar. Eso sí, cuidado con el *Mobile First*, no hay que olvidar que la mayor parte de usuarios de móviles y tablets también pasan gran parte de sus horas delante de un PC o portátil y que estos dispositivos todavía no han abandonado nuestras vidas .



```
mobile-portrait: min:320px - max:479px;
mobile-landscape: min:480px - max:599px;
small-tablet-portrait: min:600px - max:799px;
small-tablet-landscape: min:800px - max:767px;
tablet-portrait: min:768px - max:1023px;
tablet-landscape: min:1024px;
```