- IFCD0210 - Desarrollo de aplicaciones con tecnologías Web.
- Cualificación profesional que se obtiene al superar el curso: IFC154_3
- Se compone de tres módulos formativos o unidades de competencia:
    - UC0491_3: Desarrollar elementos software en el entorno cliente.
    - UC0492_3: Desarrollar elementos software en el entorno servidor.
    - UC0493_3: Implementar, verificar y documentar aplicaciones web en entornos internet, intranet y extranet.
- **Qué competencias aporta:** Desarrollar documentos y componentes software que constituyan aplicaciones informáticas en entornos distribuidos utilizando tecnologías web, partiendo de un diseño técnico ya elaborado, realizando, además, la verificación, documentación e implantación de los mismos.
- **Dónde permite trabajar:** Empresas o entidades públicas o privadas de cualquier tamaño que disponen de infraestructura de redes intranet, internet o extranet, en el área de desarrollo del departamento de informática desempeñando su trabajo tanto por cuenta propia como por cuenta ajena. *Programadores de aplicaciones informáticas. Técnicos de la web. Programador web. Programador multimedia.*
- **Qué aprenderemos en el MF1, "Programación Web en el entorno cliente"**: Este módulo se divide en tres partes:
    - **UF1 - "Elaboración de documentos Web mediante lenguajes de marcas"**:
    - **UF2 - "Desarrollo y reutilización de componentes software multimedia mediante lenguajes de guion":**
    - **UF3 - Aplicación de técnicas de usabilidad y accesibilidad en el entorno cliente:**
- **Qué aprenderemos en el MF2, "Programación Web en el entorno servidor":** Este módulo se divide en tres partes:
    - **UF1 - "Desarrollo de aplicaciones Web en el entorno servidor"**:
    - **UF2 - "Acceso a datos en aplicaciones Web del entorno servidor":**
    - **UF3 - Desarrollo de aplicaciones Web distribuidas:**
- **Qué aprenderemos en el MF3, "Implantación de aplicaciones Web en entornos internet, intranet y externet".** Sólo tiene una parte.

Principios de diseño Web

- **¿Qué es el diseño Web?** Consiste en la planificación, definición e implementación de sitios web. Requiere tener en cuenta la **navegabilidad** (facilidad con la que un usuario puede desplazarse por todas las páginas que componen un sitio web), **interactividad** (que permita una comunicación con el usuario a modo de diálogo), **usabilidad** (facilidad con que las personas pueden utilizar nuestro sitio Web con el fin de alcanzar un objetivo concreto), **arquitectura de la información** (disciplina y arte encargada del estudio, análisis, organización, disposición y estructuración de la información en espacios de información, y de la selección y presentación de los datos en los sistemas de información interactivos y no interactivos.) y la **interacción de medios** como el audio, texto, imagen, enlaces y vídeo.
- Vamos a estudiar diferentes formas de afrontar el diseño de un sitio Web.

Diseño orientado al usuario

- **¿Qué es el diseño orientado al usuario?** Es una filosofía de diseño que tiene por objetivo la creación de productos (en nuestro caso productos Web) que resuelvan

necesidades concretas de sus usuarios finales, consiguiendo la mayor satisfacción y mejor experiencia de uso posible con el mínimo esfuerzo de su parte.

- **Un diseño orientado al usuario supone tres etapas:**
  - Conocer a fondo a los usuarios finales, normalmente usando investigación cualitativa o investigación cuantitativa. En esta etapa se realizará una **recopilación de datos** por ejemplo mediante entrevistas directas con el cliente, pero también con el estudio de toda la información de la que dispongamos del cliente.
  - Diseñar un producto que resuelva sus necesidades y se ajuste a sus capacidades, expectativas y motivaciones. Una vez **analizados los datos recogidos** en la etapa anterior se planifica, se diseña y se fabrica el producto. No olvides que **los usuarios tienen modelos mentales muy asentados** y si recurres a soluciones comunes, la gente comprenderá mucho antes el funcionamiento de tu página. Algunos elementos comunes son:
    1. Logotipo de la empresa en la esquina superior izquierda, enlazando a la página de inicio.
    2. Caja de búsquedas en la esquina superior derecha, compuesta por una caja de texto y un botón con la leyenda "Buscar".
    3. Accesos rápidos en la parte superior de la página con secciones del tipo "Contacta", "Ayuda", "Tu cuenta", "Cesta de la compra".
    4. Debajo de estos tres elementos, una barra horizontal con enlaces a las secciones más importantes de la web (navegación global)
    5. Menú vertical en la parte izquierda para la navegación local dentro de cada sección.
    6. Enlaces de color azul, subrayados y que cambien de color cuando ya están visitados.

    Se seguirán las siguientes etapas en el diseño:

    1. **Diseño conceptual:** definir el esquema de organización, funcionamiento y navegación del sitio. No se especifica qué apariencia va a tener el sitio, sino que se centra en el concepto mismo del sitio: su arquitectura de información.
    2. **Diseño visual y definición de estilo:** En esta fase se especifica el aspecto visual del sitio web: composición de cada tipo de página, aspecto y comportamiento de los elementos de interacción y presentación de elementos multimedia.
    3. **Diseño de contenidos:** En el diseño de contenidos hipermedia se debe mantener un equilibrio entre lo que serían contenidos que no aprovechasen las nuevas posibilidades hipertexto y multimedia, y lo que serían contenidos caóticos o desorientativos debido a un uso excesivo y no sosegado de las posibilidades hipermedia. Sin prescindir de las capacidades que ofrece el nuevo medio, de lo que se trata es de diseñar contenidos interrelacionados y vinculados, manteniendo cierta coherencia informativa, comunicacional y organizativa.
    4. **Prototipado:** La evaluación de la usabilidad del sitio web se debe realizar desde las primeras etapas de diseño, pero ¿cómo evaluar un sitio web que no está implementado? A través de prototipos. La etapa de prototipado se basa en la elaboración de modelos o prototipos de

la interfaz del sitio. Su aspecto no se corresponde exactamente con el que tendrá el sitio una vez finalizado, pero pueden servir para evaluar la usabilidad del sitio sin necesidad de esperar a su implementación.

5. **Evaluación:** La evaluación de la usabilidad es la etapa más importante en el proceso de Diseño Centrado en el Usuario.
    - **Método por inspección: evaluación heurística**
      Los métodos de inspección de la usabilidad de un sitio web son aquellos realizados por el experto en usabilidad, y que se basan en el recorrido y análisis del sitio identificando errores y problemas de diseño. La Evaluación Heurística es un tipo de método de inspección, que tiene como ventaja la facilidad y rapidez con la que se puede llevar a cabo. Un modelo de evaluación heurística podría ser:
        - **Aspectos generales:** Objetivos, apariencia y aspecto, coherencia y nivel de actualización de contenidos.
        - **Identidad e Información:** Identidad del sitio e información proporcionada sobre el proveedor y la autoría de los contenidos.
        - **Lenguaje y redacción:** Calidad de los contenidos textuales.
        - **Rotulado:** Significación y familiaridad del rotulado de los contenidos.
        - **Estructura y Navegación:** Idoneidad de la arquitectura de información y navegación del sitio.
        - **Esquema de la página:** Distribución y aspecto de los elementos de navegación e información en la interfaz.
        - **Búsqueda:** Buscador interno del sitio.
        - **Elementos multimedia:** Grado de adecuación de los contenidos multimedia al medio web.
        - **Ayuda:** Documentación y ayuda contextual ofrecida al usuario para la navegación.
        - **Accesibilidad:** Cumplimiento de directrices de accesibilidad.
        - **Control y retroalimentación:** Libertad del usuario en la navegación.

- Poner a prueba lo diseñado, normalmente usando test de usuarios.
    1. El test con usuarios es una prueba de usabilidad que se basa en la observación y análisis de cómo un grupo de usuarios reales utiliza el sitio web, anotando los problemas de uso con los que se encuentran para poder solucionarlos posteriormente.
    2. Como toda evaluación de usabilidad, cuanto más esperamos para su realización, más costoso resultará la reparación de los errores de diseño descubiertos. Esto quiere decir que no sólo debemos realizar este tipo de pruebas sobre el sitio web una vez implementado, sino también, sobre los prototipos del sitio.
    3. Es una prueba complementaria a la evaluación heurística, pero un test con usuarios es más costoso, por lo que es recomendable realizarlo siempre después de una evaluación heurística, ya que sería

desperdiciar tiempo y dinero utilizarlo para descubrir errores de diseño motivados por el no cumplimiento en el desarrollo de principios generales de usabilidad (heurísticos).

4. La ventaja que ofrecen los test de usuarios frente a otro tipo de evaluaciones es que por un lado es una demostración con hechos, por lo que sus resultados son más fiables, y por otro porque posibilitan el descubrimiento de errores de diseño imposibles o difíciles de descubrir mediante la evaluación heurística.

Diseño orientado a objetivos

- Otra forma de enfocar el diseño de un sitio Web es orientarlo a los objetivos que se desean obtener con el desarrollo del mismo. Por ejemplo:
    - Objetivos económicos: vender un producto nuevo o mejorar las ventas de un producto ya existente.
    - Concienciar a la población sobre un problema social.
- Difusión cultural, científica o tecnológica.
  Las etapas de un diseño orientado a objetivos serían equivalentes a los diseños orientados a usuario, introduciendo además de los elementos necesarios para mejorar la experiencia del usuario, los elementos necesarios para la consecución de los objetivos.

Diseño orientado a la implementación

- **Definición de implementación:** Una implementación es la instalación de una aplicación informática, realización o la ejecución de un plan, idea, modelo científico, diseño, especificación, estándar, algoritmo o política.
- Es decir, tendremos en cuenta en el diseño de nuestro sitio Web las restricciones de implementación a las que podemos estar sometidos. Por ejemplo:
    - Limitaciones económicas. Presupuesto del que disponemos para realizar el diseño y desarrollo de nuestro sitio Web.
    - Plataformas en las que se va a distribuir nuestro sitio Web: ¿está pensado para su uso sólo en ordenadores conectados a internet? ¿va a utilizarse en dispositivos móviles o *tablet*? ¿va a utilizarse sólo en expositores?
- Las etapas de un diseño orientado a objetivos vuelven a ser equivalentes a los diseños orientados a usuario, introduciendo además de los elementos necesarios para mejorar la experiencia del usuario, los elementos necesarios para la consecución de los objetivos previstos para el sitio y los problemas o requerimientos específicos de implementación de los mismos..

Principios de un buen diseño Web:

Ver la presentación:

http://www.slideshare.net/conectmx/principios-de-un-buen-diseo-web

El proceso de diseño Web

Vamos a estudiar con algo más de detalle las etapas del proceso de diseño Web.

Estructura de un sitio Web y navegabilidad

- El diseño de un sitio Web nos llevará a tener un conocimiento de qué contenidos vamos a mostrar en él y de cuáles serán las directrices de diseño a seguir en su programación.
- El siguiente paso será la **definición de la estructura del sitio Web:** Una vez que se ha hecho el trabajo de identificación de contenidos, se debe avanzar hacia las definiciones relacionadas con la forma que tendrá el sitio que se está desarrollando. Ello implicará trabajar en tres áreas concretas, a través de las cuales se definirá la estructura del sitio, el árbol de contenidos y los sistemas de navegación que se ofrecerá a los usuarios para que avancen a través de sus contenidos. Las tres áreas mencionadas son:
    - **Creación de la Estructura:** Se refiere al proceso de identificar la forma que tendrá el Sitio Web que se está desarrollando. En este sentido es importante hacer una diferencia entre estructura y diseño.
        1. **Estructura:** se refiere a la forma que tendrá el Sitio Web en términos generales con sus secciones, funcionalidades y sistemas de navegación. No considera ni incluye elementos gráficos (logotipos, viñetas, etc.).
        2. **Diseño:** se refiere a la solución gráfica que se creará para el sitio, en la cual aparecen colores, logotipos, viñetas, y otros elementos de diseño que permiten identificar visualmente al sitio.

      Dado lo anterior, cuando hablamos de la estructura nos estamos refiriendo básicamente a cuál será la experiencia que tendrá un usuario cuando accede al sitio. De esta manera podremos determinar dónde estarán ubicados los servicios interactivos (buscador, sistemas de encuestas, áreas de contenidos).

      Gracias a la realización de esta etapa es posible discutir en términos muy prácticos cuál será la oferta de elementos de información e interacción que tendrá el usuario. Al no incluir elementos de diseño, se permite que la discusión sobre la estructura se desarrolle en aspectos concretos, sin que intervengan aún consideraciones estéticas que habitualmente atrasan la aprobación de esta etapa del desarrollo.

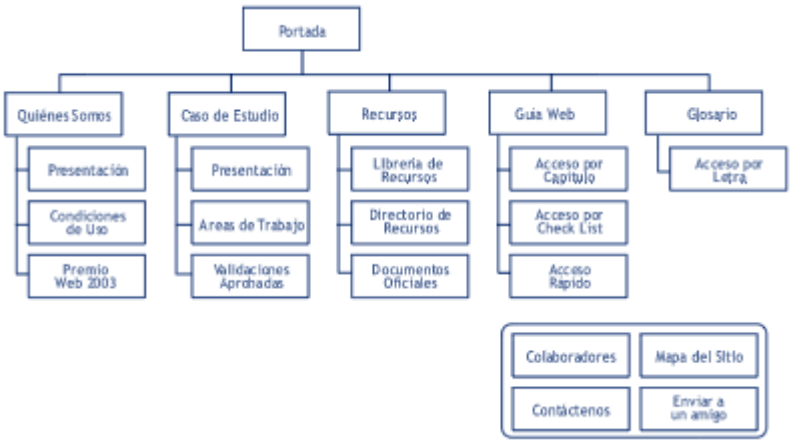
    - **Mapas Permanentes del Sitio:** Se refiere al proceso de crear un árbol de contenido en el que se muestre de manera práctica cuántas secciones tendrá el sitio en desarrollo y cuántos niveles habrá dentro de cada uno.

      Cuando se usa la idea de crear un árbol, se refiere exactamente a generar un diagrama que cuente con un tronco, ramas y hojas, para mostrar las zonas principales, secundarias y contenidos finales que se irán incorporando.

      En este sentido se debe evitar a toda costa que el árbol de contenidos represente la estructura de la organización, dado que ésta es conocida y comprendida internamente, pero constituye una barrera de entrada para usuarios externos.

