

UD7

DESARROLLO WEB



Este documento está bajo una licencia de Creative Commons

Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional



ÍNDICE

1. La World Wide Web
2. HTML (HyperText Markup Language)
 - a) Formateando textos
 - b) Tablas y capas
 - c) HTML5
3. CSS
4. Diseño de Interfaz
 - a) Diseño Responsive
 - b) Bootstrap
5. Javascript

LA WORLD WIDE WEB (WWW)



- La World Wide Web es un sistema de intercambio de información multimedia desarrollado en 1989 en el CERN (ginebra)
 - Fue en el CERN donde se desarrolló el primer navegador, son los programas que usamos para visualizar sitios web.
- El *World Wide Web Consortium* ([W3C](#)) es una organización que agrupa instituciones involucradas en el desarrollo de la WWW, proponiendo nuevas funcionalidades, evaluando propuestas e implementando herramientas.
- La WWW se apoya en el uso del protocolo HTTP y un sistema de descripción de ficheros: el lenguaje HTML

NAVEGADORES

- En el CERN se desarrolló el primer servidor web y el primer navegador
 - Más tarde el NCSA norteamericano desarrolló el navegador gráfico Mosaic lo que dio dimensión multimedia a la WWW.
- Los navegadores son los programas que usamos para visualizar sitios web. Todos contienen un motor de renderizado
 - Webkit (Safari, y anteriormente Chrome, Opera y Edge, la nueva versión de IE)
 - Blink (Chrome, Opera), es un "fork" de Webkit
 - Gecko (Firefox)
 - Trident (Internet Explorer para windows)
 - Servo: optimizado para plataformas Android



Hay que entender que el HTML es un lenguaje "interpretado", que no requiere compilación



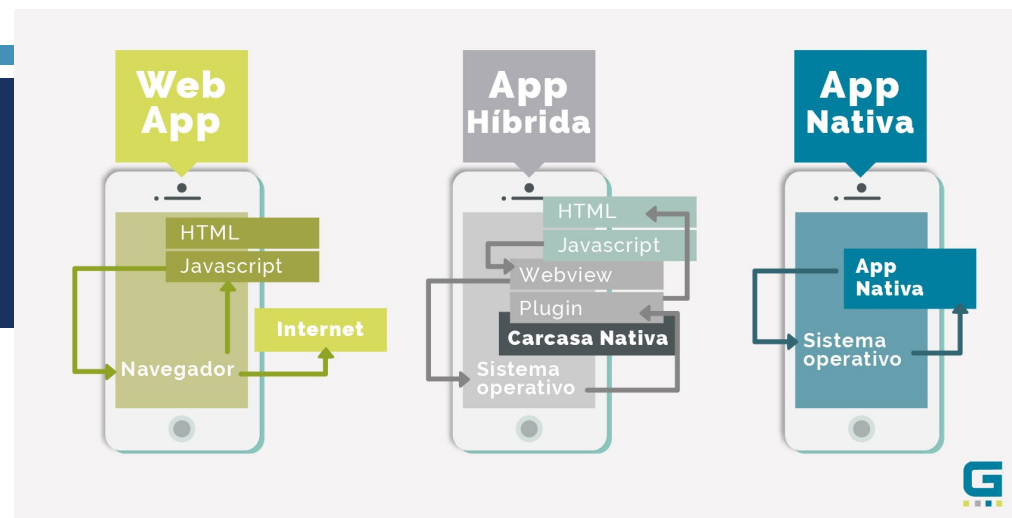


VS



WEB VS APP

- Cuando nos planteamos diseñar una aplicación, se nos presenta una pregunta principal: ¿hacemos una aplicación web (Web App) o una aplicación instalable en dispositivos móviles u ordenadores (App Nativa)
 - En la actualidad, hay una versión intermedia, conocida como “App Híbrida” que utiliza las crecientes funcionalidades que ofrece la Web
- Decidir una cosa u otra tendrá implicaciones en todo el proceso de desarrollo y, por supuesto en el resultado final y las funcionalidades de la aplicación:
 - Web: a través de una página web que sólo requiere crear el producto una vez y puede ser disfrutado en todo tipo de dispositivos a través del navegador. Su problema: el acceso limitado a las funcionalidades del dispositivo
 - Mediante diseño responsive se adapta a las diferentes pantallas.
 - Cada vez son más funcionalidades a las que tiene acceso un navegador (como la cámara, el geolocalizador, activar alertas, etc.)
 - Aplicaciones híbridas: usan un Framework de desarrollo (como React Native) que permite exportar aplicaciones nativas pero usando lenguajes web (HTML y Javascript). Simplifica el proceso, lo hace independiente del SO, pero no consigue todas las ventajas de las nativas
 - Aplicaciones nativas: se adaptan al 100% al dispositivo y puede utilizar sin mayores problemas todas las funcionalidades del mismo (Rendimiento gráfico, GPS, cámara, acelerómetro, acceso a archivos, etc.). Su principal problema es que exige un desarrollo independiente para cada SO (y actualizaciones)

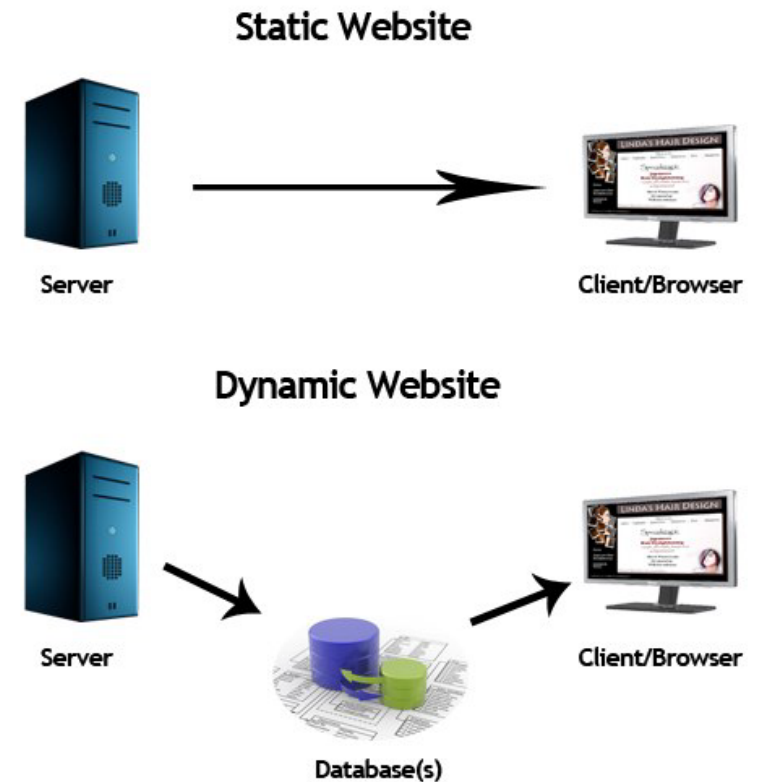


	Web App	App Híbrida	App Nativa
Coste de Desarrollo	Bajo	Medio	Alto
Tiempo de Desarrollo	Corto	Medio	Largo
Mantenimiento	Fácil	Medio	Complejo
Experiencia de Usuario	Buena	Bastante Buena	Excelente
Funcionalidad Offline	Compleja	Compleja	Fácil
Acceso al dispositivo	Parcial	Alto/Complejo	Completo
Velocidad	Rápida	Rápida	Muy Rápida
App Stores	No disponible	Disponible(con limitaciones)	Disponible
Portabilidad del código	Completa	Alta	Nula
Seguridad	Normal	Normal	Alta

Fuente: <https://www.gsoft.es/articulos/que-necesito-web-apps-app-nativa-o-app-hibrida/>

PÁGINAS DINÁMICAS VS. ESTÁTICAS

- A la hora de crear documentos web se puede hacer de dos maneras
 - Páginas estáticas: el contenido se sube a la web, y cuando hay que cambiarlo se vuelve a subir de nuevo.
 - Sólo son útiles si los contenidos no se actualizan con mucha frecuencia (algo cada vez menos habitual)
- Páginas dinámicas: el contenido con el cual se construye la web se encuentra alojado externamente, por lo general en bases de datos o en archivos de intercambio de datos, por lo que no es necesario modificar la web para que el contenido se actualice.
 - Un ejemplo son las páginas PHP que funciona sobre Linux y con bases de datos MySQL (software libre) ó ASP.NET que funciona sobre Windows y bases de datos Access (software propietarios)



CMS (CONTENT MANAGEMENT SYSTEM)

- Un CMS o "Gestor de Contenidos" es una aplicación completa para la creación y gestión de un sitio web. Los más conocidos:
 - Wordpress (<http://es.wordpress.com/>)
 - Joomla (<http://www.joomla.org>)
 - Drupal (<http://www.drupal.org>)
- En cualquier gestor de contenidos es importante diferenciar entre **Frontend** (lo que el su usuario final ve, la parte pública) y el **backend** (el panel de control donde el administrador gestiona el sitio web)



HTML



CSS



JS



```
<defs>
  <linearGradient x1="100%" y1="0%" x2="0%" y2="100%">
    <stop stop-color="#06101F" offset="0%" />
    <stop stop-color="#1D304B" offset="100%" />
  </linearGradient>
</defs>
<rect width="800" height="450" rx="8" fill="url(#media-control)" />
</svg>
<div class="media-control">
  <svg width="96" height="96" viewBox="0 0 96 96">
    <defs>
      <linearGradient x1="87.565%" y1="15.872%">
        <stop stop-color="#FFF" stop-opacity="0" offset="0%" />
        <stop stop-color="#FFF" offset="100%" />
      </linearGradient>
      <filter x="-500%" y="-500%" width="1000%" height="1000%">
        <feOffset dy="16" in="SourceAlpha" result="shadowOffset" />
        <feGaussianBlur stdDeviation="24" in="shadowOffset" result="shadowBlur" />
        <feColorMatrix values="0 0 0 0 0 0 0 0 0 0 0 0 0 0 0" />
      </filter>
    </defs>
    <rect fill-rule="evenodd" cx="48" cy="48" r="48" fill="url(#media-control)" />
  </svg>
</div>
```

HTML

HYPERTEXT MARKUP LANGUAGE

HTML

- El código es la estructura invisible sobre la que se construye cualquier página o aplicación web.



¿QUÉ ES UN DOCUMENTO HTML?

- Un documento HTML, ó "página web" es un archivo "hipertexto"
<HyperText Markup Language>
- Se puede construir con cualquier programa de edición de texto (Notepad en Windows o TextEdit de MAC), pero lo mejor es usar editores de código específicos (como el [CODA](#), [Sublime TEXT](#), [Brackets](#) o Dreamweaver con su editor WYSIWYG)
 - Además del editor de texto, estos programas suelen incorporar módulos con funcionalidades adicionales, como la sincronización con el servidor remoto, o *plugins* para scripts y lenguajes concretos
- El documento HTML sólo contendrá texto, por lo que el resto de contenidos multimedia se vincularán a él desde "el exterior":
 - Imágenes: vinculadas a una fuente ("src") mediante una dirección relativa o absoluta, y en formatos adecuados para la web: jpg, png, gif, svg
 - Objetos: todo aquello que no sea texto o imagen (vídeos, música, películas, objetos flash, etc)



LENGUAJE DE ETIQUETAS

`<casa>` ← etiqueta de apertura
Muy bonita, en el campo, con muchas posibilidades. ← contenido de la etiqueta
`</casa>` ← etiqueta de cierre, siempre precedida de /

- HTML = "Hipertexto en Lenguaje Marcado de Etiquetas"
 - La manera de introducir y jerarquizar la información dentro de la página web es a través de etiquetas.
 - Una etiqueta viene definido por los símbolos "<" y ">", seguido del contenido de dicha etiqueta, y cerrada con la barra "/"

ANIDACIÓN

- Dentro de una etiqueta y, antes de su cierre, podemos anidar otras etiquetas que definan, apoyen o contengan elementos propios de la etiqueta padre.

`<casa>` ← etiqueta de apertura
Muy bonita, en el campo, con muchas posibilidades. ← contenido de la etiqueta
`<ventana>` ← etiqueta de apertura
Para que entre un poco de luz. ← contenido de la etiqueta
`</ventana>` ← etiqueta de cierre, siempre precedida de /
`<puerta>` ← etiqueta de apertura
Para poder entrar en la casa. ← contenido de la etiqueta
`</puerta>` ← etiqueta de cierre, siempre precedida de /
`</casa>` ← etiqueta de cierre, siempre precedida de /

```
<puerta color="abedul"
altura="1mt" hojas="2">
</puerta>
```

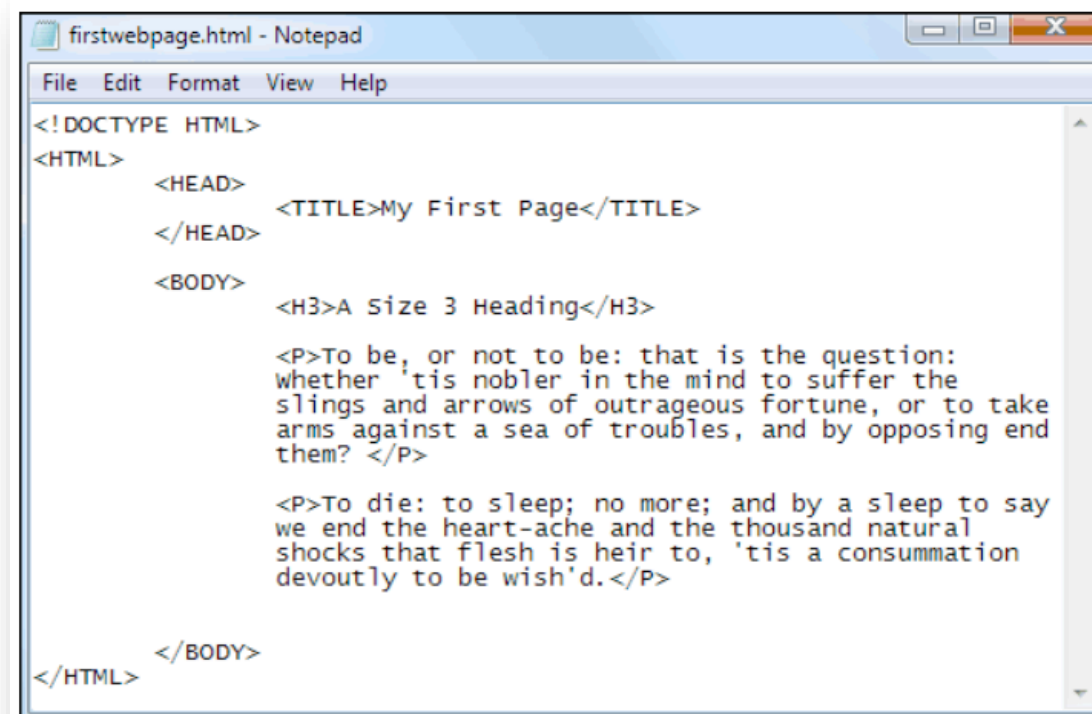
ATRIBUTOS

- Cada etiqueta, además de abrirse y cerrarse y contener otras etiquetas anidadas, puede tener una serie de atributos que la definan y diferencien de las demás.
- A los atributos de una etiqueta se les asigna un valor el cuál se coloca detrás de un símbolo de igual (=) y un valor entre comillas (""), todo ello colocado antes del cerrar la etiqueta de apertura.

<HTML><HEAD><BODY>

Estas tres etiquetas son la parte fundamental de todo documento HTML. Cada una define una parte del documento y contiene determinados elementos fundamentales para él:

- `<html>`
 - Abre y cierra el documento, define la codificación del texto y el tipo de HTML a usar (XHTML, HTML5, etc.)
- `<head>`
 - La primera etiqueta anidada dentro del `<html>`
 - Definirá el título de la página, las palabras claves y la descripción (SEO), los archivos o las definiciones de estilo CSS y las librerías de scripts (Javascript, etc.).
- `<body>`
 - Inmediatamente tras cerrar el `<head>`
 - Es donde se ubica la web (textos, imágenes, vídeos, etc.)
 - Al cerrar el `<body>` se cierra también el `<html>` y el documento.



```
firstwebpage.html - Notepad
File Edit Format View Help
<!DOCTYPE HTML>
<HTML>
  <HEAD>
    <TITLE>My First Page</TITLE>
  </HEAD>
  <BODY>
    <H3>A size 3 heading</H3>
    <P>To be, or not to be: that is the question:
    whether 'tis nobler in the mind to suffer the
    slings and arrows of outrageous fortune, or to take
    arms against a sea of troubles, and by opposing end
    them? </P>
    <P>To die: to sleep; no more; and by a sleep to say
    we end the heart-ache and the thousand natural
    shocks that flesh is heir to, 'tis a consummation
    devoutly to be wish'd.</P>
  </BODY>
</HTML>
```



FORMATEANDO TEXTO EN HTML

PÁRRAFOS Y ENCABEZADOS

- La etiqueta básica para componer texto en un párrafo es `<p></p>`
 - Ej.- `<p>Hola mundo</p>`
- Otra forma de crear párrafos de manera jerarquizada son las etiquetas de encabezados (*headings*) `<h></h>`. Se dispone desde 1 hasta 6 por defecto

```
<h1>Main Heading</h1>
├── <h2>Sub Heading</h2>
│   └── <h3>Sub Sub Heading</h3>
├── <h2>Sub Heading</h2>
│   ├── <h3>Sub Sub Heading</h3>
│   └── <h3>Sub Sub Heading</h3>
```

ETIQUETAS DE EDICIÓN DE TEXTO

- Negrita (*bold*): `` ó ``
- Cursiva (*italic*): `` ó `<i></i>`
- Subrayado (*underline*): `<u></u>`
 - Nota: En HTML5 la etiqueta `<u>` ha sido depreciada, siendo recomendable emplear atributos de estilo ([text-decoration: underline;](#))
- Listas:
 - *Unordered list*: `...`
 - *Ordered list*: `...`
 - *List Item*: ``
- Líneas horizontales: `<hr>`

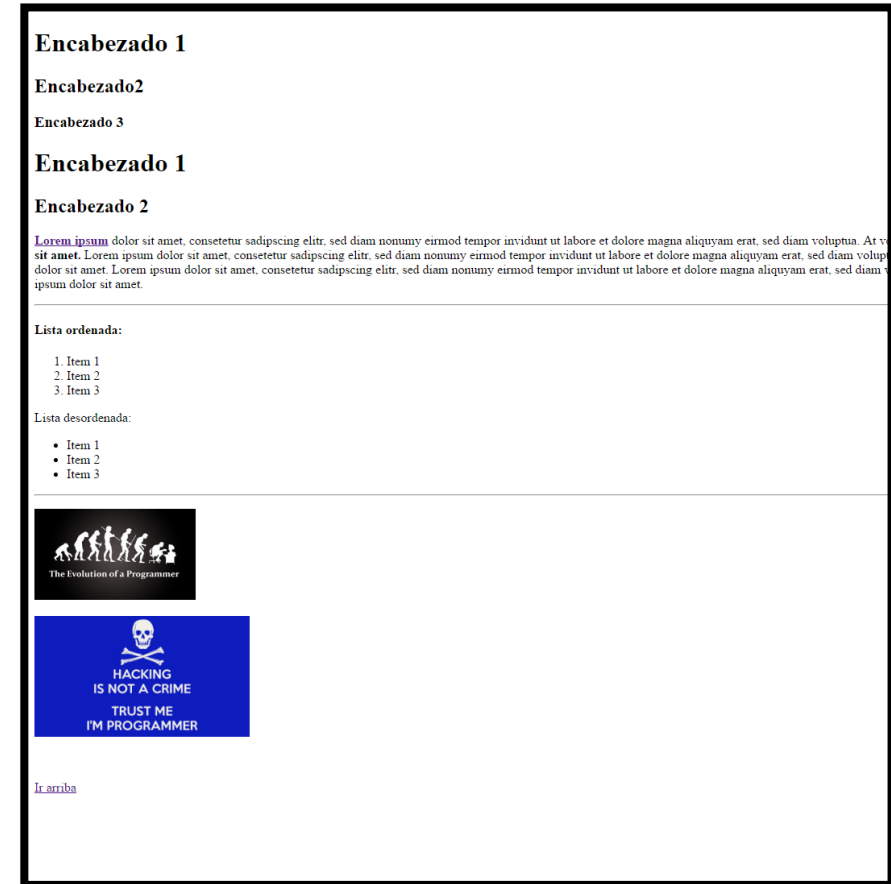
IMÁGENES

- El contenido multimedia más sencillo que se puede incluir en nuestro documento HTML son las imágenes.
- Se utiliza la etiqueta ``, con el atributo básico de "src".
 - Ej.- ``
- Otros atributos son:
 - **width** (ancho) y **height** (alto) expresados en píxeles (si no se especifican la imagen adopta su tamaño real), y si sólo se indica uno de los dos parámetros, el otro adoptará automáticamente el que permita conservar la proporción de la imagen original
 - **alt**: texto alternativo para cuando la imagen no se encuentre o para usuarios con discapacidad visual que usan un interpretador
 - **title**: pequeña descripción de la imagen que aparece al pasar el ratón sobre ella.
- Las rutas de las imágenes (como cualquier elemento externo) puede tener dos tipos de enlaces:
 - Absolutas (ej.- `http://www.midominio.com/images/mifoto.jpg`)
 - Relativas, en relación al documento desde el que se enlaza. Es importante guardar todos los archivos en un árbol de directorios del sitio, y evitar acentos y caracteres especiales (espacios, ñes, etc.)

HIPERVÍNCULOS

- Enlace: `<a>`. Dentro de estas etiquetas estará el contenido visible (texto, imagen, etc.) y dentro de la etiqueta se definen los atributos:
 - **href**="...": dirección a la que apunta el enlace
 - **target**="_blank": abrirá el enlace en una página nueva
 - Anclas: permiten crear un enlace a una parte del documento, poniendo como destino el nombre del ancla (del atributo **"name"** o **"id"**) precedido por un **#**. Ejemplo:
 - ``
 - `Ir arriba`
- Además de un texto, se puede enlazar una imagen incluyendo la etiqueta `` dentro de la etiqueta `<a>`
- Los enlaces pueden ser igualmente absolutos o relativos, y en este caso hay que cuidar la estructura del sitio.