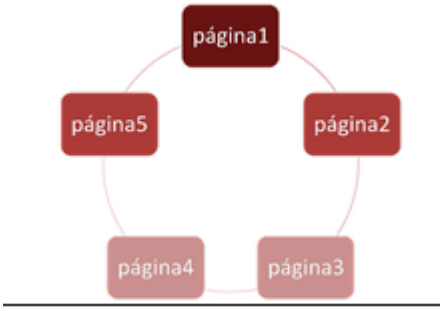
Las formas más comunes de definir estos mapas son:

Estructura jerárquica o de árbol:



Estructura lineal:

Estructura en red:



La estructura más habitual es la de árbol. Las recomendaciones para la generación de este árbol son las siguientes:

- **Secciones:** se debe intentar que sean las menos posibles, con el fin de concentrar las acciones del usuario en pocas áreas; hay que considerar que cada una de las áreas a integrar en el árbol requerirá de mantenimiento posterior en contenidos, gráfica y funcionalidad, lo que encarecerá el costo final de operación del sitio. Dado lo anterior, se recomienda que las secciones se sitúen entre 5 y 7.
- **Niveles:** se debe intentar que el usuario esté siempre a menos de tres clic del contenido que anda buscando. Por ello no se debería

crear más de tres niveles de acceso; esto significa una Portada, una Portadilla de Sección y los Contenidos propiamente dichos.

- **Contenidos relacionados:** se debe considerar que habrá funcionalidades que estén presentes en todo el sitio. Entre ellas se incluyen elementos como Buscador, Preguntas Frecuentes y Formularios de Contacto. Se recomienda que este tipo de elementos quede fuera del árbol y floten sobre éste, con el fin de indicar que desde todas las páginas habrá enlaces a ellos.

- **Definición de los Sistemas de Navegación:** Una vez que se cuenta con los árboles de contenido desarrollados en el paso anterior, la tarea siguiente consiste en generar los sistemas de acceso a dichos contenidos en el Sitio Web. A través de estos, los usuarios podrán avanzar por sus diferentes áreas, sin perderse.

  En la generación de dichos sistemas se debe atender a dos elementos que serán muy importantes:

  1. **Textual:** se refiere a que la navegación se hará a través de elementos concretos, tales como menús, guías, botones y otros elementos que deben ser claramente distinguibles dentro de la interfaz. Para generarlos se debe conseguir que cada uno de ellos represente claramente la función para la que fueron designados y no dejar lugar a dudas sobre su función ni sobre la acción que desarrollarán al ser usados. Es decir, un botón debe parecer tal y no sólo un parche de color sobre la pantalla. Adicionalmente, es muy importante que las palabras escogidas para indicar acciones (etiquetado de menús), sean claras y precisas. En este sentido, si un botón necesita ser explicado, es mejor desecharlo y buscar otra solución.

  2. **Contextual:** es todo lo referido a cómo se presenta la información, utilizando para ellos elementos basados en texto o gráficos. Los elementos relevantes en este caso, serán todos aquellos que permiten mostrar la navegación en la pantalla y ayudar al usuario en el contexto del contenido que está viendo. Entre ellos, la gráfica utilizada, la redacción de los textos que se muestran, los contenidos relacionados, las nubes de etiquetas e incluso el nombre del dominio (URL) que permitirá que el usuario sienta que está en el lugar indicado.

  Al generar el sistema de navegación, se deben tener en cuenta las siguientes características:

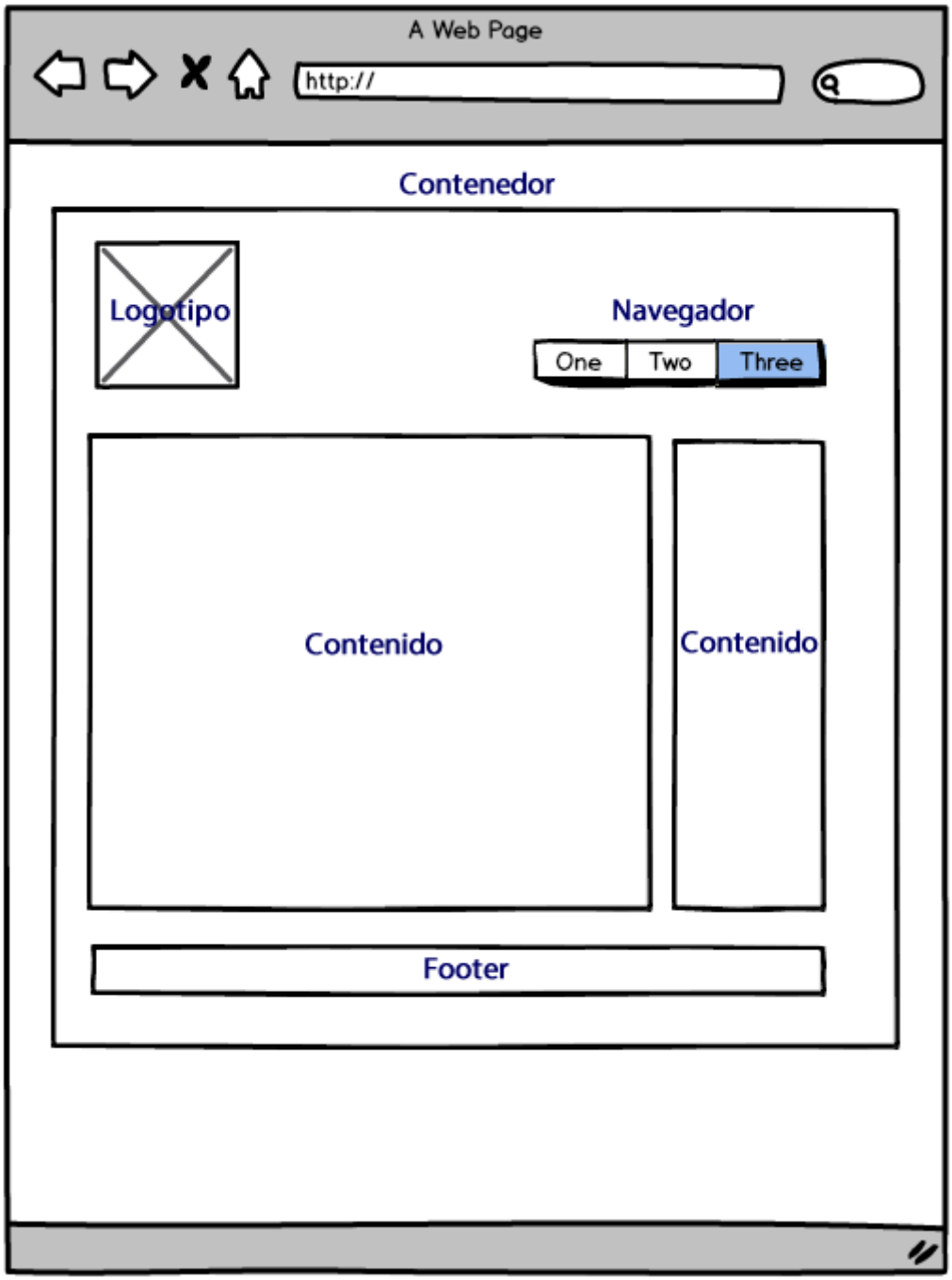
  - **Consistente:** el sistema debe ser similar en todo el sitio, en lo referido a su ubicación y disposición en las páginas. Esto se aplica también a aquellas instituciones que pueden tener más de un Sitio Web.

  - **Uniforme:** el sistema debe utilizar similares términos con el fin de que el usuario que lo vea en las páginas, confíe en que sus opciones llevan siempre hacia los mismos lugares dentro del sitio.

- **Visible:** el sistema debe distinguirse claramente dentro del sitio, con el fin de que el usuario cuente con él, como si se tratara de una guía permanente en el área en que se encuentre del sitio.

Estructura y composición de páginas.

- Una vez hemos definido la estructura global del sitio, podemos abordar la estructura y composición de las páginas que lo componen:
- Hay miles de formas diferentes de estructurar una página, pero son pocas las que funcionan y adquieren el sentido que necesitamos para nuestro trabajo.



- **Contenedor:** Esto es lo que definimos en la regla <Body> o en algún de nuestra estructura. Dentro de este contenedor estarán todos los elementos del sitio; módulos, contenidos, imágenes, etc.

  Este contenedor puede tener un ancho fijo; el ancho será igual para todos los navegadores y dispositivos, o uno fluido que se adaptará al ancho de nuestra ventana. Si usamos web responsive, podemos manejarlo con CSS sin problemas.

- **Logotipo:** Acá es dónde debemos dejar la imagen de la empresa, que identificará la marca en todo el sitio. Puede ser el logotipo de la empresa o el nombre. Al estar arriba acompañara la navegación por todo el sitio ayudando al reconocimiento de la marca.
- **Navegación:** La forma más aceptada y fácil, es el menú horizontal, que siempre tiene a mano lo más importante del sitio web.

  Los menús verticales pueden usarse como complementos o sub menús, de los horizontales, y siempre cerca de la línea horizontal del menú principal y deben ser visibles en el primer pantallazo, al entrar.

- **Contenido:** Lo más importante y fundamental de una buena página. Lo que mantiene al visitante cautivo e interesado. Si el contenido es malo o poco interesante, abandonará la página en pocos segundos. Para eso, debe estar centralizado y enfocado y a la primera vista. El mejor lugar debe ser para los contenidos. Para Google, el contenido es el rey.
- **Footer o pie:** Localizado al final del sitio, abajo, generalmente dejamos ahí información de Copyright y legales o menús secundarios. Aunque en los últimos años, se usa mucho para incluir información de redes sociales, direcciones, noticias.
- **Espacio Negativo:** Esto es tan importante como el espacio positivo. También llamado espacio blanco, es todo lo que no tiene información de ningún tipo, pero que cumple un rol importantísimo para la web; nos ayuda a tener espacios de respiración, a equilibrar y dar balance a todo el diseño. Si eres novato, o tu cliente es ignorante en la materia, tratará de llenar esos espacios, con la consiguiente saturación de contenidos, dejando una web ilegible e innavegable.

Compatibilidad con navegadores

- Hay que tener en cuenta que los visitantes de nuestros sitios Web utilizarán **distintos navegadores, tanto a nivel de fabricante como a nivel de versión**. Por lo tanto, hay que tener en cuenta el concepto de "compatibilidad de navegadores" durante el desarrollo de nuestros sitos Web.
- **Compatibilidad de navegadores:** Normalmente, los usuarios acceden a tu sitio web a través de un navegador. Cada navegador interpreta el código del sitio de manera ligeramente distinta, por lo que es posible que se muestre de forma diferente en función del navegador que emplee el usuario. Por lo general, es aconsejable no confiar en las funciones específicas de los navegadores. Para asegurarnos de que nuestro sitio no se comporte de forma inesperada, podemos seguir los pasos que se indican a continuación:
  - Probar el sitio en el mayor número de navegadores posible: Cuando hayamos diseñado el sitio, debemos revisar su aspecto y funcionalidad en varios navegadores para asegurarnos de que todos los usuarios puedan acceder a él tal y como lo diseñamos. Sería conveniente comenzar a realizar pruebas durante el proceso de desarrollo del sitio lo antes posible. Cada navegador, e

incluso las distintas versiones de un mismo navegador, podrían mostrar el sitio de forma diferente. Podemos utilizar servicios como Google Analytics para saber cuáles son los navegadores que más utilizan los usuarios para acceder a nuestro sitio.

- **Utilizar una codificación HTML clara y válida:** Es posible que nuestro sitio se muestre correctamente en algunos navegadores aunque el lenguaje HTML no sea válido, pero ello no nos garantiza que así sea en todos los navegadores (ni en navegadores que aparezcan más adelante). La mejor forma de asegurarnos de que nuestra página tenga la misma apariencia en todos los navegadores es redactarla con una codificación HTML y CSS válida y, a continuación, probarla en el mayor número de navegadores posible.

  Si además utilizamos CSS, que diferencia la presentación del contenido, las páginas se cargarán y se procesarán más rápidamente. Las herramientas de validación como, por ejemplo los validadores gratuitos que hay en línea para HTML y para CSS proporcionado por W3 Consortium, son útiles para comprobar el sitio. (**http://validator.w3.org/**).

- **Especificar la codificación de caracteres:** Para ayudar a los navegadores a mostrar el texto de tu página, especificaremos siempre una codificación para el documento. Esta codificación debe aparecer en la parte superior del documento, ya que algunos navegadores no reconocen declaraciones de conjuntos de caracteres incluidas en el cuerpo del mismo.

- **Tener en cuenta la accesibilidad:** Quizás no todos los usuarios tengan habilitado JavaScript en sus navegadores. Por otro lado, es posible que tecnologías como Flash o ActiveX no se ejecuten de forma correcta en todos los navegadores o que sencillamente no se ejecuten. Debemos proporcionar alternativas de solo texto para los contenidos y las funciones de soportes interactivos. Esto hará también nuestros sitios más accesibles para los usuarios que utilicen tecnologías alternativas como son los lectores de pantalla.

Diferencia entre diseño orientado a presentación e impresión.

- Los diseños de nuestras páginas Web pueden ir dirigidos a su presentación por la pantalla del ordenador (o de una tablet, teléfono móvil, expositor o cualquier dispositivo similar) o puede estar dirigidos en un momento dado a obtener una copia impresa del contenido directamente en papel.
- Algunas de las diferencias básicas entre un diseño Web para presentación y un diseño Web para impresión pueden ser:
  - **Interactuar en vez de mostrar:** El diseño impreso está basado en ver, pero el diseño web está basado en hacer. La función del diseño tradicional es contar historias, la misión del diseño digital es entablar conversaciones.
  - **Espacio y entorno indefinido en vez de definido:** El diseño impreso se visualiza en espacio limitado pero el diseño web se tiene que poder visualizar en múltiples espacios. Además el diseño web se puede extender fuera de la pantalla (podemos que utilizar barras de desplazamiento o scrolling).

    Cuando se diseña para imprimir, el resultado va a ser exactamente igual o muy parecido al original que hemos diseñado y guardado, en cambio, cuando se diseña una web, debemos considerar factores externos de los cuales no

tenemos ningún control (el usuario puede tener resoluciones de pantalla diferente o utilizar un navegador web distinto, además de no tener las tipografías utilizadas,…).

- **Problema con las tipografías:** En el diseño impreso se puede utilizar cualquier tipografía, pero el diseño web normalmente solo muestra las tipografías que el usuario tenga instaladas en su ordenador.
- **Archivos pequeños en vez de grandes:** El diseño impreso se trabaja con altas resoluciones de archivos y se procura mantener la mejor calidad en las imágenes y originales, mientras que en el diseño web se trabaja con resoluciones bajas para ahorrar recursos del servidor y permitir que el usuario abra la página rápidamente. En diseño web, debemos optimizar las imágenes y animaciones, es decir, bajar la resolución lo más posible sin perder una calidad significativa. El peso (calculado en Kb o Mb) de una página web se calcula como la suma de todos pesos de los documentos que abre una página web (imágenes, archivos flash, animaciones...).

  Con las conexiones de internet actual podemos calcular muy orientativamente que por cada incremento de peso de 500Kb a 1Mb, una página tarda un segundo más a descargarse y abrirse.