- Crearemos una página web HTML5 desde cero.
- Tiene que tener todos los elementos
 - Etiquetas básicas de un documento HTML (partir desde una página en blanco): html, head y body
 - Incluir también:
 - `<!DOCTYPE html>`
 - `<meta charset="utf8">`
 - `<title></title>`
 - Encabezados
 - Párrafos
 - Texto en negrita (usando “b” y “strong”), cursiva y subrayado
 - Una línea horizontal
 - Listas ordenadas y desordenadas
 - Una imagen con ruta relativa y otra con ruta absoluta, con todos los parámetros necesarios
 - Enlaces: a otras url abriéndose en ventanas nuevas, o a la parte superior del documento mediante anclas



PRÁCTICA: PRIMERA PÁGINA WEB

Crearemos una página web con los elementos básicos vistos hasta ahora

TABLAS Y CAPAS (DIVS)

- Antiguamente se usaban tablas dentro del código HTML para organizar el contenido dentro de la web. Actualmente sólo se aceptan para organizar datos tabulares.

```
<table width="250px">
  <tr>
    <td>Columna 1</td>
    <td>Columna 2</td>
  </tr>
</table>
```



- <DIV> ó "capa": la manera correcta, actual y aprobada por la W3C de diseñar webs y agrupar contenido en los documentos HTML
 - Mediante hojas de estilo definiremos la posición y apariencia de estos elementos
 - Gracias a librerías javascript y el nuevo lenguaje CSS3 perdemos cambiar las propiedades de estas capas y crear elementos dinámicos.



Otra forma de agrupar elementos es mediante la etiqueta ""

header

colmask

colmid

colleft

col1 (center)

col2 (left)

col3 (right)

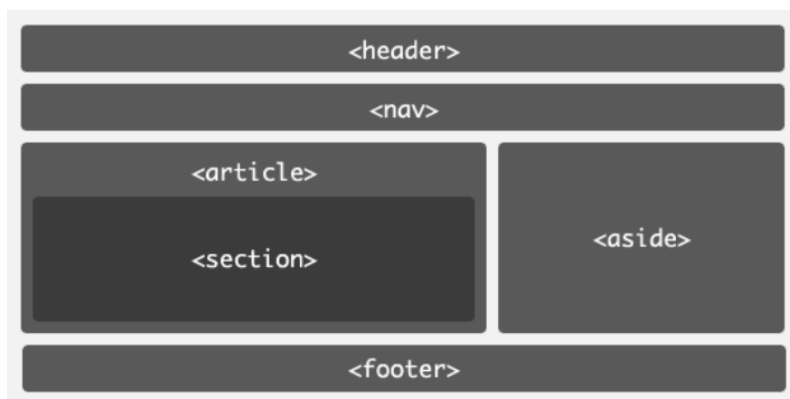
footer

HTML5



- Última versión del lenguaje, que incorpora importantes novedades que amplían las posibilidades multimedia del HTML:

1. Elementos semánticos: etiquetas que ayudan a los motores de búsqueda para organizar el contenido de una página



2. Nuevos elementos de control en formularios: número, fecha, tiempo, calendario, etc.



3. Elementos gráficos: imágenes vectoriales creadas directamente en el código <canvas> y <svg>



4. Elementos multimedia (audio y vídeo)



5. Otros: geolocalización, "drag and drop", almacenamiento de datos local, etc.

- Más información y ejemplos en:

w3schools.com

```
background: 0px 0px 0px #fff;
text-shadow: 0px 0px 0px #fff;
filter: dropshadow(color=#fff,
color:#777;
}
header #main-navigation ul li span:hover {
box-shadow: 0px 0px 1px rgba(50, 50, 50, 0.5);
-webkit-box-shadow: 0px 0px 2px rgba(50, 50, 50, 0.5);
-moz-box-shadow: 0px 0px 1px rgba(50, 50, 50, 0.5);
background-color:#F9F9F9;
color:#444;
}
header #main-navigation ul li.active span {
color:#b90000;
}
header #main-navigation ul li span.dashboard {
background:#F5F5F5 url('../img/dashboard.png');
}
header #main-navigation ul li span.dashboard {
background:#F9F9F9 url('../img/dashboard.png');
}
header #main-navigation ul li.active span.dashboard {
background:#F9F9F9 url('../img/dashboard.png');
}
```

LIBRERÍAS Y HOJAS DE ESTILO CSS

MEJORANDO LA APARIENCIA
DE NUESTRA PÁGINA WEB

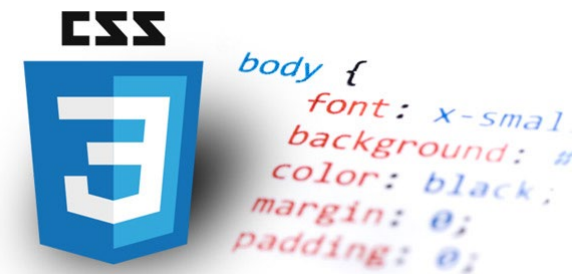


LIBRERÍAS

- Todo aquello que vaya a ser usado en más de una página de un sitio web, en vez de ser repetido en cada una de ellas se externaliza en un archivo aparte y se vincula a cada página que lo requiera
 - Esto permite no sólo reducir el tamaño de la página, sino que al compartir esos scripts o librerías se optimiza el proceso en caso de que haya que modificarlos, ya que en ese caso se hará solo una vez para todas las páginas.
- Mediante la etiqueta `<link>` podemos enlazar el documento con un recurso externo. Se suele ubicar dentro de las etiquetas `<head>` de la página.
- Es un elemento vacío, por lo que sólo tendrá atributos. Los más importantes: "type" y "href" (ubicación)
 - Ej.- `<link type="text/css" href="theme.css">`
- Las librerías más habituales son las del hojas de estilos css y las librerías javascript (veremos las segundas más adelante)

HOJAS DE ESTILO (CSS)

- Mediante los estilos CSS podremos vincular una serie de atributos de apariencia a varios textos a lo largo del documento y de todo el sitio.
- Se pueden vincular de tres formas:
 1. Mediante el atributo "style" dentro de un elemento, conocido como "inline" (no es aconsejable)
 2. Dentro del <head> de la página y a través de la etiqueta <style>. En este caso sólo afectarán a ese documento
 3. Creando un archivo .css y posteriormente vinculándolo en aquellos documentos en los que queremos que se aplique.



```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue">This is a Blue Heading</h1>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="styles.css">
</head>

<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>

</html>
```

```
<!DOCTYPE html>
<html>

<head>
<style>
    body {background-color:lightgrey}
    h1 {color:blue}
    p {color:green}
</style>
</head>

<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>

</html>
```

CÓMO ADJUNTAR ESTILOS CSS: ETIQUETAS, CLASES E IDS

- Sintaxis en la hoja de estilos CSS: nombre del estilo (clase, etiqueta ó ID) y posteriormente encerrados entre "{" y "}" las definiciones del estilo (*propiedad:valor*), separadas por punto y coma (;)

- Ej.-

```
.estilo
{
    width: 630px;
    font-size: 12pt;
    color: #666;
}
```

- Dependiendo de a qué tipo de elemento quiero añadirle el estilo, y cómo, emplearé una de estas 3 fórmulas:
 - Etiquetas HTML: el estilo lleva el nombre de la etiqueta (por ejemplo, body) y se aplicará a todas las etiquetas del documento.
 - Estilos propios: aplicable a cualquier etiqueta mediante el atributo "class" (ej.-). En este caso el estilo deberá ir precedido de un punto.
 - Ids: mediante el atributo ID, presente en el elemento al que le queremos aplicar el estilo y que permite identificarlo dentro de todo el documento. El estilo en ese caso deberá ir precedido de un #

IMPORTANTE: como siempre que se utiliza un identificador (ID), ya sea en una página web, en una base de datos o en lenguaje de programación, no puede haber nunca dos elementos con el mismo ID

ATRIBUTOS CSS MÁS HABITUALES

- Las posibilidades que ofrecen las hojas de estilo son casi ilimitadas. Veamos algunas de las más básicas:
 - Color, tamaño, alineación y fuente de los tipos de texto
 - Imágenes de fondo y forma de mostrar esa imagen (fija, repetida, etc.)
 - Pseudo-clases o estados: permiten modificar el estilo de un enlace en función de su estado:
 - a:hover (al pasar el ratón por encima)
 - a:visited (un enlace ya visitado)
 - a:link (enlaces que no han sido visitados)
 - a:active (al pinchar el ratón en el enlace)
- Más información y ejemplos en:

<http://www.w3schools.com/css/default.asp>

TIPOGRAFÍA MEDIANTE CSS

- Mediante hojas de estilo, podemos determinar la tipografía utilizada:
 - familia tipográfica -> `font-family`
 - Tamaño -> `font-size`
 - Medida absoluta (ej.- 14px)
 - Medida relativa, (em ó rem) más recomendada por cuestiones de accesibilidad.
 - 1em = 100% del tamaño de la fuente (por defecto 16px en los navegadores = 12pt)
 - 0.5em = 50 % del tamaño de la fuente
 - Alineación -> `text-align` (`center` / `right` / `justify`)
 - Color -> `color`
 - Decoración -> `text-decoration` (`none` / `underline` / `line-through` / ...)
 - Estilo -> `font-style` (`normal` / `italic` / `oblique`)
 - Peso -> `font-weight` (`normal` / `bold` / `lighter` /...)
 - Interlineado -> `line-height` (se recomienda entre 1.2em y 1.5em)
 - Espaciado -> `letter-spacing: XXpx;`
 - Versalitas -> `Font-variant: small-caps`
 - Sangría -> `text-indent`
 - Capitular: o bien creando un estilo propio y una etiqueta SPAN, o bien utilizando la subclase `“:first-letter”`;

Más información y recursos:

http://www.w3schools.com/css/css_text.asp

http://www.w3schools.com/css/css_font.asp

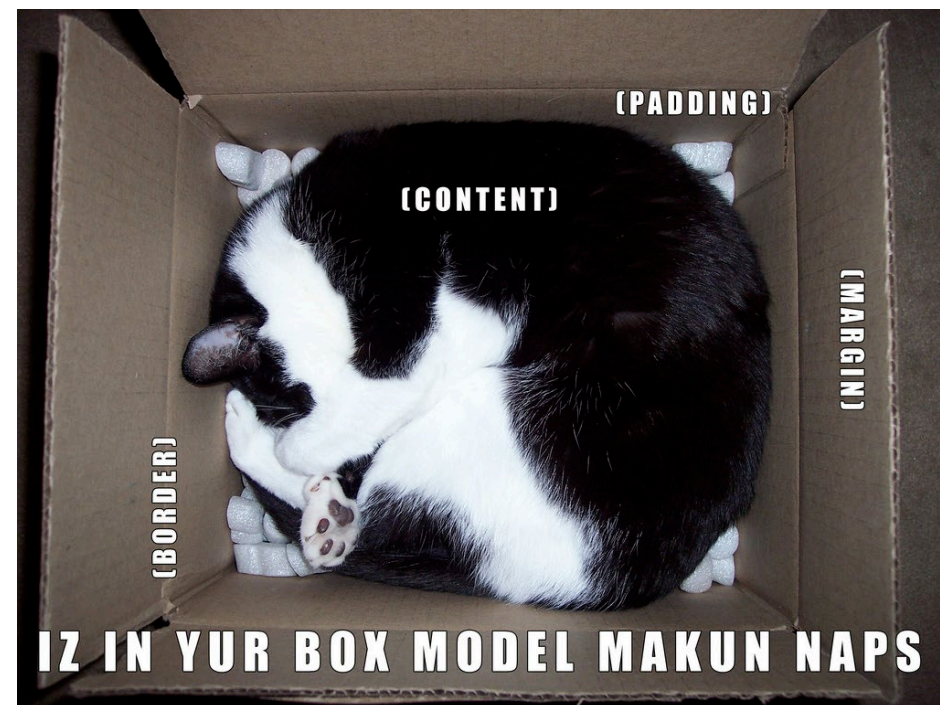
<http://www.font-combinator.com/>

<http://pxtoem.com/>

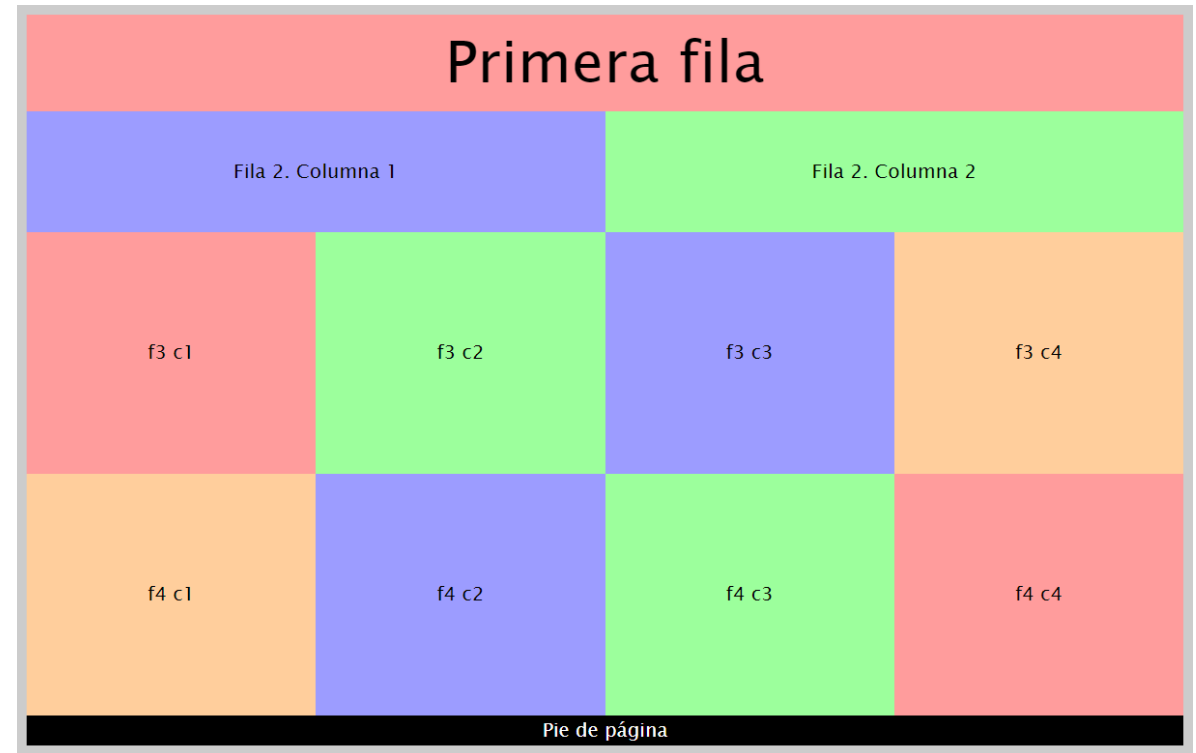
<http://letteringjs.com/>

EL MODELO DE CAJAS DE CSS (BOX MODEL)

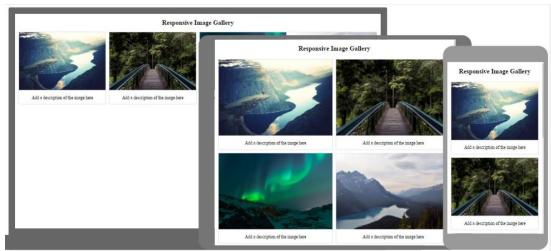
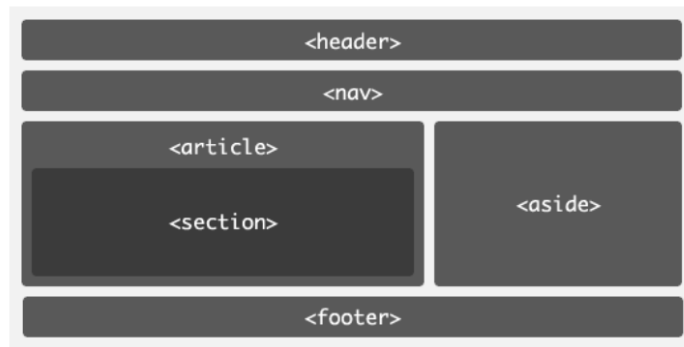
- Uno de los aspectos básicos en CSS es cómo tratar los elementos como si fueran cajas que contienen en su interior los contenidos de la página web (textos, imágenes, etc.)
- Básicamente son:
 - Content: texto e imágenes básicamente
 - Padding: un área que rodea el contenido dentro de la caja
 - Border: el borde, sea visible (*solid*) o no
 - Margin: rodea a la caja y la aleja de otros elementos



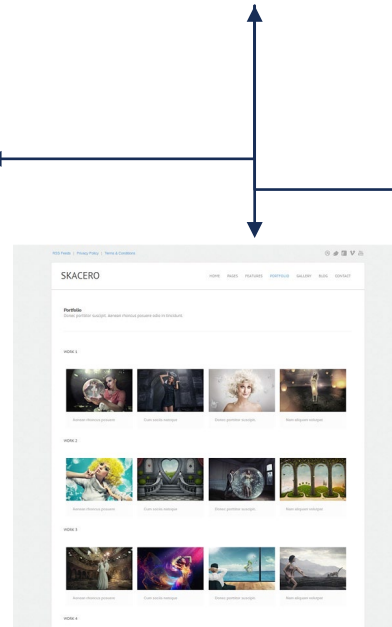
- Mediante capas "div" y librerías CSS se deberá conseguir la estructura que aparece a la derecha.
- Requisitos:
 - No se puede utilizar el atributo "style" dentro de los elementos
 - Se debe optimizar para usar el menor número de líneas de código
 - El cuerpo de la página debe tener un color de fondo que no sea blanco
 - Los colores elegidos deben seguir una armonía de color (puedes ayudarte de herramientas online como la siguiente <https://www.sessions.edu/color-calculator/>)
- Ayudas:
 - Deberás anidar capas ("divs") y estilos CSS
 - Se pueden concatenar estilos (ej.- class="estilo1 estilo2")
 - Para que los divs encabalguen deberás usar la propiedad "float" (y si es necesario "clear")
 - La capa del pie, al tener fondo negro deberá usar texto de color blanco



index



Galería



Portfolio



Artículo

- Reto:
 - Crear una estructura web compleja con varias páginas en las que practicar las librerías CSS y el box-model
 - Añadir botones que permitan la navegación mediante hipervínculos
 - Crear estructuras de malla (wireframe) similares y compartir hoja de estilos entre todas las páginas
 - Crear diseños responsive
- Opcional:
 - Menús complejos
 - Estructuras mediante flexbox
 - Frameworks para diseño responsive como W3.css o Bootstrap

PRÁCTICAS CON CSSY HTML

Construir un sitio web complejo a partir de un mapa de navegación

CSS3

- Es el último estándar de CSS
- Es retrocompatible con versiones anteriores de CSS
- Ofrece unas posibilidades gráficas antes inimaginables en la web si no era con flash.

http://www.w3schools.com/css/css3_intro.asp



MEDIA QUERIES

- Los “media queries” son una técnica introducida en CSS3 que permite aplicar un estilo u otro dependiendo del tamaño de la pantalla. Ej.-
- Permiten ajustarse al ancho de la pantalla y a su orientación (landscape / portrait)
- También permiten redactar usando operadores lógicos “not”, “and” y “only”:

```
@media (min-width: 700px) and (orientation: landscape) { ... }
```

```
<style>
  @media (max-width: 600px) {
    .sidebar {
      display: none;
    }
  }
</style>
```

El estilo sidebar se ocultará en dispositivos con un ancho menor a 600px

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

En apaisado el fondo se volverá azul

Más ejemplos en https://developer.mozilla.org/es/docs/CSS/Media_queries



TIPOGRAFÍA EN LA WEB



TIPOGRAFÍA DIGITAL

- Es importante tener en cuenta la dificultad para la lectura de fuentes en un monitor de ordenador, donde se forman con píxeles en una baja resolución.
- Recomendaciones:
 - Evitar textos largos, y si es necesario presentarlos en bloques pequeños fáciles de abarcar a simple vista (por ejemplo 30 veces el tamaño de la fuente)
 - Tipografías muy fina, o con remates pequeños, pueden dar problemas de contraste, especialmente en cuerpo pequeños
 - El tamaño no puede ser el mismo que usado en medios impresos. Un cuerpo inferior a 14 píxeles puede ser incómodo en textos largos.
 - A diferencia de los textos impresos, se debe recurrir a anchos uniformes (evitar el *Kerning*), utilizar principalmente fuentes de palo seco y sin serifa (especialmente en cuerpos pequeños), ya que los remates pueden verse irregulares y desdibujados al convertirse en píxeles. El uso de cursiva a veces también dificulta la legibilidad.
 - El interlineado aplica espacios encima y debajo del texto (no sólo debajo como ocurría con los tipos móviles). Se recomienda un 150% del tamaño del texto
 - Las pantallas de ordenador también puede provocar cansancio visual, especialmente con imágenes de alto contraste. Una práctica habitual en web es usar fondos gris muy claro sin llegar a blanco y textos gris muy oscuro sin llegar a negro.
 - Por último, en sistemas informáticos hay que contar con que el usuario tenga las fuentes instaladas en su sistema. Por eso es recomendable usar fuentes habituales.

TIPOGRAFÍA EN LA WEB

Websafe fonts:

- Originalmente, tanto HTML como en CSS, sólo soportaban una pequeña lista de fuentes, las que se suponía que cualquier usuario tendría instaladas por defecto.
- En el CSS se cargan en la web de forma jerárquica, de forma que si no tiene una fuente instalada, utiliza la siguiente.
 - Ej.- “font-family:Verdana, Geneva, sans-serif;”
- El diseño web era limitado y se recurría a imágenes si se quería crear efectos de texto más elaborados.
- Ejemplos y disponibilidad:
 - http://www.w3schools.com/cssref/css_websafe_fonts.asp
 - <http://www.cssfontstack.com/>

Webfonts

- Habitualmente una fuente se carga en un cliente mediante los formatos True Type (TTF) o Postscript (PS), pero ofrece problemas en representaciones multi-idioma.
- Actualmente se pueden alojar tipografías localmente (mediante el atributo `@font-face` de css) o en servidores y cargarlas mediante URLs absolutas.
- Las Open Type Fonts (.otf) es un formato cada vez más estándar en los navegadores, dado su gran compatibilidad multi-idioma y sus capacidades de edición.
- Se carga en el HEAD del documento desde servidores, de pago o gratuitos como las [google fonts](#) :
 - `<link href=http://fonts.googleapis.com/css?family=Cagliostro rel="stylesheet" type="text/css">`
 - Font-family: ‘Cagliostro’, sans-serif



CODIFICACIÓN DE TEXTOS

RECONOCIMIENTO ÓPTICO DE CARACTERES (OCR), SUBTITULADO Y CODIFICACIÓN



RECONOCIMIENTO ÓPTICO DE CARACTERES

- El reconocimiento óptico de caracteres (OCR) es un proceso dirigido a la digitalización de textos a partir de imágenes para su conversión en archivos de textos editables.
- Cualquier software de escaneo de imágenes incluye un paquete reducido de OCR
- Recomendaciones para realizar un OCR:
 - Usar imágenes de alta resolución (300 dpi mínimo)
 - Usar textos de alto contraste y fuentes no demasiado pequeñas
 - Si el papel es fino y está impreso por ambas caras, usar una superficie opaca para evitar que se transparente
 - Si el texto incluye imágenes, utilizar software capaz de reconocerlas y maquetar el texto en función de ellas
 - Utilizar reconocimiento inteligente de caracteres (ICR) y correctores ortográficos para corregir posibles errores (reparar bien datos numéricos, inmunes a los correctores automáticos)