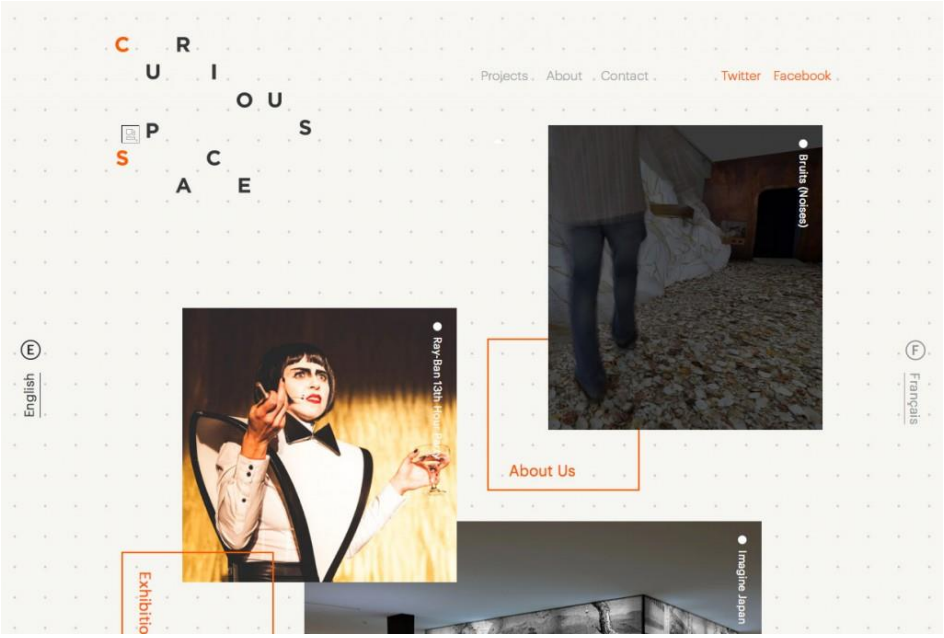
- **RGB en vez de CMYK:** En el diseño impreso se trabaja colores substractivos o CMYK, en cambio en el diseño web solo se trabaja con colores aditivos o RGB. El color es un error común de los diseñadores gráficos cuando quieren diseñar para web. En la escuela de diseño te enseñan a utilizar siempre CMYK para que la impresión quede bien, pero en el mundo web, es todo lo contrario, las pantallas emiten luz únicamente interpretan datos RGB o aditivos.

- Si pensamos que nuestras páginas Web pueden ir orientadas tanto a su presentación por pantalla como a impresión, podremos crear hojas de estilo diferentes para estos dos medios diferentes.

Tendencias de diseño Web 2016

1. Diseño fuera de cuadrícula

Con el diseño web responsive, hemos utilizado cuadrículas como si no hubiera un mañana. Dejando de lado la posibilidad del “desorden ordenado”, de cierta libertad creativa en el mundo web. Hemos llegado a un punto en que el que todas las webs consideradas “bonitas” responden a webs con grandes fotos en cuadrículas, consiguiendo asi que todo tenga el mismo aspecto.
Con el diseño fuera de cuadrícula, da igual que unas fotos floten sobre otras. O que al verla en móvil la disposición de estas respondan más a su buena visualización que en comparación al recurso estilístico visualizándolo en pantalla estandar.
Podéis ver un ejemplo en Curious space (http://www.curiousspace.com/).

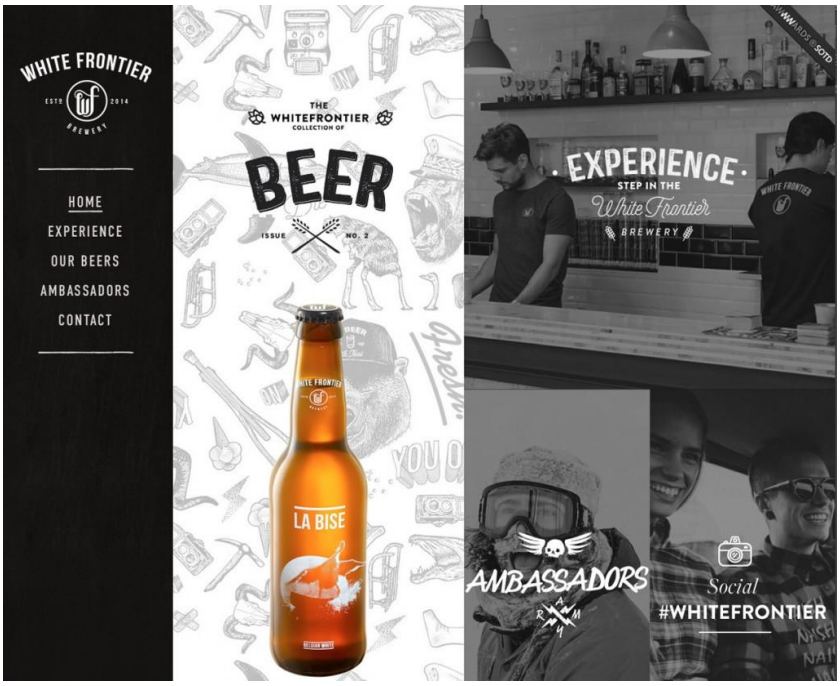


Curious Space

2. Uso de tipografías como recurso gráfico

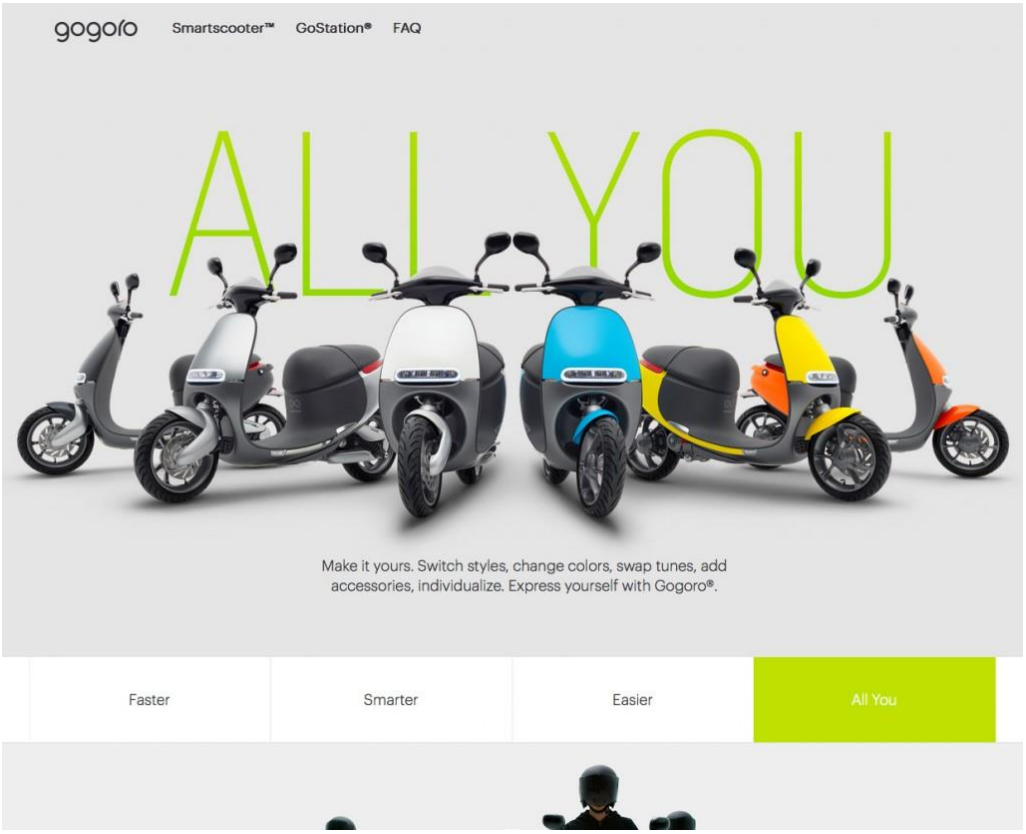
En 2016 promete ya que el catálogo de google fonts va creciendo día día.
Pero lo que más nos gusta es cómo cada vez se utilizan las tipografías más creativas y en algunos casos tienen un gran peso en la resolución de algunas páginas web.
Podemos ver estas tendencias:

- Tipografías grandes
- Tipos superpuestas a imágenes
- Tipografías customizadas
- Uso de tipografías artísticas en la web



White Frontier

Jeff Bridges



Gogoro Scooter

3. Contar historias a través de la web o el uso del “Storytelling”.

El acto de narrar viene de tiempos inmemoriables, y es un hecho que cualquier campaña que nos cuente una historia, será más fácil de asimilar y empatizar que campañas con otro tipo de impactos publicitarios.

Navega por los ejemplos, en el primer ejemplo podrás conocer la historia de las olimpiadas y en el segundoel mundo de diseño en el ámbito infantil.

Historia de los juegos olímpicos

Historia de la infancia del MoMA

4. Continuación del uso del diseño Semi-flat
Desde que windows, sacó su estilo “Metro” en 2013, la tendencia al flat desing se ha mantenido aún con sus problemas de usabilidad.
De ahi a que surgiera una tendencia intermedia o semi-flat que seguirá durante el 2016 en tendencia.

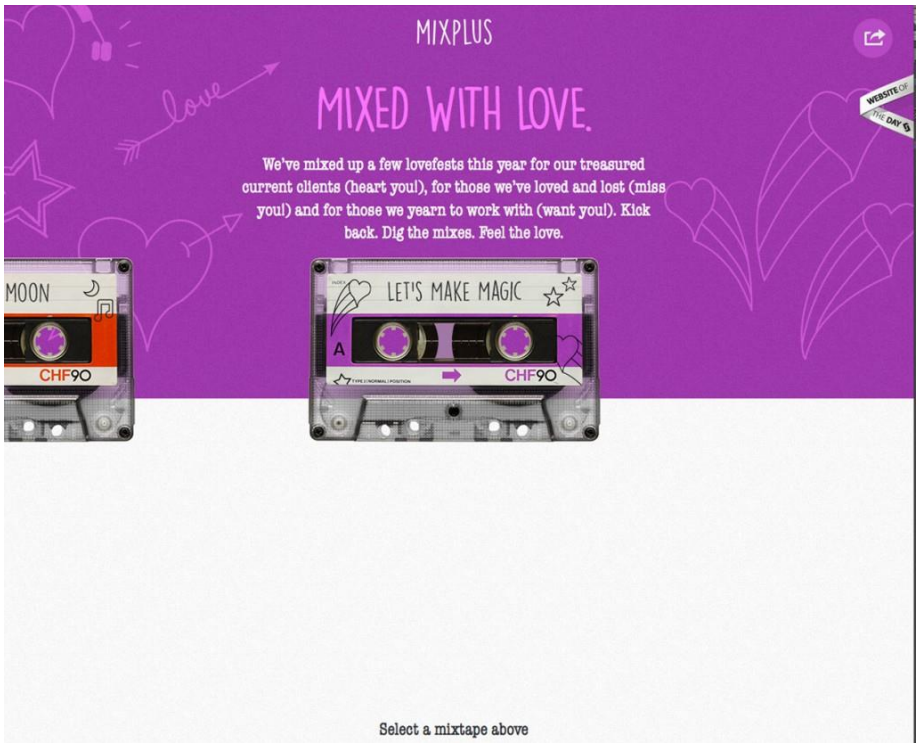
5. Continuación del RWD (responsive web design)
Como ya te hemos dicho más de una vez si tu web no es responsive te estás quedando atrás. De hecho desde el pasado 21 de abril google no abre páginas en web móvil que no sean responsive.
En 2016 seguiremos haciendo webs responsive porque seguirá siendo tendencia.

6. Usos creativos del color
Veremos el uso de **colores brillantes pero no como protagonistas**, sino acentuando detalles, una de las tendencias que vienen y nos encanta es la **tendencia de los monocromáticos**, con el fin de utilizar la misma tonalidad de color para evitar distracciones.

Line Equality

7. El poder de la interactividad

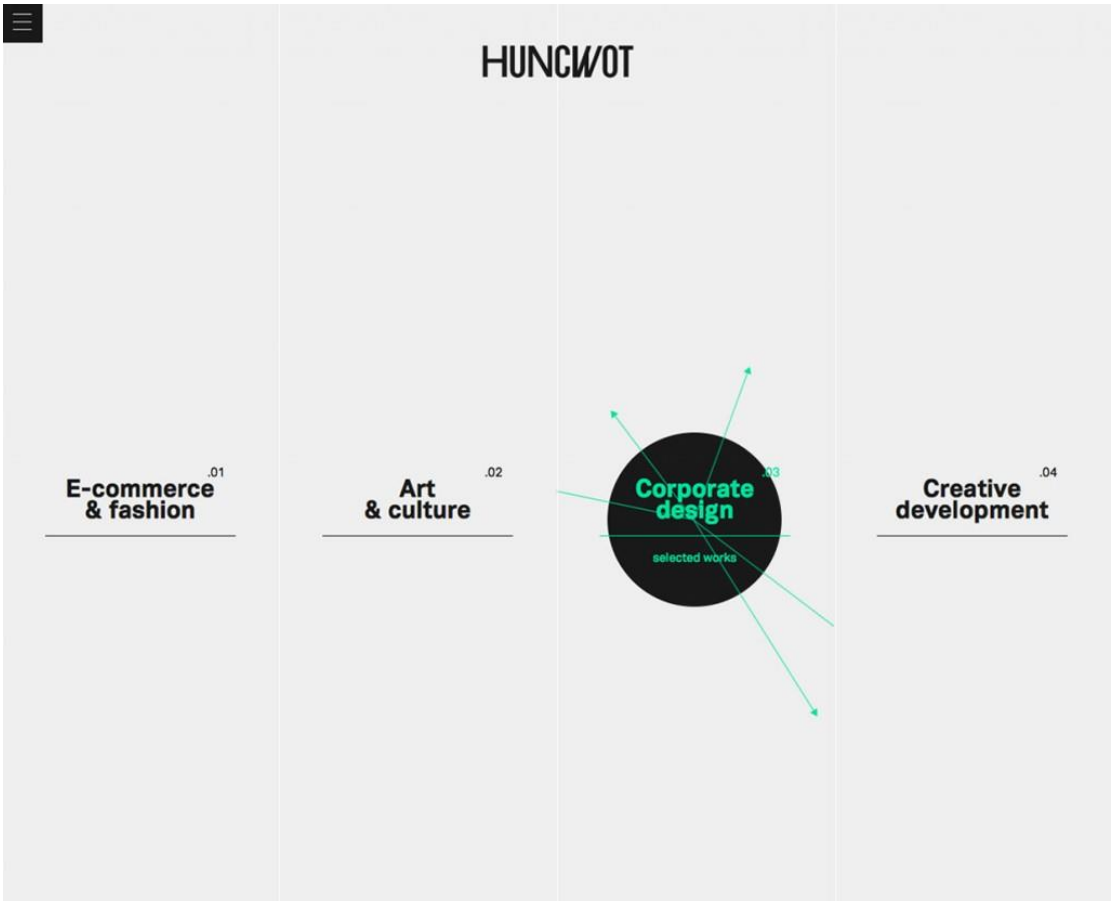
Ya no sólo navegaremos, podremos interactuar con la web, adaptarla a nuestros gustos. El poder de la interactividad será mayor y la tecnología en 2016 mejorará notablemente.