SUBTITULADO ELECTRÓNICO

- Los subtítulos son los textos que aparecen en el borde inferior de una pantalla, con frecuencia sobre el puesto a ella, adoptando información adicional sobre la misma o traduciendo una narración o diálogo conducido en un idioma extranjero.
- Hay que distinguir entre subtítulos incrustados en la imagen y los que son flotantes (se pueden activar y desactivar). Estos últimos pueden incorporarse de dos formas:
 - Contenedores multimedia (DVD y SVCD que los almacenan como imágenes, o AVUI MP4, OGM, Matroska que los almacenan como texto)
 - Archivos de texto plano que acompañan al archivo de vídeo (con el mismo nombre)
- Formatos de texto plano
 - SubRip (.srt)
 - MicroDVD (.sub)
 - Universal Subtitle Format (basado en xml)
 - Substation Alpha (.ssa)
 - Otros (.smi, .rt, .txt, .aqt, .jss, .js)
- Existe una amplia gama de [software de edición de subtítulos](#)

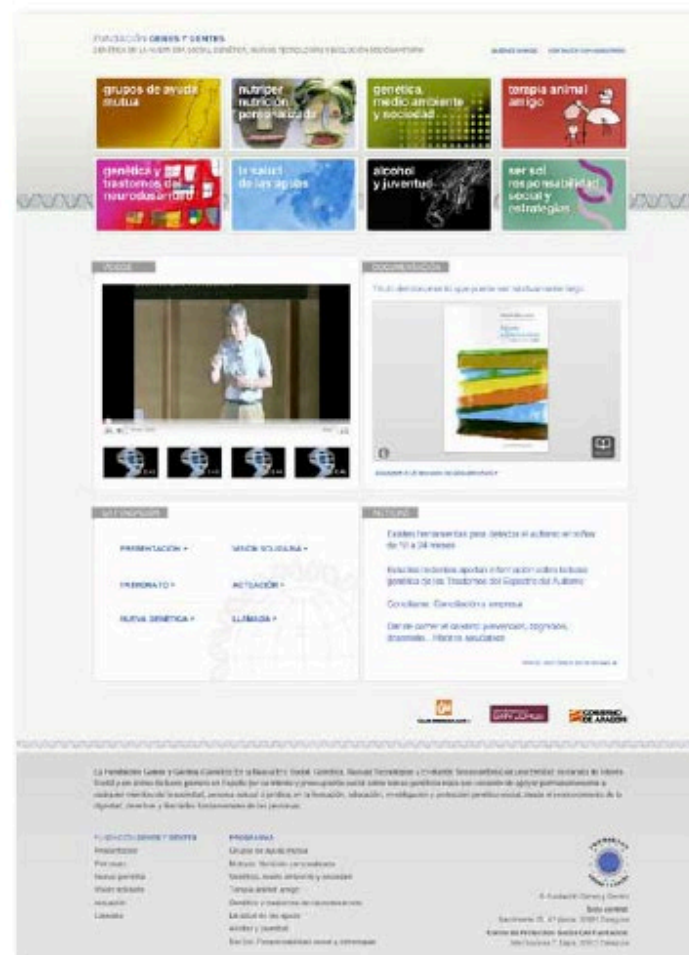
COMPATIBILIDAD E INTERCAMBIO DE FUENTES ENTRE PLATAFORMAS

- La codificación de caracteres es el método que permite convertir un carácter de un lenguaje natural en un símbolo de otro sistema de representación, como un número o una secuencia de pulsos eléctricos en un sistema electrónico, aplicando normas o reglas de codificación
- Hay que tener en cuenta que cada idioma tiene sus propios caracteres (o glifos)
- Normas de codificación:
 - ASCII
 - Limitado a 128 caracteres (7 dígitos binarios)
 - El ASCII Extendido subió a 8 dígitos, pero no era suficiente para dar cobertura a múltiples idiomas.
 - Está recogido en el conjunto de estándares ISO/IEC 8859
 - UNICODE
 - Incluye todos los caracteres de uso común en la actualidad (más de cien mil), tanto de idiomas actuales como de escrituras antiguas.
 - Es ideal para páginas web multi-idioma
 - El nuevo formato tipográfico Open Type, desarrollado por Microsoft y Adobe, y heredera del Tru Type, utiliza este sistema de codificación
 - Según su arquitectura (nº de bits utilizados para representar cada caracter) podemos hablar de 3 formas de codificación:
 - UTF-8
 - UTF-16
 - UTF-32
- En html se establece mediante la etiqueta charset dentro de los metadatos del encabezado
 - `<meta charset="UTF-8" />`



DISEÑO DE INTERFAZ PARA WEB

CAPA VISUAL EN UNA PÁGINA WEB



Elementos gráficos:

- Identidad corporativa
- Tipografía
- Color
- Imágenes y vídeos
- Iconos
- Elementos de navegación

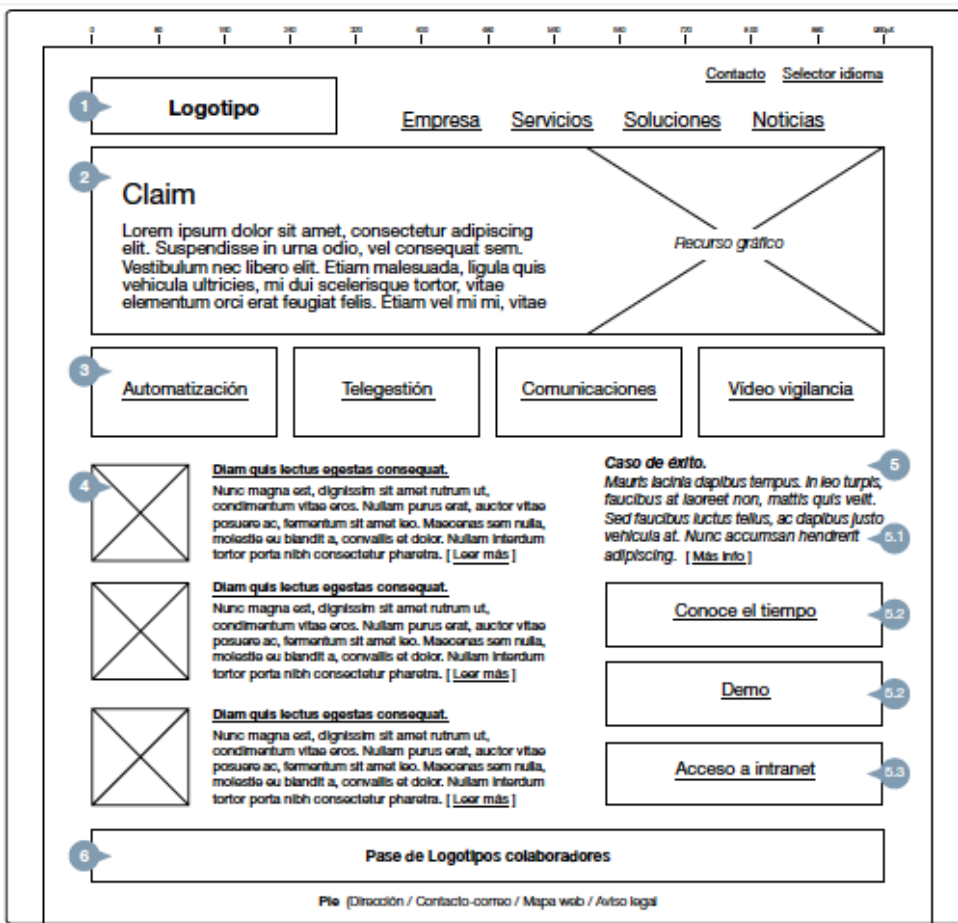
WIREFRAME

PROYECTO:
Tafyesa web

PROYECTO:
Home Vs1

AUTOR: JOSÉ LUIS LEANO
MODIFICADO: 13 SEPTIEMBRE 2010

- Cabecera**
Compuesta de Logotipo (pulzando se va a Inicio), Menú principal, contacto y selector de idioma
- Zona de atención**
Zona de impacto mediante composición gráfica que incluya el claim "al poder del control" y unas frases descriptivas de la actividad, resumen del espíritu de Tafyesa, que contenga enlaces a secciones de la web.
- Accesos directos**
En la página de Inicio aparecen 4 accesos directos gráficamente destacados a estas cuatro áreas de servicio destacadas: (automatización, telegestión, comunicaciones y video vigilancia)
- Noticias**
En Inicio aparecen varias noticias, las más recientes, incorporando una imagen thumbnail.
- Sidebar (constante en toda la página)**
En un sidebar aparecen varios elementos:
 - Caso de éxito.
Breve descripción con enlace para ver la información ampliada de un caso de éxito. prioritariamente será una declaración de un cliente afirmando el caso.
 - Enlaces destacados gráficamente a "Conoce el tiempo" y a "Demo"
 - Acceso a Intranet
- Pase de logotipos**
Logos de entidades/empresas colaboradoras que rotaran en su aparición
- Pie**
Compuesto de dirección física, contacto (correo electrónico), mapa web y aviso legal, si procede.



- Un wireframe para un sitio web, también conocido como un esquema de página o plano de pantalla, es una guía visual que representa el esqueleto o estructura visual de un sitio web.
- El wireframe esquematiza el diseño de página u ordenamiento del contenido del sitio web, incluyendo elementos de la interfaz y sistemas de navegación, y cómo funcionan en conjunto.
- Usualmente este esquema carece de estilo tipográfico, color o aplicaciones gráficas, ya que su principal objetivo reside en la funcionalidad, comportamiento y jerarquía de contenidos.
- Los esquemas pueden ser dibujados con lápiz y papel o esquemas en una pizarra, o pueden ser producidos con medios de diseño de aplicaciones de software libre o comerciales.

El poder del control

Adaptamos nuestra tecnología a sus necesidades y recursos, mejorando las instalaciones mediante la automatización. Logramos un notable ahorro tanto de consumo como económico, además de el control de un proceso controlado y evaluable. Ponemos nuestra experiencia y capacidad a disposición de su proyecto. [¡Contáctanos!](#)



Los servicios de TAFYESA



Facilitamos el acceso a los servicios de TAFYESA, siempre en los puntos de acceso de cada una de las instalaciones. Proporcionamos un servicio de atención al cliente que garantiza la satisfacción de nuestros clientes. Siempre con el objetivo de mejorar la calidad de los servicios que ofrecemos. [¡Contáctanos!](#)

El poder del control



Desde el momento en que se instala el sistema de control, el usuario puede acceder a los servicios de TAFYESA de forma sencilla y segura. Esto garantiza la satisfacción de nuestros clientes. Siempre con el objetivo de mejorar la calidad de los servicios que ofrecemos. [¡Contáctanos!](#)

Con nuestros servicios de automatización



Facilitamos el acceso a los servicios de TAFYESA, siempre en los puntos de acceso de cada una de las instalaciones. Proporcionamos un servicio de atención al cliente que garantiza la satisfacción de nuestros clientes. Siempre con el objetivo de mejorar la calidad de los servicios que ofrecemos. [¡Contáctanos!](#)

Los servicios de TAFYESA



Desde el momento en que se instala el sistema de control, el usuario puede acceder a los servicios de TAFYESA de forma sencilla y segura. Esto garantiza la satisfacción de nuestros clientes. Siempre con el objetivo de mejorar la calidad de los servicios que ofrecemos. [¡Contáctanos!](#)

Gracias a la tecnología de TAFYESA, nuestros clientes pueden acceder a los servicios de TAFYESA de forma sencilla y segura. Esto garantiza la satisfacción de nuestros clientes. Siempre con el objetivo de mejorar la calidad de los servicios que ofrecemos. [¡Contáctanos!](#)

MARCO ANTONIO MARCO LÓPEZ
Asesor de ventas y soporte
cliente

[Ver más de este](#)

MeteoTafyesa

Hecho pensando en la comodidad de los usuarios, el sistema de TAFYESA permite acceder a los servicios de TAFYESA de forma sencilla y segura. [¡Contáctanos!](#)

DemoTafyesa

Conozca los servicios de TAFYESA de forma sencilla y segura. [¡Contáctanos!](#)

Acceso a Internet



DISEÑO ADAPTATIVO (RESPONSIVE) Y DISEÑO FLUIDO

ADAPTAR EL DISEÑO AL TAMAÑO DE LA PANTALLA



EL PROBLEMA DE LA RESOLUCIÓN DE PANTALLA

Desde 2560x1440 (iMac 27") hasta 320 x 480 (iPhone 3Gs), pasando por 768 x 1024 (iPad).