Mixed with love

8. Minimalismo

Arraigado en el movimiento artístico posterior a la Segunda Guerra Mundial, el minimalismo ha vuelto a emerger como una poderosa técnica en el diseño web moderno. Encarnando el concepto de “menos es más”, los sitios minimalistas utilizan espacios negativos, tipografías sin adornos, y los elementos dispersos. En 2016 veremos mucho esta moderna tendencia.

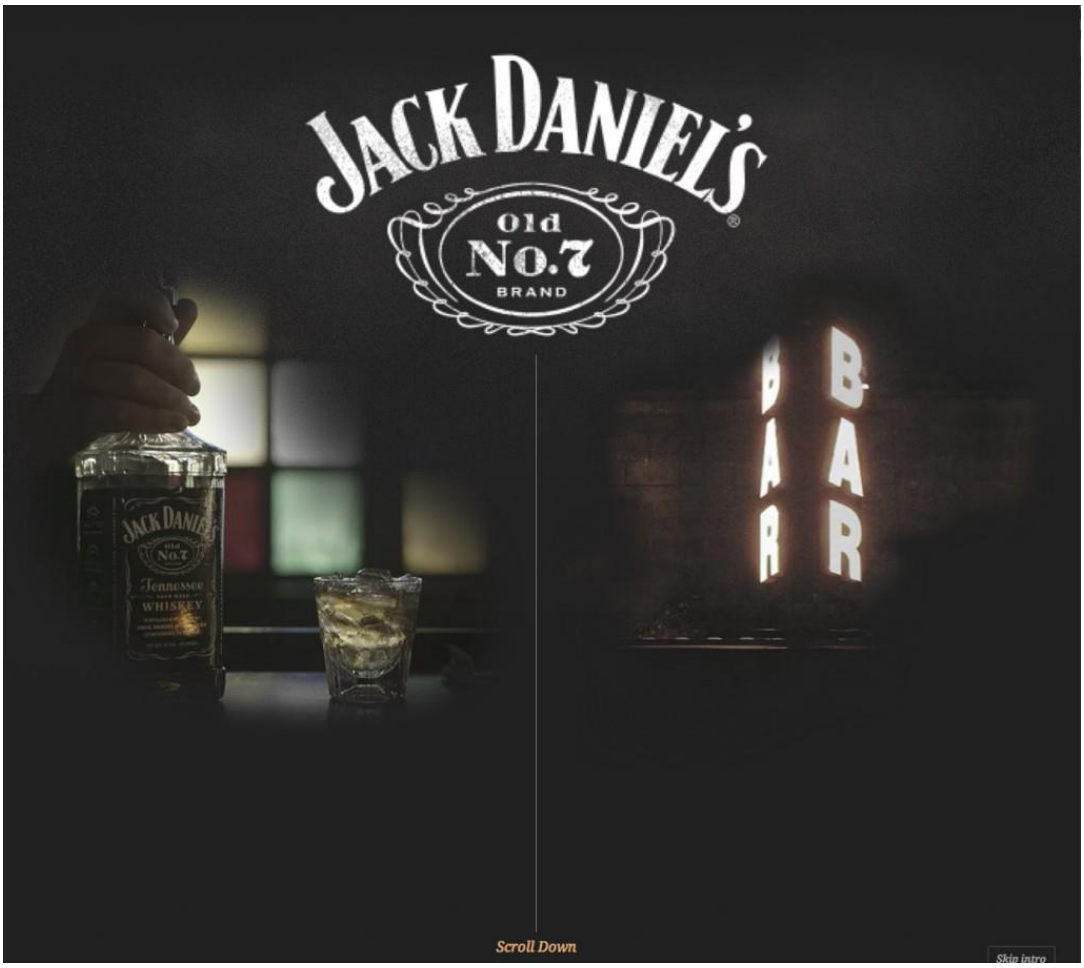


Huncwot

9. Bienvenidos scrolls interminables

Los scrolls se harán más largos que un día sin pan, impulsado este recurso por la dominación del uso de móvil y tablets.

En pocas palabras, cuanto menor sea la pantalla, mayor será el desplazamiento. Esto supondrá un reto a la hora diseñar y ser más creativos planteando técnicas narrativas.



Bar Jack Daniels

Origen de los lenguajes de marcado generales: SGML y XML.

- **Historia:** Los lenguajes de marcas se llaman así por la práctica tradicional de marcar los manuscritos con instrucciones de impresión en los márgenes. En la época de la imprenta, esta tarea ha correspondido a los marcadores, que indicaban el tipo de letra, el estilo y el tamaño, así como la corrección de errores, para que otras personas compusieran la tipografía. Esto condujo a la creación de un grupo de marcas estandarizadas. Con la introducción de las computadoras, se trasladó un concepto similar al mundo de la informática.
- Orígenes: El concepto de lenguaje de marcas fue expuesto por vez primera por William W. Tunnicliffe en 1967.2 La mayor novedad consistía en la separación entre la presentación y la estructura del texto. Tunnicliffe dirigiría más tarde el desarrollo de un estándar al que bautizaría como GenCode, destinado a la industria editorial. El editor Stanley Fish también expuso ideas similares a finales de los años 1960. Brian Reid, en su disertación de 1980 en la Carnegie Mellon University, mostró su teoría y una implementación práctica de un lenguaje descriptivo todavía en uso.

  Sin embargo, quien es considerado el padre de los lenguajes de marcas es Charles Goldfarb, investigador de la compañía IBM. Goldfarb participó en la creación del lenguaje GML, y posteriormente dirigió el comité que elaboró el estándar SGML, la piedra angular de los lenguajes de marcas. En cualquier caso, y a pesar de las

controversias sobre su origen, es comúnmente aceptado que la idea surgió de forma independiente varias veces durante los 70, y que se generalizó en los 80.

- **Los lenguajes primitivos:** El primer lenguaje que diferenció claramente la estructura de la presentación fue ciertamente el Scribe, desarrollado por Brian Reid y descrito en 1980 en su tesis doctoral.

  Otro de los principales estándares de publicación es TeX, creado y mantenido por Donald Knuth en los años 70 y 80. TeX se centra en la estructura detallada del texto y la descripción de las fuentes, fundamentalmente en el campo de las publicaciones matemáticas especializadas.

  Al margen de la industria editorial también surgieron algunas iniciativas, como los lenguajes troff y nroff, lenguajes utilizados para maquetación en sistemas UNIX.

- **La generalización de los lenguajes de marcas:** La iniciativa que sentaría las bases de los actuales lenguajes, partiría de la empresa IBM, que buscaba nuevas soluciones para mantener grandes cantidades de documentos. El trabajo fue encomendado a Charles F. Goldfarb, que junto con Edward Mosher y Raymond Lorie, diseñó el Generalized Markup Language o GML. Este lenguaje heredó del proyecto GenCode la idea de que la presentación debe separarse del contenido. El marcado, por tanto, se centra en definir la estructura del texto y no su presentación visual.

  El lenguaje GML fue un gran éxito y pronto se extendió a otros ámbitos, siendo adoptado por el gobierno de Estados Unidos, con lo que surgió la necesidad de estandarizarlo. A principios de los años 1980 se constituyó un comité dirigido por Goldfarb. Tras un largo proceso, en 1986 la Organización Internacional para la Estandarización publicaría el Standard Generalized Markup Languaje con rango de Estándar Internacional con el código ISO 8879.

  El SGML especifica la sintaxis para la inclusión de marcas en los textos, así como la sintaxis del documento que especifica qué etiquetas están permitidas y donde: el Document Type Definition o schema. Esto permitía que un autor emplease cualquier marca que quisiera, eligiendo nombres para las etiquetas que tuvieran sentido tanto por el tema del documento como por el idioma. Así, el SGML es, estrictamente hablando, un metalenguaje, del que se derivan varios lenguajes especializados.

- **La popularización: el HTML:** En 1991, parecía que los editores WYSIWYG (que almacenan los documentos en formatos binarios propietarios) abarcarían casi la totalidad del procesamiento de textos, relegando al SGML a usos profesionales o industriales muy específicos. Sin embargo, la situación cambió drásticamente cuando Sir Tim Berners-Lee, que había aprendido SGML de su compañero en el CERN Anders Berglund, utilizó la sintaxis SGML para crear el HTML.

  Este lenguaje era similar a cualquier otro creado a partir del SGML, sin embargo resultó extraordinariamente sencillo, tanto que el DTD no se desarrolló hasta más tarde.

  El HTML es hoy día el tipo de documento más empleado en el mundo. Su sencillez era tal que cualquier persona podía escribir documentos en este formato, sin apenas necesidad de conocimientos de informática. Esta fue una de las razones de su éxito, pero también condujo a un cierto caos. El crecimiento exponencial de la web en los años 90 produjo documentos en cantidades ingentes pero mal estructurados,

problema agravado aún más por la falta de respeto por los estándares, por parte de diseñadores web y fabricantes de software.

- **La madurez: el XML:** La respuesta a los problemas surgidos en torno al HTML vino de la mano del XML (eXtensible Markup Language). El XML es un meta-lenguaje que permite crear etiquetas adaptadas a las necesidades (de ahí lo de "extensible"). El estándar define cómo pueden ser esas etiquetas y qué se puede hacer con ellas. Es además especialmente estricto en cuanto a lo que está permitido y lo que no, todo documento debe cumplir dos condiciones: ser válido y estar bien formado.

  El XML fue desarrollado por el World Wide Web Consortium mediante un comité creado y dirigido por Jon Bosak. El objetivo principal era simplificar el SGML para adaptarlo a un campo muy preciso: documentos en internet.

  El XML fue ideado en principio para entornos semi-estructurados, como textos y publicaciones. Uno de los ejemplos más claros es el XHTML, la redefinición del HTML en clave XML, con las ventajas que ello supone. Sin embargo pronto se observó que sus virtudes podían ser útiles en campos bien distintos. Los lenguajes basados en XML tienen aplicaciones incontables, como en la transacción de datos entre servidores, intercambio de información financiera, fórmulas y reacciones químicas, y un largo etcétera.

- **La web semántica:** Los lenguajes de marcado son la herramienta fundamental en el diseño de la web semántica, aquella que no solo permite acceder a la información, sino que además define su significado, de forma que sea más fácil su procesamiento automático y se pueda reutilizar para distintas aplicaciones. Esto se consigue añadiendo datos adicionales a los documentos, por medio de dos lenguajes expresamente creados: el RDF (Resource descriptión framework-Plataforma de descripción de recursos) y OWL (Web Ontology Language-Lenguaje de ontologías para la web), ambos basados en XML.

Características generales de los lenguajes de marcado

- **Texto plano:** Una de las principales ventajas de este tipo de codificación es que puede ser interpretada directamente, dado que son archivos de texto plano. Esto es una ventaja evidente respecto a los sistemas de archivos binarios, que requieren siempre de un programa intermediario para trabajar con ellos. Un documento escrito con lenguajes de marcado puede ser editado por un usuario con un sencillo editor de textos, sin perjuicio de que se puedan utilizar programas más sofisticados que faciliten el trabajo.

  Al tratarse solamente de texto, los documentos son independientes de la plataforma, sistema operativo o programa con el que fueron creados. Esta fue una de las premisas de los creadores de GML en los años 70, para no añadir restricciones innecesarias al intercambio de información.

- **Compacidad** (cualidad de ser compacto): Las instrucciones de marcado se entremezclan con el propio contenido en un único archivo o flujo de datos. Por ejemplo: <h1>Título</h1> o texto.
- **Facilidad de procesamiento:** Las organizaciones de estándares han venido desarrollando lenguajes especializados para los tipos de documentos de comunidades

o industrias concretas. Uno de los primeros fue el CALS, utilizado por las fuerzas armadas de EE.UU. para sus manuales técnicos. Otras industrias con necesidad de gran cantidad de documentación, como las de aeronáutica, telecomunicaciones, automoción o hardware, ha elaborado lenguajes adaptados a sus necesidades. Esto ha conducido a que sus manuales se editen únicamente en versión electrónica, y después se obtenga a partir de ésta las versiones impresas, en línea o en CD. Un ejemplo notable fue el caso de Sun Microsystems, empresa que optó por escribir la documentación de sus productos en SGML, ahorrando costes considerables. El responsable de aquella decisión fue Jon Bosak, que más tarde fundaría el comité del XML.

- **Flexibilidad:** Aunque originalmente los lenguajes de marcas se idearon para documentos de texto, se han empezado a utilizar en áreas como gráficos vectoriales, servicios web, sindicación web o interfaces de usuario. Estas nuevas aplicaciones aprovechan la sencillez y potencia del lenguaje XML.

Estructura general de un documento con lenguaje de marcado.

- Un documento HTML está definido por una etiqueta de apertura <HTML> y una etiqueta de cierre </HTML>. Dentro de este se dividen dos partes fundamentales la cabecera, delimitada por la etiqueta <HEAD> y el cuerpo, delimitado por la etiqueta <BODY>. Por tanto la estructura de un documento HTML será:

```
<HTML>

    <HEAD>

        Definiciones de la cabecera
    </HEAD>

    <BODY>

        Instrucciones HTML

    </BODY>

</HTML>
```

Ninguno de estos elementos es obligatorio, pudiendo componer documentos HTML que se muestren sin ningún problema sin incluir estas etiquetas de identificación. Si se utilizan elementos que forzosamente deban ser incluidos en la cabecera (como la etiqueta de título), no serán reconocidos correctamente si no se incluyen entre las etiquetas de <HEAD>.

Entre las etiquetas <html> y <head> se suelen colocar otras opcionales, como por ejemplo:

```
<meta name="description" content="Información sobre
el Centro,las enseñanzas que se pueden cursar, los
departamentos didácticos">
```

```
<meta name="keywords"
content="educación,enseñanza,instituto, profesores,
alumnos">
```

En este caso las etiquetas le indican a los buscadores el contenido de nuestras páginas (description) y algunas palabras clave (keywords) para su localización.

La cabecera es la sección comprendida entre <head> y </head>. En ella se encuentra necesariamente el título (entre las etiquetas <title> y </title>). El título de la página debe describir su contenido por ejemplo:

Dentro de la cabecera también se suele incluir código en JavaScript, que se reconoce porque va comprendido entre las etiquetas:

```
<script language="JavaScript">
<!--

Aquí iría el código

// -->
</SCRIPT>
```

El cuerpo (body) del documento html es normalmente lo más importante. Es aquí donde debemos colocar el contenido de nuestra página: texto, fotos, etc. El cuerpo está delimitado por las etiquetas <body> y </body>.

Metadatos e instrucciones de proceso.