Es imposible diseñar un interfaz web de dimensiones fijas que funcione bien para la mayoría de usuarios.



Desktop.

Web (Media) > 1366 x 768 px

Web (mínimo) > 1024 x 768 px

Web > 1280 X 800

MacBook Pro 13 (Retina) > 2560 x 1600 px

MacBook Pro 15 (Retina) > 2880 x 1800 px



Tablet.

iPad pro > 2047 x 2732 px

iPad retina > 1536 x 2048 px

Android 1080p > 1080 x 1920 px



Smartphone.

Móvil base > 320 x 1024 px

iPhone 5 > 642 x 1136

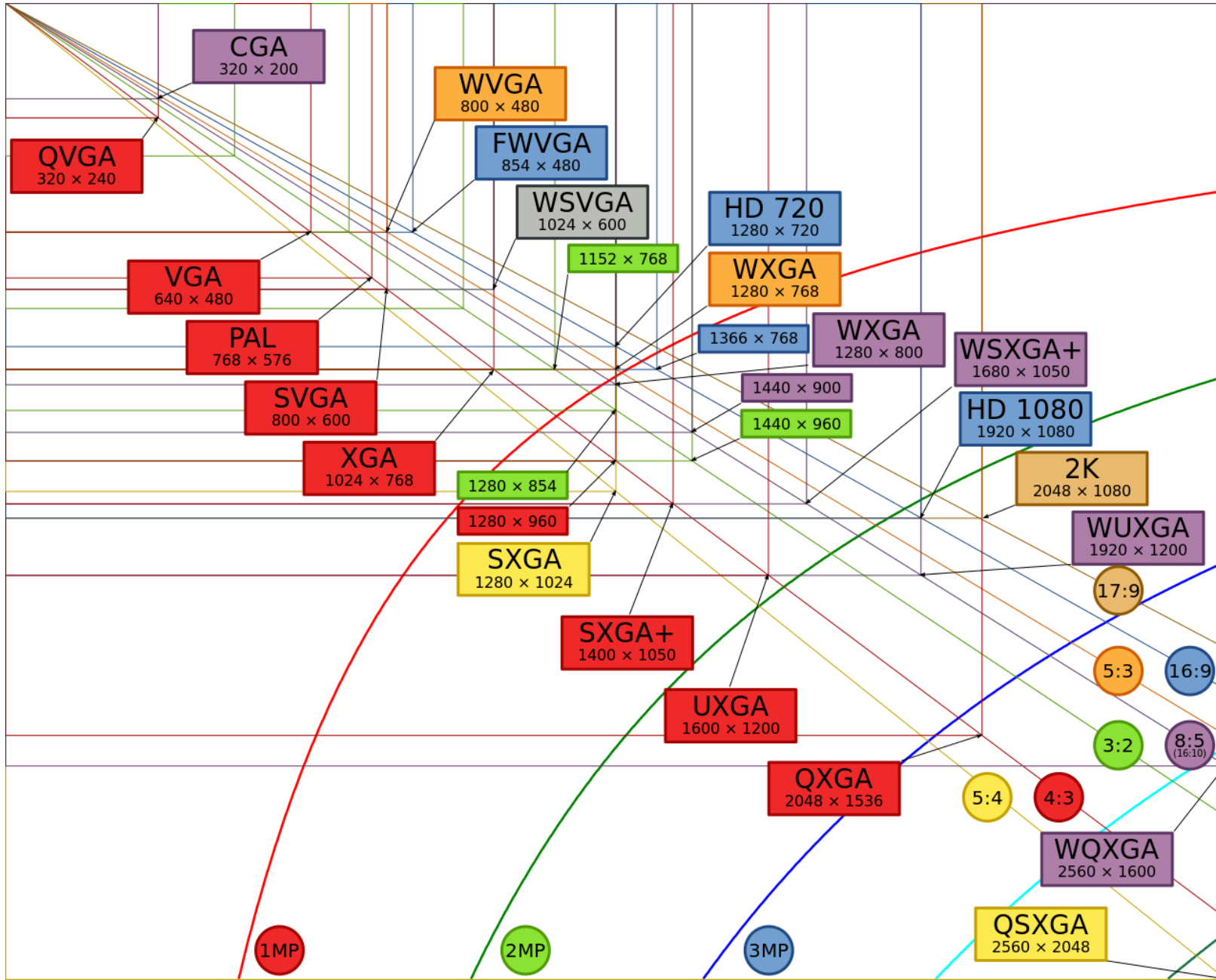
iPhone 6 > 750 x 1334

Top 10 Screen Resolutions

1	1366x768	9.17%
2	640x360	7.83%
3	1920x1080	7.70%
4	1024x768	4.46%
5	896x414	4.28%
6	667x375	4.22%
7	780x360	3.20%
8	812x375	3.03%
9	760x360	2.80%
10	1536x864	2.74%

Actualizado febrero de 2021.

Fuente: <https://www.w3counter.com/globalstats.php>



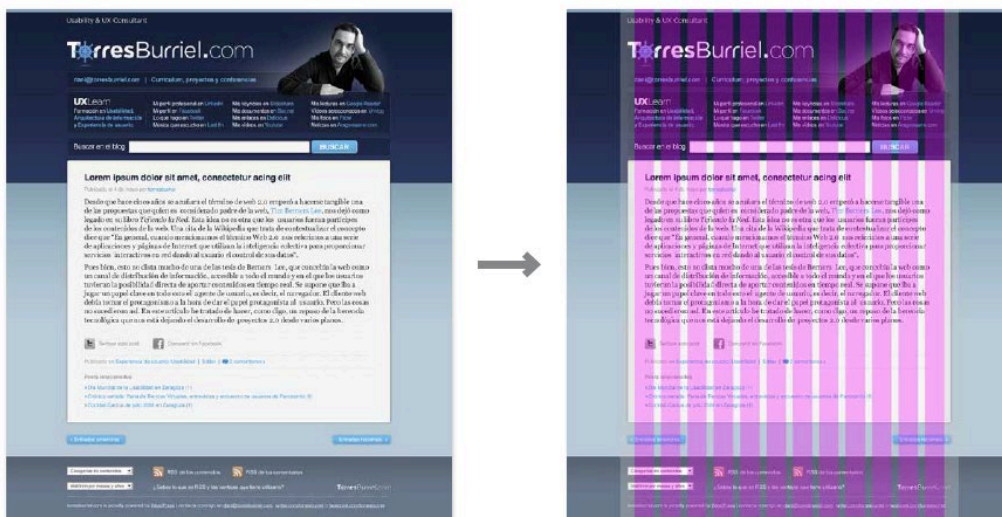
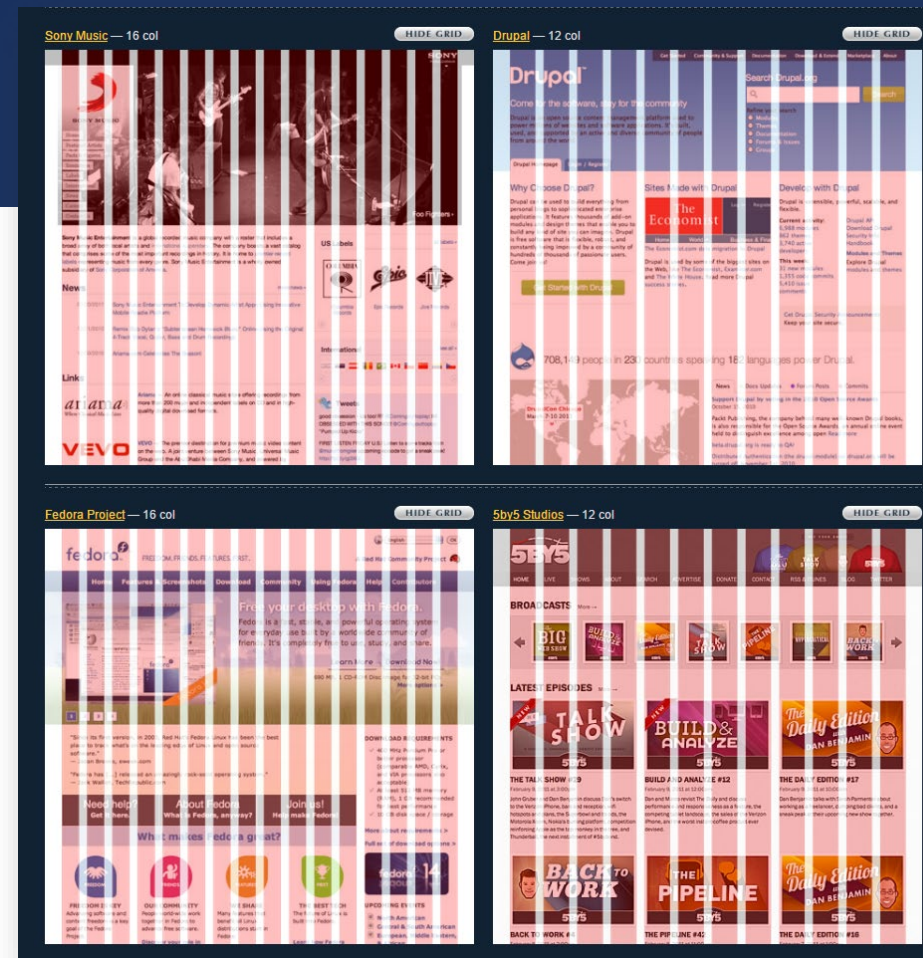
SOLUCIÓN: DISEÑO *RESPONSIVE*



- El diseño web adaptable, también llamado adaptativo o responsivo (*responsive*) permite crear un solo diseño para múltiples pantallas, no solo pequeños dispositivos móviles sino también pantallas de alta resolución
- Se basa en el concepto de One Web, desarrollado por la W3C, que hace referencia a la idea de construir una 'Web para Todos' (Web for All) y accesible desde cualquier tipo de dispositivo (Web on Everything).