- **Metadatos:** permiten incluir cualquier información relevante sobre la propia página. La especificación oficial de HTML no define la lista de metadatos que se pueden incluir, por lo que las páginas tienen libertad absoluta para definir los metadatos que consideren adecuados. La etiqueta empleada para la definición de los metadatos es <meta>.
  Atributos:
  - `name = "texto"` - El nombre de la propiedad que se define (no existe una lista oficial de propiedades)
  - `content = "texto"` - El valor de la propiedad definida (no existe una lista de valores permitidos)
  - `http-equiv = "texto"` - En ocasiones, reemplaza al atributo “name” y lo emplean los servidores para adaptar sus respuestas al documento
  - `scheme = "texto"` - Indica el esquema que se debe emplear para interpretar el valor de la propiedad. No suele utilizarse.

Los metadatos habituales utilizan solamente los atributos name y content para definir el nombre y el valor del metadato:

```
<meta name="autor" content="Miguel Ángel Mourelle" />
```

Aunque no existe una lista oficial con los metadatos que se pueden definir, algunos de ellos se utilizan en tantas páginas que se han convertido prácticamente en un estándar. A continuación se muestran los metadatos más utilizados:

- Definir el autor del documento:

```
<meta name="author" content="Juan Pérez" />
```

- Definir el programa con el que se ha creado el documento:

```
<meta name="generator"
content="WordPress 2.8.4" />
```

- Definir la codificación de caracteres del documento:

```
<meta http-equiv="Content-Type"
content="text/html; charset=UTF-8" />
```

```
<meta http-equiv="Content-Type"
content="text/html; charset=ISO-8859-1" />
```

- Definir el copyright del documento:

```
<meta name="copyright" content="librosweb.es" />
```

- Definir el comportamiento de los buscadores:

```
<meta name="robots" content="index, follow" />
```

- Definir las palabras clave que definen el contenido del documento:

```
<meta name="keywords" content="diseño, css, hojas
de estilos, web, html" />
```

- Definir una breve descripción del sitio:

```
<meta name="description" content="Artículos sobre
diseño web, usabilidad y accesibilidad" />
```

La etiqueta que define la codificación de los caracteres (http-equiv="Content-Type") se emplea prácticamente en todas las páginas y las etiquetas que definen la descripción (description) y las palabras clave (keywords) también son muy utilizadas.

Codificación de caracteres. Caracteres especiales (escape)

- **Codificación de caracteres:** Una consideración importante directamente relacionada con el texto de las páginas HTML es la codificación de los caracteres y la inserción de caracteres especiales. Algunos de los caracteres que se utilizan habitualmente en los textos no se pueden incluir directamente en las páginas web:
  - Los caracteres que utiliza HTML para definir sus etiquetas (<, > y ") no se pueden utilizar libremente.
  - Los caracteres propios de los idiomas que no son el inglés (ñ, á, ç, ¿, ¡, etc.) pueden ser problemáticos dependiendo de la codificación de caracteres utilizada. Aunque utilicemos la codificación ISO-8859-1 (Latin 1) para crear páginas en español, puede haber otras directivas (por ejemplo en el servidor, en el editor con el que se ha codificado la página Web, etc.) que contradigan la codificación que nosotros hemos especificado encontrándonos con que se muestran los caracteres de forma errónea.
- La solución a la primera limitación consiste en sustituir los caracteres reservados de HTML por unas expresiones llamadas entidades HTML y que representan a cada carácter:

Entidad	Carácter	Descripción	Traducción
&lt;	<	less than	signo de menor que
&gt;	>	more than	signo de mayor que
&amp;	&	ampersand	ampersand
&quot;	"	quotation mark	comillas
	(espacio en blanco)	non-breaking space	espacio en blanco
&apos;	'	apostrophe	apóstrofo

- Por otra parte, los caracteres propios de los idiomas diferentes al inglés también pueden ser problemáticos. El motivo es que desde que se crea una página web hasta que llega al navegador del usuario, intervienen numerosos procesos:

    - El diseñador crea la página web con su editor HTML (por ejemplo Dreamweaver).
    - Si se trata de una aplicación dinámica, el programador recorta la página HTML del diseñador y la mezcla con el resto del código de la aplicación (por ejemplo PHP).
    - El servidor web almacena las páginas HTML estáticas o el código de la aplicación web y sirve las páginas solicitadas por los usuarios.
    - El usuario solicita y visualiza las páginas web a través de su navegador.
- Si en todos los procesos anteriores se utiliza la misma codificación de caracteres, los caracteres propios de los idiomas se pueden escribir directamente. Si el editor HTML del diseñador utiliza la codificación UTF-8, el entorno de desarrollo del programador también utiliza UTF-8, el servidor web sirve las páginas con esa codificación y el navegador del usuario es capaz de visualizar las páginas con formato UTF-8, el texto anterior se verá correctamente en el navegador del usuario.
- Sin embargo, muchas veces no es posible que todos los procesos involucrados utilicen la misma codificación de caracteres. Por limitaciones técnicas o por decisiones de los diseñadores y programadores, los textos pueden pasar de codificación UTF-8 a codificación ISO-8859 en cualquier momento. Si se produce este cambio sin realizar una conversión correcta, el navegador del usuario mostrará caracteres extraños en todos los acentos y en todas las letras como la ñ.
- La solución más sencilla para asegurar que todos estos caracteres potencialmente problemáticos se van a visualizar correctamente en el navegador del usuario consiste en sustituir cada carácter problemático por su entidad HTML:

Entidad	Carácter	Descripción oficial
ñ	ñ	latin letter n with tilde
Ñ	Ñ	latin capital n letter with tilde
á	á	a acute
é	é	e acute
í	í	i acute
ó	ó	o acute
ú	ú	u acute
Á	Á	A acute
É	É	E acute
Í	Í	I acute
Ó	Ó	O acute
Ú	Ú	U acute
€	€	euro

- **¿UTF-8 o ISO-8859-1?** UTF-8 interpreta más caracteres con lo que nuestra aplicación resultará en principio más compatible, pero tiene el inconveniente de que algunos editores de texto no lo interpretan correctamente, por lo que pueden acabar mostrándose de forma errónea. Por ejemplo, Notepad++ los manejan bien. **ISO 8859-1** es una norma de la ISO que define la codificación del alfabeto latino, incluyendo los diacríticos (como letras acentuadas, ñ, ç), y letras especiales (como ß, Ø), necesarios para la escritura de las siguientes lenguas originarias de Europa occidental: afrikáans, alemán, español, catalán, euskera, aragonés, asturiano, danés, escocés, feroés, finés,

francés, gaélico, gallego, inglés, islandés, italiano, holandés, noruego, portugués y sueco.

- **CONSEJO:** UTF-8, escrito con las herramientas apropiadas con caracteres especiales en forma de entidad o escapados cuando sea necesario.
- Declaraciones de conjunto de caracteres según el W3C:

  HTML5

  ```
  <meta charset="UTF-8">
  ```

  HTML4

  ```
  <meta http-equiv="Content-type"
       content="text/html;charset=UTF-8">
  ```

  XHTML 1.x

  ```
  <meta http-equiv="Content-type"
       content="text/html;charset=UTF-8" />
  ```

- **NOTA:** Para caracteres que no disponen de entidad se utiliza la forma "escapada" con la sintaxis: &#0000; (dos, tres o cuatro dígitos). Conviene tener siempre a mano una tabla de caracteres de escape de cualquier libro o en Google buscar "caracteres de escape html".

Doctype en los documentos HTML y XHTML

En líneas generales, podemos decir que el DOCTYPE es una declaración del estándar usado al construir un documento HTML o XHTML.
El estándar que se utiliza en un documento se define utilizando un DTD, que son las siglas de Definition Type Document, y es una declaración en un metalenguaje para definir otro lenguaje. Dicho de otra manera, el DTD marca las reglas para la definición de lenguajes como el HTML y en el Doctype estamos indicando simplemente qué DTD se tiene que utilizar para interpretar el documento HTML que estamos codificando.
Así, en resumen, podríamos decir que los documentos HTML o XHTML están escritos en un lenguaje y con el Doctype especificamos este lenguaje y la versión del mismo. Los navegadores leerán esta declaración de doctype e interpretarán la página atendiendo a las reglas definidas en ese lenguaje.

Cuando se crea un documento HTML o XHTML es importante añadir una declaración de tipo de documento (doctype). Esta declaración debe ser exacta (tanto en la forma de escribirla como en su combinación de mayúsculas y minúsculas). Para facilitar esta labor, mostramos a continuación una lista de las declaraciones de doctype recomendadas por el W3C.

Para crear un nuevo documento XHTML 1.0 utilice la siguiente plantilla. Para crear documentos para otras versiones utilice los doctype de la siguiente lista:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
        <title>Plantilla estándar de XHTML XHTML 1.0 estricto</title>
        <meta http-equiv="content-type"
                content="text/html;charset=utf-8" />
</head>

<body>

     <p>… Aquí irá el contenido HTML …</p>

</body>
</html>
```

Lista de declaraciones doctype (X)HTML

HTML 4.01

Strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
   "http://www.w3.org/TR/html4/strict.dtd">
```

Transitional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
```

Frameset

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
   "http://www.w3.org/TR/html4/frameset.dtd">
```

XHTML 1.0

Strict (referencia rápida)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Transitional

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Frameset

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

XHTML 1.1 - DTD:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

XHTML Basic 1.1 (referencia rápida):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.1//EN"
    "http://www.w3.org/TR/xhtml-basic/xhtml-basic11.dtd">
```

HTML 5 [**todavía no es un estándar**]

```
<!DOCTYPE HTML>
```

Etiquetas o marcas

- **Las Etiquetas:** Un lenguaje como HTML es un lenguaje de marcas, estas marcas serán fragmentos de texto destacado de una forma especial que permiten la definición de las distintas instrucciones del código HTML. Estas marcas denominadas etiquetas son la base principal de lenguajes como HTML. En documento HTML será un fichero texto con etiquetas que variarán la forma de su presentación.
- Una etiqueta será un texto incluido entre los símbolos menor que < y mayor que >. El texto incluido dentro de los símbolos será explicativo de la utilidad de la etiqueta:
    - <B> Letra Negrita, del inglés Bold (negrita).
    - <TABLE> Definirá una tabla.
    - <IMG> Inclusión de una IMaGen.
- Todo texto que se encuentre entre los caracteres < y > se considerará una etiqueta, si la etiqueta no fuera una de las validas del lenguaje HTML no será tenida en cuenta, sin causar ningún tipo de error. Dejándose el texto o las etiquetas a las que afectaba como si no existiera la etiqueta extraña. Cuando se comete un error sintáctico al expresar una etiqueta o un atributo no se producirá ningún error, simplemente no de obtendrá el efecto que deseábamos.
- El lenguaje HTML es un lenguaje que evoluciona muy rápidamente y cada nueva versión de los programas navegadores presenta etiquetas nuevas que causan efectos más espectaculares o atributos nuevos de las etiquetas ya existentes. Esto causa que los programas más antiguos no entiendan estas nuevas etiquetas y por tanto las considere erróneas y no realice la acción que deseábamos. Dándose el caso de atributos que son válidos solo para un único navegador.
- Cuando creemos código HTML hay que hacerlo lo más estándar posible para permitir que el documento pueda ser visto de forma efectiva por distintos navegadores en máquinas distintas. Por tanto debemos renunciar a efectos espectaculares que solo tienen validez en un navegador e intentar comprobar cómo se ve el documento en una variedad de navegadores.
- El uso de mayúsculas o minúsculas en las etiquetas puede ser indiferente o no dependiendo de la versión de (X)HTML que estemos utilizando.

Elementos

- HTML define el término elemento para referirse a las partes que componen los documentos HTML. Aunque en ocasiones se habla de forma indistinta de "elementos" y "etiquetas", en realidad un elemento HTML es mucho más que una etiqueta, ya que está formado por:
    - Una etiqueta de apertura.
    - Cero o más atributos.
    - Texto encerrado por la etiqueta.
    - Una etiqueta de cierre.

  El texto encerrado por la etiqueta es opcional, ya que algunas etiquetas de HTML no pueden encerrar ningún texto. La etiqueta de fin contendrá el mismo texto que la de inicio añadiéndole al principio una barra inclinada /. El efecto que define un elemento tendrá validez para todo lo que esté incluido entre las etiquetas de inicio y fin, ya sea texto plano u otras etiquetas HTML.

IMPORTANTE: No todos los elementos incluyen etiqueta de inicio y de cierre. Son aquellos en las que el final este implícito, por ejemplo <BR> para representar un salto de línea o <IMG> para incluir una imagen de una imagen. Definen un efecto que se producirá en un punto determinado sin afectar a otros elementos.

Atributos

- **Atributos:** Los elementos pueden presentar modificadores que llamaremos atributos que permitirán definir diferentes posibilidades de la instrucción HTML. Estos atributos se definirán en la etiqueta de inicio y consistirán normalmente en el nombre del atributo y el valor que toma separados por un signo de igual. El orden en que se incluyan los atributos es indiferente, no afectando al resultado. Si se incluyen varias veces el mismo atributo con distintos valores el resultado obtenido será imprevisible dependiendo de cómo lo interprete el navegador. Cuando el valor que toma el atributo tiene más de una palabra deberá expresarse entre comillas, en otro caso no será necesario. Un ejemplo de atributo es:

```
<A HREF="http://www.uca.es">Pagina principal de la UCA</A>
```

- Un elemento podría presentar varios atributos:

```
<HR ALIGN=LEFT NOSHADE SIZE=5 WIDTH=50%>
```

En este caso la etiqueta HR presenta cuatro atributos. El segundo atributo NOSHADE es un caso especial que no presenta valor. El orden en que se especifiquen los atributos no afectarán al resultado final.

Comentarios

- Los **comentarios** en **HTML** son textos que van dentro del código fuente pero que no son mostrados por los navegadores. Estos comentarios son muy útiles para los editores de la página, ya que ayudan a una mayor comprensión del código. La forma correcta de escribir un comentario html es la siguiente:

```
<!--Esto es un comentario-->
```

En los comentarios hay una apertura y un cierre, y todo lo que va dentro de estos dos elementos es el comentario. El código de apertura es el siguiente: <!-- y el cierre del comentario se escribe así: -->.

Documentos válidos y bien formados. Esquemas.

- Un documento en un lenguaje de marcas (HTML, XHTML, XML) se dice "**bien formado**" cuando cumple una serie de reglas descritas en la especificación oficial (marcada por el W3C). Por ejemplo, en XHTML un documento bien formado quiere decir:
  - que todos los elementos deben tener etiquetas de cierre,
  - deben estar escritos de una forma determinada
  - además todos los elementos deben estar anidados correctamente.

    - Los nombres de atributos y elementos deben ir en minúsculas:
    - Tanto los elementos como los atributos deben ir en minúsculas para todos los elementos XHTML y los nombres de atributos.
    - Los elementos que no estén vacíos necesitan etiquetas de cierre:
    - Los valores de las etiquetas deben ir siempre entre comillas:
    - Todos los valores de los atributos deben ir entre comillas, incluso aquellos que sean numéricos.
- Las reglas concretas de cada lenguaje de marcas y versión se pueden consultar en las especificaciones del W3C.

Historia de HTML y XHTML. Diferencias entre versiones

- **Historia de HTML:** El origen de HTML se remonta a 1980, cuando el físico Tim Berners-Lee, trabajador del CERN (Organización Europea para la Investigación Nuclear) propuso un nuevo sistema de "hipertexto" para compartir documentos. Los sistemas de "hipertexto" habían sido desarrollados años antes. En el ámbito de la informática, el "hipertexto" permitía que los usuarios accedieran a la información relacionada con los documentos electrónicos que estaban visualizando. De cierta manera, los primitivos sistemas de "hipertexto" podrían asimilarse a los enlaces de las páginas web actuales.

  Tras finalizar el desarrollo de su sistema de "hipertexto", Tim Berners-Lee lo presentó a una convocatoria organizada para desarrollar un sistema de "hipertexto" para Internet. Después de unir sus fuerzas con el ingeniero de sistemas Robert Cailliau, presentaron la propuesta ganadora llamada WorldWideWeb (W3). El primer documento formal con la descripción de HTML se publicó en 1991 bajo el nombre HTML Tags (Etiquetas HTML) y todavía hoy puede ser consultado online a modo de reliquia informática.

  La primera propuesta oficial para convertir HTML en un estándar se realizó en 1993 por parte del organismo IETF (Internet Engineering Task Force). Aunque se consiguieron avances significativos (en esta época se definieron las etiquetas para imágenes, tablas y formularios) ninguna de las dos propuestas de estándar, llamadas HTML y HTML+ consiguieron convertirse en estándar oficial.

  En 1995, el organismo IETF organiza un grupo de trabajo de HTML y consigue publicar, el 22 de septiembre de ese mismo año, el estándar **HTML 2.0**. A pesar de su nombre, HTML 2.0 es el primer estándar oficial de HTML.

  A partir de 1996, los estándares de HTML los publica otro organismo de estandarización llamado W3C (World Wide Web Consortium). La versión **HTML 3.2** se publicó el 14 de Enero de 1997 y es la primera recomendación de HTML publicada por el W3C. Esta revisión incorpora los últimos avances de las páginas web desarrolladas hasta 1996, como applets de Java y texto que fluye alrededor de las imágenes.

  HTML 4.0 se publicó el 24 de Abril de 1998 (siendo una versión corregida de la publicación original del 18 de Diciembre de 1997) y supone un gran salto desde las versiones anteriores. Entre sus novedades más destacadas se encuentran las hojas de estilos CSS, la posibilidad de incluir pequeños programas o scripts en las páginas web, mejora de la accesibilidad de las páginas diseñadas, tablas complejas y mejoras en los formularios.

La última especificación oficial de HTML se publicó el 24 de diciembre de 1999 y se denomina **HTML 4.01**. Se trata de una revisión y actualización de la versión HTML 4.0, por lo que no incluye novedades significativas.

Desde la publicación de HTML 4.01, la actividad de estandarización de HTML se detuvo y el W3C se centró en el desarrollo del estándar XHTML. Por este motivo, en el año 2004, las empresas Apple, Mozilla y Opera mostraron su preocupación por la falta de interés del W3C en HTML y decidieron organizarse en una nueva asociación llamada WHATWG (Web Hypertext Application Technology Working Group).

La actividad actual del WHATWG se centra en el futuro estándar **HTML 5**, cuyo primer borrador oficial se publicó el 22 de enero de 2008. Debido a la fuerza de las empresas que forman el grupo WHATWG y a la publicación de los borradores de HTML 5.0, en marzo de 2007 el W3C decidió retomar la actividad estandarizadora de HTML.

De forma paralela a su actividad con HTML, W3C ha continuado con la estandarización de XHTML, una versión avanzada de HTML y basada en XML. La primera versión de XHTML se denomina **XHTML 1.0** y se publicó el 26 de Enero de 2000 (y posteriormente se revisó el 1 de Agosto de 2002).

XHTML 1.0 es una adaptación de HTML 4.01 al lenguaje XML, por lo que mantiene casi todas sus etiquetas y características, pero añade algunas restricciones y elementos propios de XML. La versión **XHTML 1.1** ya ha sido publicada en forma de borrador y pretende modularizar XHTML. También ha sido publicado el borrador de **XHTML 2.0**, que supondrá un cambio muy importante respecto de las anteriores versiones de XHTML.

- **Diferencias entre HTML y XHTML:**
  - **XHTML semántico:** XHTML trata de separar totalmente la forma del contenido, por ello ten en cuenta que en XHTML debes utilizar sólo las etiquetas y atributos que sirven para definir qué es cada cosa y no las que sirven para definir cómo se ha de ver cada cosa.
  - **Documento básico XHTML:** El documento básico HTML era un poco más simple y más maleable que el de XHTML. En XHTML estamos obligados a especificar cosas como el juego de caracteres, el DTD que estamos utilizando con el Doctype o un nuevo elemento BODY un poco más complejo.
  - **Las etiquetas deben estar bien anidadas:** Existen reglas estrictas en XHTML para la anidación de etiquetas y es obligatorio que se cierren antes las que se abrieron después. Es decir, no se puede alterar el orden de apertura y cierre de las etiquetas.

    Correcto:

    ```
    <strong><em>Hola</em></strong>
    ```

    Incorrecto:

    ```
    <em><strong>Hola</em></strong>
    ```

- **Todas las etiquetas se cierran:** Todas las etiquetas deben tener su correspondiente etiqueta de cierre. Además se permite que aquellas etiquetas como BR o IMG, que no tienen cierre en HTML, se puedan cerrar con una barra al final de la apertura.

  Correcto:

  ```
  <img> </img> o bien <img/> o bien <img />
  ```

  Incorrecto:

  ```
  <br>
  ```

- **Todas las etiquetas y atributos van en minúscula:** El lenguaje XML es sensible a mayúsculas y minúsculas, por ello para XML la etiqueta con las letras de su nombre o atributos en mayúsculas es distinta que las que van con ellas en minúsculas. Así pues, en XHTML se ha tomado la convención de escribirlo todo en minúsculas.
- **No se puede escribir contenido en el BODY sin meterlo en ninguna etiqueta:** Para que el documento XHTML esté bien formado, no se puede meter un texto directamente en el cuerpo de la página sin haberlo metido en alguna etiqueta previamente.

  Correcto:

  ```
  <body><div>texto</div></body>
  ```

  Incorrecto:

  ```
  <body>texto</body>
  ```

- **Todos los valores de los atributos deben ir entre comillas:** Es indiferente usar comillas dobles o simples, pero estamos obligados a usarlas en los valores que asignamos a cualquier atributo de las etiquetas.
- **Todos los atributos deben tener valor:** No se puede minimizar atributos, es decir, todos deben tener un valor asignado. Un ejemplo de ello es el atributo "selected" de un elemento OPTION que se indicaba para decir que debe aparecer como seleccionado por defecto. Debemos siempre colocarlo asignándole algún valor.

  Correcto:

  ```
  <option selected="selected">loquesea</option>
  ```

  Incorrecto:

  ```
  <option selected>loquesea</option>
  ```

- **No se deben insertar elementos block dentro de elementos inline:** Los elementos de tipo "bloque", como P o DIV, son más generales que los elementos de tipo "en línea", como o . Por ello no podemos colocar elementos tipo block dentro de otros menos generales como los inline.

Correcto:

```
<p>Pepe dijo: <em>Hola</em></p>
```

Incorrecto:

```
<em>Pepe dijo: <p>Hola</p></em>
```

- **Los scripts y estilos deben colocarse en bloques CDATA:** Debido a las características de XML, caracteres como "<" o "&", pueden ser interpretados como parte parte del propio etiquetado del documento XHTML y para evitar que esto ocurra se encapsulan los bloques de script o estilos. De tal modo, cuando se abre un bloque de script o de estilos CSS, para evitar casos de incompatibilidad, debemos colocar su contenido dentro de un bloque CDATA.

```
<script type="text/javascript">
<![CDATA[
//contenido del script sin que XML tenga que
preocuparse por lo que haya dentro...
]]>
</script>
```

- **Conclusión:** Con las anteriores reglas tendrás suficiente para comprobar que las diferencias entre HTML y XHTML son bastante sutiles. Algunas de ellas mínimas, pero no por ello menos importantes. El W3C recomienda el uso de XHTML, por lo que merece la pena ponerse las pilas para empezar a modelar tus páginas atendiendo a este lenguaje un poco más estricto que el HTML. Si quieres que tu página valide correctamente y respete los estándares, deberás tener en cuenta todas las anteriores reglas.
- **Cronología:**

HTML 2.0	HTML 3.2	HTML 4.0	HTML 4.1
(1995)	(1996)	(1998)	(1999)
XHTML 1.0	XHTML 1.1	XHTML 2.0	XHTML5/HTML5
(2000)	2001		Actualidad

- Para el aprendizaje del lenguaje vamos a trabajar sobre la base de XHTML 1.0 que representa un salto cuántico respecto a HTML, está más estandarizado en los navegadores más comunes que se utilizan actualmente y por lo tanto es uno de los más utilizados en los desarrollos de la Web. Alrededor de estos conocimientos, es fácil saber qué novedades incluyen otras versiones más recientes o qué elementos han desaparecido de versiones antiguas.

Estructura de un documento.

- **Estructura de un documento XHTML:** El documento XHTML válido más pequeño posible sería algo similar a esto:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
            Strict//EN" "http://www.w3.org/TR/xhtml1/
            DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
      <meta http-equiv="Content-Type"
            content="text/html; charset=iso-8859-1" />
      <title>Título de la página</title>
</head>

<body>
</body>
</html>
```

La primera etiqueta <?xml ... ?> es una etiqueta opcional, pero recomendada. No es una etiqueta XHTML, sino una declaración XML que indica el juego de caracteres que se va a utilizar en el resto del documento.

La segunda etiqueta <!DOCTYPE ... >sí que es obligatoria. Esta etiqueta indica el tipo de documento (DOCTYPE) de la página. Puesto que existen varias versiones de HTML y XHTML (y cada versión permite unas etiquetas diferentes), el navegador necesita saber a qué versión corresponde la página. La etiqueta contiene la dirección web de la dtd (definición de tipo de documento) que especifica la versión del lenguaje de etiquetas utilizado en el documento (cuál es la estructura, qué etiquetas existen y qué atributos pueden tener).

La etiqueta <html> ... </html> engloba todo el documento html. El atributo xmlns especifica el espacio de nombres del documento. Los espacios de nombres sirven para resolver el problema que aparece cuando en un mismo documento se utilizan etiquetas de distintos lengujes de marcas, en los que pueden coincidir algunos nombres. El espacio de nombres es como el "apellido" de la etiqueta.

El documento html se divide a su vez en dos partes, la cabecera (<head> ... </head>) y el cuerpo (<body> ... </body>).

La cabecera <head> ... </head> contiene información de identificación y control que en general no se muestra en la ventana del navegador, aunque puede afectar a la presentación (por ejemplo, los enlaces a hojas de estilo).

Las etiquetas <meta /> están pensada para proporcionar información sobre el documento a los programas que analicen la página.

La etiqueta de título <title> ... </title> contiene el texto que se muestra en la barra de título de la ventana del navegador.

La etiqueta title es obligatoria y debe incluirse en todas las páginas web.

El cuerpo (<body> .... </body>) contiene lo que se verá en la ventana del navegador. En el ejemplo no hay nada porque se trata de un documento recién creado.

- **Sintaxis de los elementos HTML/XHTML:** Como ya hemos visto, un elemento básico en HTML consiste en dos marcadores al principio y al final de un bloque de texto. Hay algunos elementos que no rodean al texto y, en la mayoría de los casos, los elementos pueden contener subelementos (como html que contiene head y body en el ejemplo anterior).

  Los elementos también pueden tener atributos, que pueden modificar el comportamiento del elemento e introducir un significado adicional.

```
<div id="masthead">
   <h1>Conceptos básicos del
      <abbr title="lenguaje de marcado de
hipertexto">HTML</abbr>
   </h1>
</div>
```

  Muchos de los atributos de HTML (utilizaremos el término HTML haciendo referencia global al lenguaje, independientemente de la versión concreta que estemos utilizando, HTML, XHTML, etc.) son comunes para todos los elementos, aunque algunos son específicos de uno o varios elementos concretos. Éstos tienen siempre la forma palabraclave="valor". El valor debe aparecer siempre entre comillas simples o dobles (en algunas circunstancias se pueden omitir las comillas, pero no es una práctica recomendable con respecto a la predictibilidad, la comprensión y la claridad; se deben poner siempre los valores entre comillas).

  La mayoría de los atributos y sus valores posibles están definidos por las especificaciones HTML; no es posible crear atributos propios sin invalidar el HTML, ya que ello puede confundir a los agentes de usuario y provocar problemas a la hora de interpretar correctamente la página web. Las únicas excepciones reales son los atributos id y class; sus valores están totalmente bajo vuestro control, ya que sirven para añadir significado y semántica propias a vuestros documentos.

  Un elemento que se encuentra dentro de otro elemento se conoce como "hijo" de este elemento. Así pues, en el ejemplo anterior, abbr es hijo del elemento h1, que al mismo tiempo es hijo de div. Y al revés, el elemento div sería "padre" del elemento h1. Este concepto de padre/hijo es muy importante, ya que es la base de CSS y se utiliza mucho en JavaScript.

- **Elementos de bloque y en línea:** En el HTML hay dos categorías generales de elementos que corresponden a dos tipos de contenidos y estructuras que representan estos elementos: elementos de bloque y elementos en línea.

Los elementos de bloque son elementos de nivel superior y normalmente definen la estructura del documento. Puede ser útil ver los elementos de bloque como aquellos que empiezan en una línea nueva y que representan una ruptura con lo anterior.

Algunos elementos de bloque comunes incluyen los párrafos, las listas, los títulos y las tablas.

Los elementos en línea son aquellos que se encuentran incluidos en los elementos estructurales de bloque y que incluyen sólo partes pequeñas del contenido del documento, y no párrafos enteros ni grupos de contenido.

Un elemento en línea no hará que aparezca una línea nueva en el documento; son los tipos de elementos que aparecen dentro de un párrafo de texto. Algunos elementos en línea comunes son los vínculos de hipertexto, las palabras o frases destacadas o las citas breves.

- **El elemento head:** Este elemento está determinado por las etiquetas <head> y </head>. Dentro de las mismas queda determinada la sección head la cual contiene toda la información sobre el documento.

  Esta información no será mostrada por el navegador pero es de suma importancia para los navergadores y para los motores de búsqueda.

  De acuerdo a los estándares de HTML solo un número reducido de etiquetas pueden incluirse en la sección head:

  ```
  <base>
  <link>
  <meta>
  <title>
  <style>
  <script>
  ```

- **El elemento base:** En HTML, los vínculos y las referencias a recursos externos como imágenes, hojas de estilo, etc., se especifican siempre mediante una dirección URL.