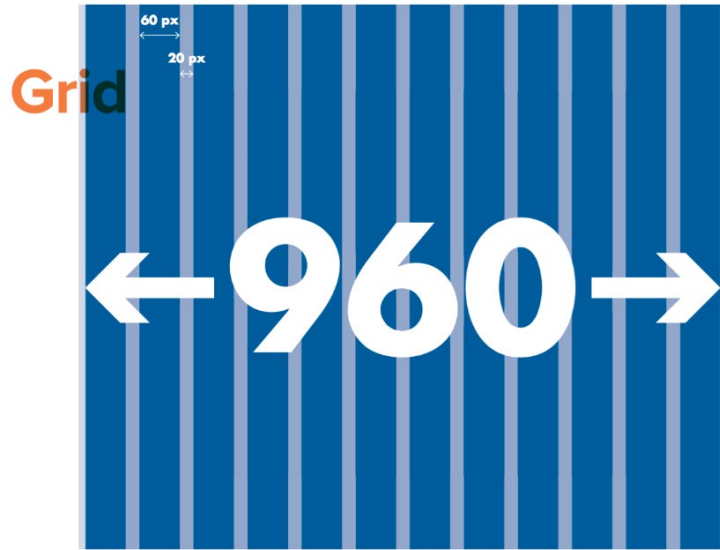
USO DE LA REJILLA (GRID)

- Permiten organizar los elementos de una forma más racional.
- Los elementos se organizan en filas (rows) que a su vez distribuyen su espacio de forma dinámica en columnas (cols), que pueden ser 12, 16 ó 24



Mediante un diseño responsive podemos decidir cuántas columnas ocupa cada elemento dependiendo de la resolución de la pantalla

Fuente: <http://960.gs/>



12 col

Ancho de columna:
60px.

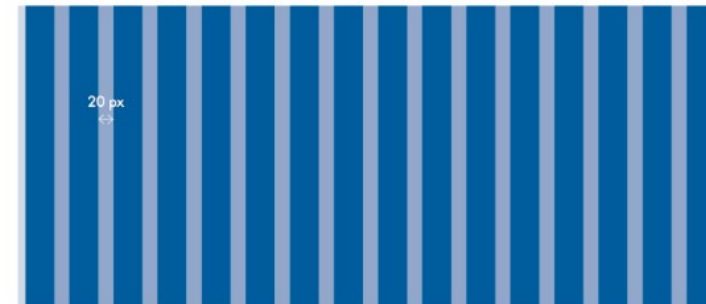
Espacio entre columnas:
20px.

Zona activa de 10px. a
cada lado.

Ejemplos de distribución de columnas para una pantalla de 960 px., según el *960 Grid System*.



16 col

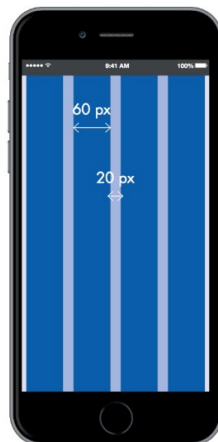


Ancho de columna:
40px.
Espacio entre columnas:
20px.

Zona activa de 10px. a
cada lado.

Grid

**Smartphone
4 col**

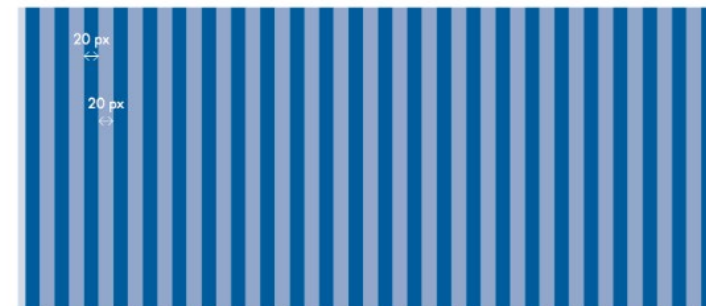


Ancho de columna:
60px.

Espacio entre columnas:
20px.

Zona activa de 10px. a cada
lado.

24 col



Ancho de columna:
20px.

Espacio entre columnas:
20px.

Zona activa de 10px. a cada
lado.

FRAMEWORK RESPONSIVE

- Un framework (“marco de trabajo”) es un conjunto de herramientas, encaminadas a facilitar y sistematizar el desarrollo de una serie de tareas.
- En diseño de software y diseño web es habitual usar frameworks, que a su vez contienen una serie de librerías (ej.- Jquery)
- Aunque Bootstrap es el proyecto más popular de *framework* responsive, existen otros similares:
 - [Foundation](#),
 - [Gumby](#) (depreciado),
 - [Responsive Aeon](#), etc.

Bootstrap 3

Sleek, intuitive, and powerful mobile-first front-end framework for faster and easier web development.

Head's up! Until the final v3 is released, downloads may be behind the development branch.

Download Bootstrap



YAML 4

CSS Framework

A modular CSS framework for truly flexible, accessible and

Browser Support

BOOTSTRAP



- Bootstrap, en su actual versión 4, es un framework de código abierto basado en HTML y CSS, que utiliza librerías Jquery entre otras, para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño, así como, extensiones de JavaScript.

Pasos:

I. Vincular las librerías CSS, Javascript y Jquery al proyecto.

- Se pueden enlazar directamente desde los CDN o [descargarlas](#) para que funcionen en local (el enlace no incluye librerías Jquery ó Popper.js)

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>

<!-- Popper JS -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js"></script>

<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js"></script>
```

Pasos:

2. Crear el documento HTML, incluyendo el <Doctype>, el idioma y la condificación de caracteres.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
  </head>
</html>
```

3. Añadir en el HEAD la etiqueta que permite a Bootstrap establecer el tamaño inicial de ventana correspondiente al del dispositivo.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

4. Crear el contenedor (*container*) que recoge todo a lo que queramos aplicar las librerías. Hay dos tipos:
 - a. `.container` -> todo queda contenido en una capa
 - b. `.container-fluid` -> el contenido se adapta al ancho de la página de forma dinámica (lo que se conoce como "diseño fluido", frente a "responsive" el cual se adapta al ancho pero no de forma continua)

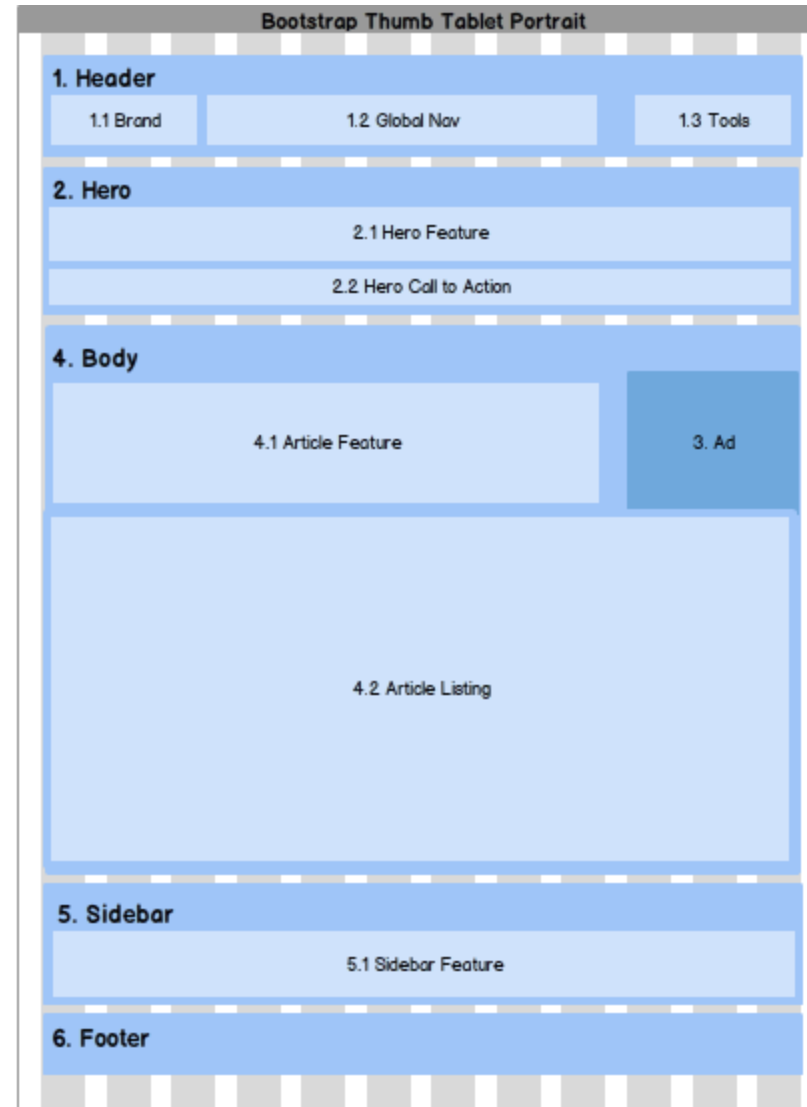
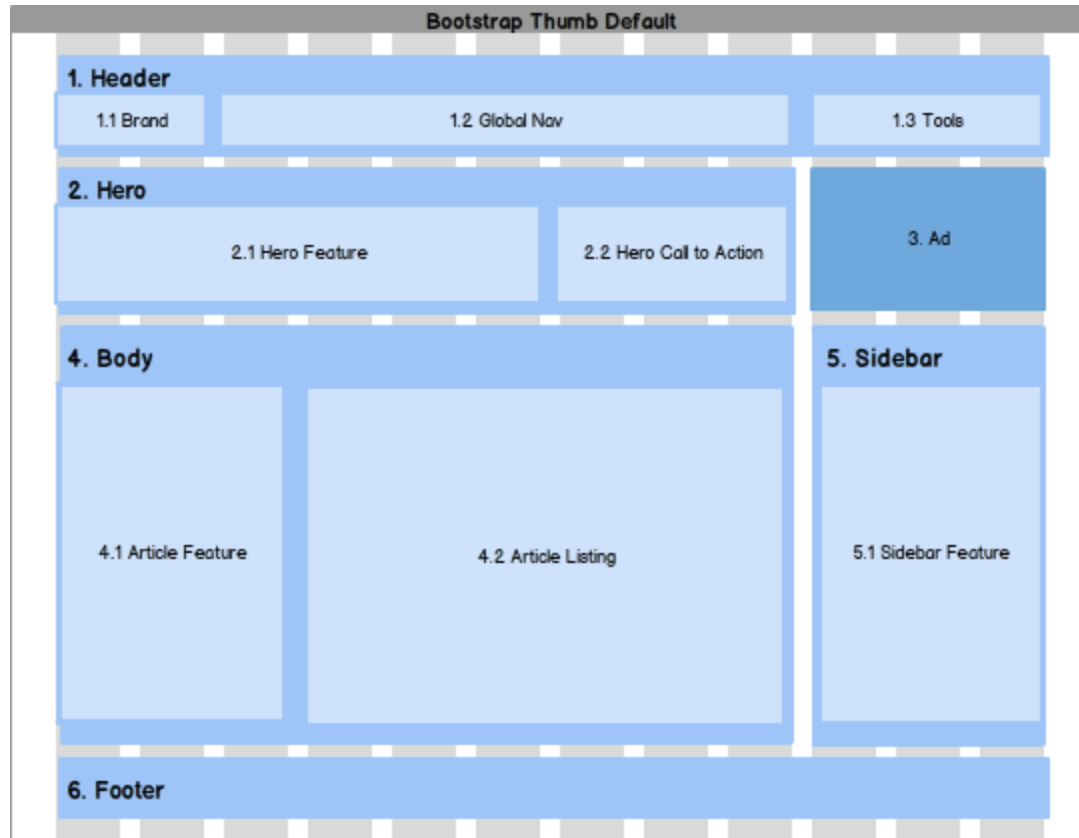
```
<body>
<div class="container">
  <h1>Mi primera página Bootstrap</h1>
  <p>Un texto de prueba.</p>
</div>
</body>
```

SISTEMA DE MALLA (GRID)

- Bootstrap se distribuye en una maya de 12 columnas agrupadas en filas
- Cuántas columnas abarca una capa (div) en cada dispositivo viene determinada por su estilo asociado:
 - col- (extra small devices) <576px
 - col-sm- (small devices) => 576px
 - col-md- (medium devices) => 768px
 - col-lg- (large devices) => 992px
 - col-xl- (xlarge devices) => 1200px

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4			
span 4				span 8							
span 6						span 6					
span 12											

- Las capas deben incorporar los estilos que se aplicarán en cada dispositivo (dependiendo del tamaño de la pantalla) mediante su prefijo seguido del n° de columnas.
 - Hay que tener en cuenta que al sumar 12, el siguiente elemento pasa a la fila siguiente
 - Si queremos que un mismo grupo de columnas se agrupe, deberemos incluirlas en una capa de estilo “row”
 - Si sólo indicamos un tamaño de dispositivo, su ancho se aplicará siempre, y si no indicamos un dispositivo intermedio, se aplicará el inmediatamente inferior



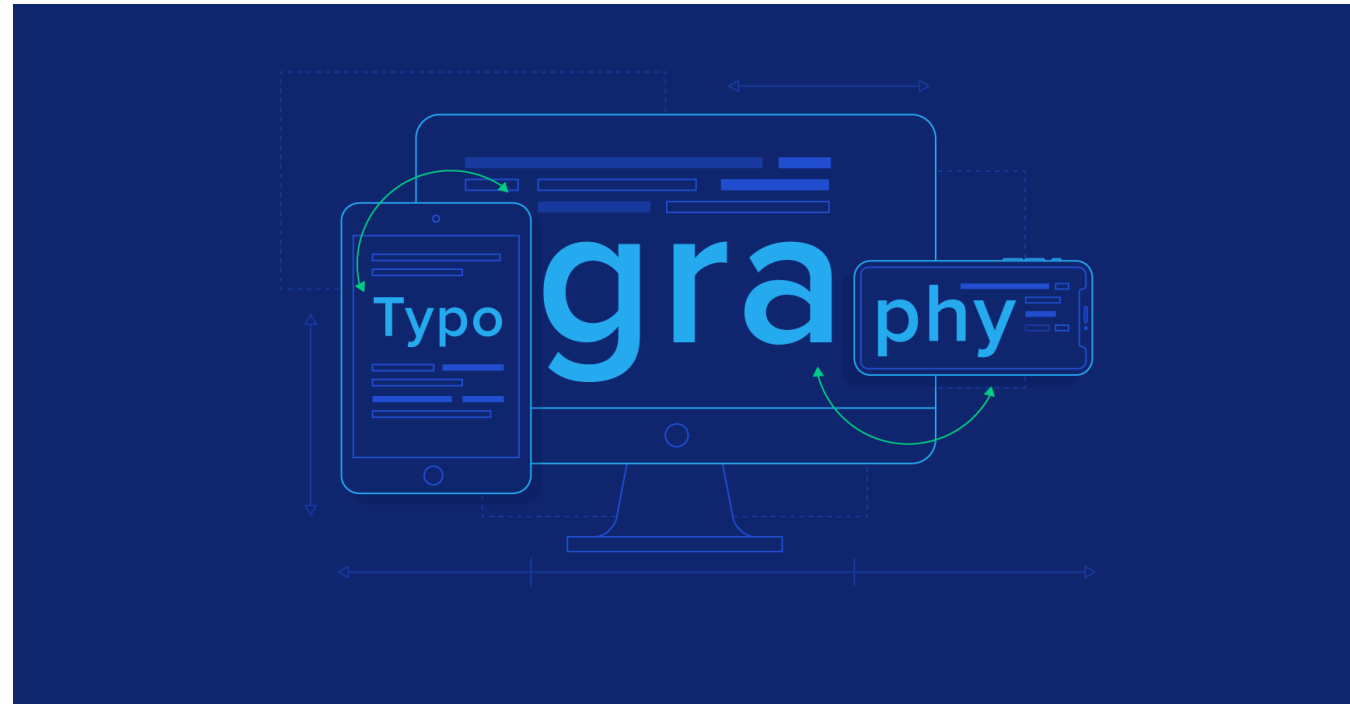
```

<!--Ejemplo primera fila -->

<div class="row">
  <div class="col-lg-2 col-md-2 col-sm-12">1.1 Brand</div>
  <div class="col-lg-7 col-md-7 col-xs-8">1.2 Global Now</div>
  <div class="col-sm-3 col-md-3 col-xs-4">1.3 Tools</div>
</div>

```

TIPOGRAFÍA EN LA WEB



TIPOGRAFÍA DIGITAL

- Es importante tener en cuenta la dificultad para la lectura de fuentes en un monitor de ordenador, donde se forman con píxeles en una baja resolución.
- Recomendaciones:
 - Evitar textos largos, y si es necesario presentarlos en bloques pequeños fáciles de abarcar a simple vista (por ejemplo 30 veces el tamaño de la fuente)
 - Tipografías muy fina, o con remates pequeños, pueden dar problemas de contraste, especialmente en cuerpo pequeños
 - El tamaño no puede ser el mismo que usado en medios impresos. Un cuerpo inferior a 14 píxeles puede ser incómodo en textos largos.
 - A diferencia de los textos impresos, se debe recurrir a anchos uniformes (evitar el *Kerning*), utilizar principalmente fuentes de palo seco y sin serifa (especialmente en cuerpos pequeños), ya que los remates pueden verse irregulares y desdibujados al convertirse en píxeles. El uso de cursiva a veces también dificulta la legibilidad.
 - El interlineado aplica espacios encima y debajo del texto (no sólo debajo como ocurría con los tipos móviles). Se recomienda un 150% del tamaño del texto
 - Las pantallas de ordenador también puede provocar cansancio visual, especialmente con imágenes de alto contraste. Un a práctica habitual en web es usar fondos gris muy claro sin llegar a blanco y textos gris muy oscuro sin llegar a negro.
 - Por último, en sistemas informáticos hay que contar con que el usuario tenga las fuentes instaladas en su sistema. Por eso es recomendable usar fuentes habituales.

TIPOGRAFÍA EN LA WEB

Websafe fonts:

- Originalmente, tanto HTML como en CSS, sólo soportaban una pequeña lista de fuentes, las que se suponía que cualquier usuario tendría instaladas por defecto.
- En el CSS se cargan en la web de forma jerárquica, de forma que si no tiene una fuente instalada, utiliza la siguiente.
 - Ej.- “font-family:Verdana, Geneva, sans-serif;”
- El diseño web era limitado y se recurría a imágenes si se quería crear efectos de texto más elaborados.
- Ejemplos y disponibilidad:
 - http://www.w3schools.com/cssref/css_websafe_fonts.asp
 - <http://www.cssfontstack.com/>

Webfonts

- Habitualmente una fuente se carga en un cliente mediante los formatos True Type (TTF) o Postscript (PS), pero ofrece problemas en representaciones multi-idioma.
- Actualmente se pueden alojar tipografías localmente (mediante el atributo `@font-face` de css) o en servidores y cargarlas mediante URLs absolutas.
- Las Open Type Fonts (.otf) es un formato cada vez más estándar en los navegadores, dado su gran compatibilidad multi-idioma y sus capacidades de edición.
- Se carga en el HEAD del documento desde servidores, de pago o gratuitos como las [google fonts](#) :
 - `<link href=http://fonts.googleapis.com/css?family=Cagliostro rel="stylesheet" type="text/css">`
 - Font-family: ‘Cagliostro’, sans-serif

TIPOGRAFÍA MEDIANTE CSS

- Mediante hojas de estilo, podemos determinar la tipografía utilizada:
 - familia tipográfica -> `font-family`
 - Tamaño -> `font-size`
 - Medida absoluta (ej.- 14px)
 - Medida relativa, (em ó rem) más recomendada por cuestiones de accesibilidad.
 - 1em = 100% del tamaño de la fuente (por defecto 16px en los navegadores = 12pt)
 - 0.5em = 50 % del tamaño de la fuente
 - Alineación -> `text-align` (`center` / `right` / `justify`)
 - Color -> `color`
 - Decoración -> `text-decoration` (`none` / `underline` / `line-through` / ...)
 - Estilo -> `font-style` (`normal` / `italic` / `oblique`)
 - Peso -> `font-weight` (`normal` / `bold` / `lighter` /...)
 - Interlineado -> `line-height` (se recomienda entre 1.2em y 1.5em)
 - Espaciado -> `letter-spacing: XXpx;`
 - Versalitas -> `Font-variant: small-caps`
 - Sangría -> `text-indent`
 - Capitular: o bien creando un estilo propio y una etiqueta SPAN, o bien utilizando la subclase “:first-letter”;

Más información y recursos:

http://www.w3schools.com/css/css_text.as

p

http://www.w3schools.com/css/css_font.as

p

<http://www.font-combinator.com/>

<http://pxtoem.com/>

<http://letteringjs.com/>



CODIFICACIÓN DE TEXTOS

RECONOCIMIENTO ÓPTICO DE CARACTERES (OCR), SUBTITULADO Y CODIFICACIÓN



SUBTITULADO ELECTRÓNICO

- Los subtítulos son los textos que aparecen en el borde inferior de una pantalla, con frecuencia sobre el puesto a ella, adoptando información adicional sobre la misma o traduciendo una narración o diálogo conducido en un idioma extranjero.
- Hay que distinguir entre subtítulos incrustados en la imagen y los que son flotantes (se pueden activar y desactivar). Estos últimos pueden incorporarse de dos formas:
 - Contenedores multimedia (DVD y SVCD que los almacenan como imágenes, o AVUI MP4, OGM, Matroska que los almacenan como texto)
 - Archivos de texto plano que acompañan al archivo de vídeo (con el mismo nombre)
- Formatos de texto plano
 - SubRip (.srt)
 - MicroDVD (.sub)
 - Universal Subtitle Format (basado en xml)
 - Substation Alpha (.ssa)
 - Otros (.smi, .rt, .txt, .aqt, .jss, .js)
- Existe una amplia gama de [software de edición de subtítulos](#)

COMPATIBILIDAD E INTERCAMBIO DE FUENTES ENTRE PLATAFORMAS

- La codificación de caracteres es el método que permite convertir un carácter de un lenguaje natural en un símbolo de otro sistema de representación, como un número o una secuencia de pulsos eléctricos en un sistema electrónico, aplicando normas o reglas de codificación
- Hay que tener en cuenta que cada idioma tiene sus propios caracteres (o glifos)
- Normas de codificación:
 - ASCII
 - Limitado a 128 caracteres (7 dígitos binarios)
 - El ASCII Extendido subió a 8 dígitos, pero no era suficiente para dar cobertura a múltiples idiomas.
 - Está recogido en el conjunto de estándares ISO/IEC 8859
 - UNICODE
 - Incluye todos los caracteres de uso común en la actualidad (más de cien mil), tanto de idiomas actuales como de escrituras antiguas.
 - Es ideal para páginas web multi-idioma
 - El nuevo formato tipográfico Open Type, desarrollado por Microsoft y Adobe, y heredera del Tru Type, utiliza este sistema de codificación
 - Según su arquitectura (nº de bits utilizados para representar cada caracter) podemos hablar de 3 formas de codificación:
 - UTF-8
 - UTF-16
 - UTF-32
- En html se establece mediante la etiqueta charset dentro de los metadatos del encabezado
 - `<meta charset="UTF-8" />`