  Sintaxis

  ```
  <head>
        <title>Título del documento</title>
        <base href="http://www.dominio.com/imagenes/">
  </head>
  ```

  En este ejemplo podemos observar que la dirección de referencia sirve de base para todas las imagenes que se inserten en la página.

Es decir, que en la sección body para cada imagen que deseo crear solo debo escribir:

```
<body>
     <img src="/paisaje.jpg">
</body>
```

En lugar de:

```
<img src="http://www.dominio.com/imagenes/paisaje.jpg">
```

- **El elemento link:** Este elemento está definido por la etiqueta <link> y establece un enlace que solo puede aparecer en la sección head.

  Sintaxis

```
<head>
     <title>Título del documento</title>
     <link href="http://www.susitio.com/estilos.css"
     type="text/css" rel="stylesheet">
</head>
```

  En este ejemplo podemos observar cómo definir un enlace hacia un archivo de hojas de estilo externo.

- **El elemento meta:** Este elemento está definido por la etiqueta <meta> y puede ser usado para identificar propiedades de un documento (autor, descripción, palabras claves, etc.)

  Sintaxis

```
<head>
<title>Título del documento</title>
     <meta name="author" content="Miguel Ángel
     Mourelle">
     <meta name="description" content="La sección
     head de html contiene información sobre el sitio
     web">
</head>
```

  En este ejemplo vemos cómo definir el autor y la descripción de un sitio.

- **El elemento title:** Este elemento debe figurar en la sección head y nos define el título de la página Web.Para ello utilizamos las etiquetas <title> y </title>. Es recomendable poner títulos ricos en contexto ya que estos aparecen en los motores de búsqueda y ayudan al usuario a identificar el contenido de la página.

  Sintaxis

```
<head>
     <title>Curso de diseño de páginas Web</title>
</head>
```

- **El elemento style:** Este elemento define estilos dentro del documento y utiliza las etiquetas <style> y </style>.

  Sintaxis

```
<head>
    <title>Título del documento</title>
    <style type="text/css">
        p {color:blue}
        h2 {color:red}
    </style>
</head>
```

- **El elemento script:** El elemento script se utiliza para insertar scripts en el documento. Lo definimos por medio de las etiquetas <script> y </script>

  Sintaxis

```
<head>
    <title>Título del documento</title>
    <script type="text/javascript"
    src="http://misitio.com/calculos/calculos.js">
    </script>
</head>
```

- **El elemento body:** Todo lo que queramos que se vea en nuestra página web deberemos escribirlo dentro de la etiqueta body. Eso es lo que llamamos el "cuerpo" del documento. Es la parte visible. Parámetros de personalización del cuerpo de un documento:
  - **Color de fondo:** bgcolor: A través de este parámetro podremos definir el color de fondo que queramos que tenga nuestra página.

    ```
    <body bgcolor="#FF0000"></body>
    ```

  - **Imagen de fondo:** background: Puede que no quieres que tu página tenga un color sólo de fondo, sino que quieras que tu página tenga una imagen. En ese caso deberás indicarlo con la etiqueta "background". La etiqueta quedaría de la siguiente manera:

    ```
    <body background="URL_de la imagen"> </body>
    ```

    Dónde leemos "URL" deberemos escribir la dirección de la imagen que queramos que sea nuestro fondo. Una cosa muy importante que debes saber es que si la imagen no es suficientemente grande para rellenar toda la página, la imagen se repetirá tanto a lo ancho como a lo largo hasta rellenar todo el espacio.

  - **Color de texto:** text: Una vez tenemos el fondo definido, tendremos que predefinir ahora el color del texto de nuestra web. Es decir, tendremos que decirle al

navegador de qué color queremos que sea nuestro texto. Esto lo definiremos con el parámetro "text". Como ejemplo vamos a poner que queremos que nuestro texto sea en negro, con lo que escribiremos lo siguiente:

```
<body text="#000000"></body>
```

- **Márgenes:** leftmargin, topmargin, rightmargin y bottommargin: Para especificar los márgenes utilizaremos el parámetro margin, con su correspondiente indicación delante. Así encontraremos "leftmargin" para el margen izquierdo, "topmargin" para el margen de arriba, "rightmargin" para el margen de la derecha y "bottommargin" para el margen de abajo.

```
<body leftmargin="10px" topmargin="10px"
rightmargin="10px" bottommargin="10px"></body>
```

- **Color de links:** link, alink y vlink: En body también podemos (y de hecho debemos hacerlo) definir el color de los vínculos de nuestra páginas, definir el color con el que se mostraran los links. Aquí debemos diferenciar tres tipos de instrucciones: Debemos definir el color del link (con la etiqueta "link"), el color del link activo (con la etiqueta "alink") y el color del link ya visitado (con la etiqueta "vlink").

```
<body link= "#FF0000" alink= "#FF0000" vlink= "#
0000FF"></body>
```

Nota: Como veremos más adelante, el orden de prioridad para establecer formato (dar estilo) a los elementos HTML es la siguiente (de menor a mayor): estilos en línea (dentro de la propia etiqueta), hoja de estilos incrustada en el documento Web (mediante una etiqueta <style>) y hoja de estilos externa (archivo css independiente)..

Color.

- **Cómo especificar colores en HTML:** Una página web está diseñada y pensada para verse a través de un monitor. El monitor utiliza los colores primarios aditivos rojo, verde y azul. Esto es conocido como el sistema RGB por las siglas de los colores en inglés.

  De esta forma podemos crear un color indicando la cantidad de cada uno de los primarios que interviene en la mezcla. Para cada color podemos especificar 256 intensidades distintas, y al ser tres colores podemos obtener un total aproximado de 16 millones de colores.

  En html, podemos definir un color RGB con valores hexadecimales, siguiendo el formato #F7F0E2 (el más utilizado), con valores para cada color entre el 00 (mínimo) y FF (máximo). Podemos expresarlo con valores decimales, con la fórmula rgb(247, 240, 226), con valores entre 0 y 255, o siguiendo este mismo formato con valores porcentuales, como rgb(90%, 60%, 35%). Estos tres formatos definen el mismo color: #FA75C4, rgb(250, 117, 196) o rgb(98%, 46%, 77%).

Los colores que tienen repetidos los valores de cada color básico, se pueden representar con sólo un valor para cada color. Por ejemplo, #FF9900 es lo mismo que #F90, y #FFFFFF es igual que #FFF.

Además los colores más comunes tienen un nombre propio en inglés. Por lo que en vez de poner #0000FF podemos escribir directamente blue.

- **Colores seguros:** Los colores no se ven igual en todos los monitores, hay un conjunto de colores que son recomendables para su utilización en las páginas web ya que muestran pocas diferencias de color en distintos monitores, son conocidos como colores web o colores seguros.
- **El significado de los colores:**

Blanco:

- El blanco se asocia a la luz, la bondad, la inocencia, la pureza y la virginidad. Se le considera el color de la perfección.
- El blanco significa seguridad, pureza y limpieza. A diferencia del negro, el blanco por lo general tiene una connotación positiva. Puede representar un inicio afortunado.
- En heráldica, el blanco representa fe y pureza.
- En publicidad, al blanco se le asocia con la frescura y la limpieza porque es el color de nieve. En la promoción de productos de alta tecnología, el blanco puede utilizarse para comunicar simplicidad.
- Es un color apropiado para organizaciones caritativas. Por asociación indirecta, a los ángeles se les suele representar como imágenes vestidas con ropas blancas.
- El blanco se le asocia con hospitales, médicos y esterilidad. Puede usarse por tanto para sugerir para anunciar productos médicos o que estén directamente relacionados con la salud.
- Es un color apropiado para organizaciones caritativas. Por asociación indirecta, a los ángeles se les suele representar como imágenes vestidas con ropas blancas.
- A menudo se asocia a con la pérdida de peso, productos bajos en calorías y los productos lácteos.

Amarillo:

- El amarillo simboliza la luz del sol. Representa la alegría, la felicidad, la inteligencia y la energía.
- El amarillo sugiere el efecto de entrar en calor, provoca alegría, estimula la actividad mental y genera energía muscular. Con frecuencia se le asocia a la comida.
- El amarillo puro y brillante es un reclamo de atención, por lo que es frecuente que los taxis sean de este color en algunas ciudades. En exceso, puede tener un efecto perturbador, inquietante. Es conocido que los bebés lloran más en habitaciones amarillas.
- En exceso, puede tener un efecto perturbador, inquietante. Es conocido que los bebés lloran más en habitaciones amarillas.
- Cuando se sitúan varios colores en contraposición al negro, el amarillo es en el que primero se fija la atención. Por eso, la combinación amarillo y negro es usada para resaltar avisos o reclamos de atención.

- En heráldica el amarillo representa honor y lealtad.
- En los últimos tiempos al amarillo también se le asocia con la cobardía.
- Es recomendable utilizar amarillo para provocar sensaciones agradables, alegres. Es muy adecuado para promocionar productos para los niños y para el ocio.
- Por su eficacia para atraer la atención, es muy útil para destacar los aspectos más importantes de una página web.
- Los hombres normalmente encuentran el amarillo como muy desenfadado, por lo que no es muy recomendable para promocionar productos caros, prestigiosos o específicos para hombres. Ningún hombre de negocios compraría un reloj caro con correa amarilla.
- El amarillo es un color espontáneo, variable, por lo que no es adecuado para sugerir seguridad o estabilidad.
- El amarillo claro tiende a diluirse en el blanco, por lo que suele ser conveniente utilizar algún borde o motivo oscuro para resaltarlo. Sin embargo, no es recomendable utilizar una sombra porque lo hacen poco atrayente, pierden la alegría y lo convierten en sórdido.
    - El amarillo pálido es lúgubre y representa precaución, deterioro, enfermedad y envidia o celos.
    - El amarillo claro representa inteligencia, originalidad y alegría.

Naranja:

- El naranja combina la energía del rojo con la felicidad del amarillo. Se le asocia a la alegría, el sol brillante y el trópico.
- Representa el entusiasmo, la felicidad, la atracción, la creatividad, la determinación, el éxito, el ánimo y el estímulo.
- Es un color muy caliente, por lo que produce sensación de calor. Sin embargo, el naranja no es un color agresivo como el rojo.
- La visión del color naranja produce la sensación de mayor aporte de oxígeno al cerebro, produciendo un efecto vigorizante y de estimulación de la actividad mental.
- Es un color que encaja muy bien con la gente joven, por lo que es muy recomendable para comunicar con ellos.
- Color cítrico, se asocia a la alimentación sana y al estímulo del apetito. Es muy adecuado para promocionar productos alimenticios y juguetes
- Es el color de la caída de la hoja y de la cosecha.
- En heráldica el naranja representa la fortaleza y la resistencia.
- El color naranja tiene una visibilidad muy alta, por lo que es muy útil para captar atención y subrayar los aspectos más destacables de una página web.
- El naranja combina la energía del rojo con la felicidad del amarillo. Se le asocia a la alegría, el sol brillante y el trópico.
    - El naranja oscuro puede sugerir engaño y desconfianza.
    - El naranja rojizo evoca deseo, pasión sexual , placer, dominio, deseo de acción y agresividad
    - El dorado produce sensación de prestigio. El dorado significa sabiduría, claridad de ideas, y riqueza. Con frecuencia el dorado representa alta calidad.

Rojo:

- El color rojo es el del fuego y el de la sangre, por lo que se le asocia al peligro, la guerra, la energía, la fortaleza, la determinación, así como a la pasión, al deseo y al amor.
- Es un color muy intenso a nivel emocional. Mejora el metabolismo humano, aumenta el ritmo respiratorio y eleva la presión sanguínea.
- Tiene una visibilidad muy alta, por lo que se suele utilizar en avisos importantes, prohibiciones y llamadas de precaución.
- Trae el texto o las imágenes con este color a primer plano resaltándolas sobre el resto de colores. Es muy recomendable para connminar a las personas a tomar decisiones rápidas durante su estancia en un sitio web.
- En publicidad se utiliza el rojo para provocar sentimientos eróticos. Símbolos como labios o uñas rojos, zapatos, vestidos, etc., son arquetipos en la comunicación visual sugerente.
- El rojo es el color para indicar peligro por antonomasia.
- Como está muy relacionado con la energía, es muy adecuado para anunciar coches motos, bebidas energéticas, juegos, deportes y actividades de riesgo.
- En heráldica el rojo simboliza valor y coraje. Es un color muy utilizado en las banderas de muchos países
- El color naranja tiene una visibilidad muy alta, por lo que es muy útil para captar atención y subrayar los aspectos más destacables de una página web.
- El naranja combina la energía del rojo con la felicidad del amarillo. Se le asocia a la alegría, el sol brillante y el trópico.
    - El rojo claro simboliza alegría, sensualidad, pasión, amor y sensibilidad.
    - El rosa evoca romance, amor y amistad. Representa cualidades femeninas y pasividad.
    - El rojo oscuro evoca energía, vigor, furia, fuerza de voluntad, cólera, ira, malicia, valor, capacidad de liderazgo. En otro sentido, también representa añoranza.
    - El marrón evoca estabilidad y representa cualidades masculinas.
    - El marrón rojizo se asocia a la caída de la hoja y a la cosecha.

Púrpura:

- El púrpura aporta la estabilidad del azul y la energía del rojo.
- Se asocia a la realeza y simboliza poder, nobleza, lujo y ambición. Sugiere riqueza y extravagancia.
- El color púrpura también está asociado con la sabiduría, la creatividad, la independencia, la dignidad.
- Hay encuestas que indican que es el color preferido del 75% de los niños antes de la adolescencia. El púrpura representa la magia y el misterio.
- Debido a que es un color muy poco frecuente en la naturaleza, hay quien opina que es un color artificial.
- El púrpura brillante es un color ideal para diseños dirigidos a la mujer. También es muy adecuado para promocionar artículos dirigidos a los niños.
    - El púrpura claro produce sentimientos nostálgicos y románticos.
    - El púrpura oscuro evoca melancolía y tristeza. Puede producir sensación de frustración.

Azul:

- El azul es el color del cielo y del mar, por lo que se suele asociar con la estabilidad y la profundidad.
- Representa la lealtad, la confianza, la sabiduría, la inteligencia, la fe, la verdad y el cielo eterno.
- Se le considera un color beneficioso tanto para el cuerpo como para la mente. Retarda el metabolismo y produce un efecto relajante. Es un color fuertemente ligado a la tranquilidad y la calma.
- En heráldica el azul simboliza la sinceridad y la piedad.
- Es muy adecuado para presentar productos relacionados con la limpieza (personal, hogar o industrial), y todo aquello relacionado directamente con:
    - El cielo (líneas aéreas, aeropuertos)
    - El aire (acondicionadores paracaidismo)
    - El mar (cruceros, vacaciones y deportes marítimos)
    - El agua (agua mineral, parques acuáticos, balnearios)
- Es adecuado para promocionar productos de alta tecnología o de alta precisión.
- Al contrario de los colores emocionalmente calientes como rojo, naranja y amarillo, el azul es un color frío ligado a la inteligencia y la consciencia.
- El azul es un color típicamente masculino, muy bien aceptado por los hombres, por lo que en general será un buen color para asociar a productos para estos.
- Sin embargo se debe evitar para productos alimenticios y relacionados con la cocina en general, porque es un supresor del apetito.
- Cuando se usa junto a colores cálidos (amarillo, naranja), la mezcla suele ser llamativa. Puede ser recomendable para producir impacto, alteración.
    - El azul claro se asocia a la salud, la curación, el entendimiento, la suavidad y la tranquilidad.
    - El azul oscuro representa el conocimiento, la integridad, la seriedad y el poder.

Verde:

- El verde es el color de la naturaleza por excelencia. Representa armonía, crecimiento, exuberancia, fertilidad y frescura.
- Tiene una fuerte relación a nivel emocional con la seguridad. Por eso en contraposición al rojo (connotación de peligro), se utiliza en el sentido de "vía libre" en señalización.
- El verde oscuro tiene también una correspondencia social con el dinero.
- El color verde tiene un gran poder de curación. Es el color más relajante para el ojo humano y puede ayudar a mejorar la vista.
- El verde sugiere estabilidad y resistencia.
- En ocasiones se asocia también a la falta de experiencia: "está muy verde" para describir a un novato, se utiliza en varios idiomas, no sólo en español.
- En heráldica el verde representa el crecimiento y la esperanza.
- Es recomendable utilizar el verde asociado a productos médicos o medicinas.
- Por su asociación a la naturaleza es ideal para promocionar productos de jardinería, turismo rural, actividades al aire libre o productos ecológicos.
- El verde apagado y oscuro, por su asociación al dinero, es ideal para promocionar productos financieros, banca y economía:
    - El verde "Agua" se asocia con la protección y la curación emocional.

- El verde amarillento se asocia con la enfermedad, la discordia, la cobardía y la envidia.
- El verde oscuro se relaciona con la ambición, la codicia, la avaricia y la envidia.
- El verde oliva es el color de la paz.

Negro:

- El negro representa el poder, la elegancia, la formalidad, la muerte y el misterio.
- Es el color más enigmático y se asocia al miedo y a lo desconocido ("el futuro se presenta muy negro", "agujeros negros"...).
- El negro representa también autoridad, fortaleza, intransigencia. También se asocia al prestigio y la seriedad.
- En heráldica el negro representa el dolor y la pena.
- En una página web puede dar imagen de elegancia, y aumenta la sensación de profundidad y perspectiva. Sin embargo, no es recomendable utilizarlo como fondo ya que disminuye la legibilidad.
- Es conocido el efecto de hacer más delgado a las personas cuando visten ropa negra. Por la misma razón puede ayudar a disminuir el efecto de abigarramiento de áreas de contenido, utilizado debidamente como fondo.
- Es típico su uso en museos, galerías o colecciones de fotos on-line, debido a que hace resaltar mucho el resto de colores. Contrasta muy bien con colores brillantes.
- Combinado con colores vivos y poderosos como el naranja o el rojo, produce un efecto agresivo y vigoroso.

Texto

- **Etiquetas de bloque:**
  - **Título:** <title>
    La etiqueta <title> identifica el título de la página, que se muestra en la barra de título de la ventana del navegador. La etiqueta <title> es obligatoria y sólo puede aparecer una vez en el documento, en la sección <head>. No se le puede aplicar prácticamente ningún estilo (y el único que se le podría aplicar, la dirección de escritura, no funciona actualmente (octubre de 2013) en los navegadores.
  - **Párrafo: <p>**
    La etiqueta de texto más común es la etiqueta <p>, pensada para contener párrafos, es decir todo lo que no tenga un significado especial (títulos, etc).
    En general, los navegadores no muestran los espacios en blanco ni los saltos de línea del código fuente (salvo en la etiqueta <pre>).
    A diferencia de los procesadores de textos, en los que se pueden separar dos párrafos mediante párrafos vacíos, si un párrafo <p> no contiene nada, los navegadores no lo muestran (salvo que la hoja de estilo incluya bordes o márgenes)
  - **Cabeceras:** <h1> <h2> <h3> <h4> <h5> <h6>
    Para los títulos y subtítulos de los apartados de un documento debes utilizar las etiquetas <h1>, <h2>, <h3>, <h4>, <h5> y <h6>. Debes utilizar la etiqueta <h1> para el título principal del documento (a no confundir con la etiqueta

<title>, que corresponde al texto de la barra de título de la ventana del navegador). Debes utilizar la etiqueta <h2> en los títulos de los apartados del documento, <h3> para los subapartados de cada apartado, y así sucesivamente.
La hoja de estilo por omisión de los navegadores suele mostrar las cabeceras en tamaño cada vez más pequeño (incluso más pequeño que el tamaño de los párrafos <p>), en negrita y con márgenes mayores que los los párrafos <p>.

- **Preformateado:** <pre>
  La etiqueta <pre> se utiliza cuando se quiere conservar los espacios en blanco y los saltos de línea del texto original. En el resto de etiquetas, los navegadores no muestran ni las líneas en blanco ni varios espacios en blanco seguidos.
  La hoja de estilo por omisión de los navegadores suele mostrar el bloque preformateado <pre> en tipo de letra monospace (normalmente Courier).
- **Dirección:** <address>
  La etiqueta <address> está pensada para contener información de contacto del autor de la página.
  La hoja de estilo por omisión de los navegadores suele mostrar el bloque de dirección <address> en cursiva.

- **Etiquetas en-línea:** Las etiquetas en-línea son las que abarcan porciones de texto (una o varias palabras) que están contenidas en alguna etiqueta de bloque.
  - **Salto de línea:**
    La etiqueta
 (del inglés break) permite insertar saltos de línea en un párrafo (o cualquier etiqueta de bloque). Por ejemplo, para mostrar varios versos de un poema o canción.
  - **Tipo de información:**
    1. <abbr>
       Pensada para etiquetar abreviaturas. El significado de la abreviatura debe escribirse mediante el atributo title. <abbr title="significado">Abreviatura</abbr>
    2. <acronym>
       Pensada para etiquetar acrónimos (siglas que se pronuncian como palabras). El significado del acrónimo debe escribirse mediante el atributo title. <acronym title="Significado de las siglas">Siglas</acronym>
    3. <cite>
       Pensada para identificar una cita o referencia a otras fuentes.
       Los navegadores suelen mostrar la etiqueta <cite> en cursiva.
    4. <code>
       Pensada para etiquetar fragmento de código de ordenador.
       Los navegadores suelen mostrar la etiqueta <code> en tipo de letra monospace (normalmente Courier).
    5. <del>
       Pensada, junto con la etiqueta <ins>, para etiquetar modificaciones en un texto. Esta etiqueta debería etiquetar el texto que se ha eliminado de una página al revisarlo (si se quiere que se sepa que se ha eliminado, claro).
       Los navegadores suelen mostrar la etiqueta <del> tachada.
    6. <dfn>

7. Pensada para identificar la primera aparición de un término en un texto.
   Los navegadores suelen mostrar la etiqueta <dfn> en cursiva.
8.
9. Pensada para resaltar una porción de texto dándole énfasis, aunque no tanto como con la etiqueta .
   Los navegadores suelen mostrar la etiqueta en cursiva.
10. <ins>
    Pensada, junto con la etiqueta <del>, para etiquetar modificaciones en un texto. Esta etiqueta debería etiquetar el texto que se ha añadido a una página al revisarlo (si se quiere que se sepa que se ha añadido, claro).
    Los navegadores suelen mostrar la etiqueta <ins> subrayada.
11. <kbd>
    Pensada para identificar el texto que debe teclear el usuario.
    Los navegadores suelen destacar la etiqueta <kbd> cambiando el tipo de letra.
12. <samp>
    Pensada para identificar un ejemplo de la salida de un programa o de un script.
    Los navegadores suelen destacar la etiqueta <samp> cambiando el tipo de letra.
13.
    Pensada para resaltar una porción de texto dándole énfasis, aún más que con la etiqueta .
    Los navegadores suelen mostrar la etiqueta en negrita.
14. <var>
    Pensada para identificar una instancia de una variable o de un argumento de programa.
    Los navegadores suelen mostrar la etiqueta <var> en cursiva.

- **Elementos de carácter:** Estas etiquetas están en su mayoría desaconsejadas, ya que pueden conseguirse sus efectos utilizando las propiedades correspondientes de las hojas de estilo.
  1. <b>
     Desaconsejada. Originalmente pensada para etiquetar texto en negrita.
     Los navegadores suelen mostrar la etiqueta <b> en negrita.
  2. <bdo>
     Pensada para elegir la dirección del texto (de izquierda a derecha o de derecha a izquierda). Es obligatorio especificar el atributo dir, con el valor rtl (de derecha a izquierda) o ltr (de izquierda a derecha).
  3. <big>
     Desaconsejada. Originalmente pensada para etiquetar texto de mayor tamaño.
     Los navegadores suelen mostrar la etiqueta <big> con la propiedad font-size: larger.
     HTML5. La etiqueta <big> no está incluida en HTML5. En su lugar se recomienda utilizar propiedades CSS.

4. <i>
   Desaconsejada. Originalmente pensada para etiquetar texto en itálica.
   Los navegadores suelen mostrar la etiqueta <i> con la propiedad font-style: italic.
5. <q>
   Pensada para identificar una cita o referencia a otras fuentes. La recomendación HTML 4.0 especifica que los navegadores deben añadir automáticamente comillas al texto marcado.
6. <small>
   Desaconsejada. Originalmente pensada para etiquetar texto de menor tamaño.
   Los navegadores suelen mostrar la etiqueta <small> con la propiedad font-size: smaller.
7. <sub>
   Pensada para identificar texto en subíndice.
8. <sup>
   Pensada para identificar texto en superíndice.
9. <tt>
   Desaconsejada. Originalmente pensada para etiquetar texto de fuente de espaciado fijo.
   Los navegadores suelen mostrar la etiqueta <tt> en tipo de letra monospace (normalmente Courier).
   HTML5. La etiqueta <tt> no está incluida en HTML5. En su lugar se recomienda utilizar la etiqueta <code> o propiedades CSS.

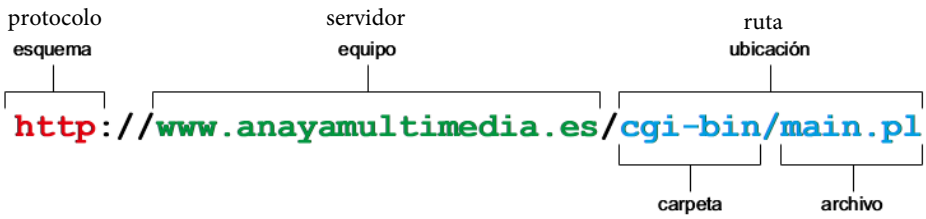
- **Etiquetas de contenedores de texto:**
  - **Bloque de cita:** <blockquote>
    La etiqueta <blockquote> está pensada para identificar una cita larga, que puede contener varios párrafos u otras etiquetas.
    Los navegadores suelen mostrar la etiqueta <blockquote> con márgenes a izquierda y derecha.
  - **División:**
    Las divisiones son el mecanismo más importante para agrupar diversos elementos. Los trataremos con más profundidad más adelante.
    Los navegadores no suelen mostrar la etiqueta de ninguna manera en especial.
- No vamos a abordar por lo general en nuestro estudio de HTML/XHTML los atributos de las etiquetas, ya que vamos a abordar el formato de nuestros documentos mediante CSS. Una referencia completa de todos estos atributos por si los necesitamos en algún momento se encuentra en http://www.w3c.es/Divulgacion/GuiasReferencia/XHTML1/.

Enlaces de hipertexto

- El **hipertexto** es una herramienta de software con estructura no secuencial que permite crear, agregar, enlazar y compartir información de diversas fuentes por medio de enlaces asociativos.
  La forma más habitual de hipertexto en informática es la de hipervínculos o referencias cruzadas automáticas que van a otros documentos. Si el usuario selecciona un hipervínculo, el programa muestra el documento enlazado.

No tiene por qué ser un documento. Puede ser también una imagen o cualquier otro tipo de archivo.

- La estructura general de un enlace o dirección de Internet (URL) es:



- **Enlaces relativos y absolutos:** Las páginas web habituales suelen contener decenas de enlaces de diferentes tipos. Cuando se pincha sobre algunos enlaces, el navegador abandona el sitio web para acceder a páginas que se encuentran en otros sitios. Estos enlaces se conocen como "enlaces externos". Sin embargo, la mayoría de enlaces de un sitio web apuntan a páginas del propio sitio web, por lo que se denominan "enlaces internos".
  Además de internos/externos, la otra característica que diferencia a los enlaces (y por tanto, también a las URL) es si el enlace es absoluto o relativo. Las URL absolutas incluyen todas las partes de la URL (protocolo, servidor y ruta) por lo que no se necesita más información para obtener el recurso enlazado.

  Una URL relativa es una versión abreviada de una URL absoluta. Su objetivo es eliminar todas las partes de la URL absoluta que se pueden adivinar a partir de la información de contexto de la página web. En otras palabras, las URL relativas aprovechan la inteligencia de los navegadores para crear URL incompletas que los navegadores pueden completar deduciendo la información que falta.
  Por ejemplo, si la URL a la que se quiere enlazar utiliza el mismo protocolo y se encuentra en el mismo servidor que la página de origen, la URL relativa puede prescindir de esas partes:

  URL absoluta: http://www.ejemplo.com/ruta1/ruta2/pagina2.html
  URL relativa: /ruta1/ruta2/pagina2.html

  Aunque el ejemplo mostrado es el caso más sencillo de URL relativa, existen otros casos más avanzados en los que se prescinde de parte o toda la ruta del recurso que se enlaza. A continuación se muestran los cuatro tipos diferentes de URL relativas:

  1) **El origen y el destino del enlace se encuentran en el mismo directorio**

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada	Página web llamada pagina2.html y que se encuentra en el mismo directorio
URL absoluta	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina2.html
URL relativa	pagina2.html

Cuando el navegador encuentra una URL relativa que sólo consiste en el nombre de un recurso, supone que el protocolo, servidor y directorio del recurso enlazado son los mismos que los del origen del enlace.

2) **El destino del enlace se encuentra cerca de su origen y en un nivel superior**
En este caso, el recurso que se enlaza no está en el mismo directorio que el origen del enlace pero sí que está cerca y en algún directorio superior. La URL relativa debe indicar de alguna manera que es necesario subir un nivel en la jerarquía de directorios para llegar hasta el recurso.
Para indicar al navegador que debe subir un nivel, se incluyen dos puntos y una barra (../) en la ruta del recurso enlazado. De esta forma, cada vez que aparece ../ en una URL relativa, significa que se debe subir un nivel.

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada	Página web llamada pagina2.html y que se encuentra en el directorio superior llamado ruta2
URL absoluta	http://www.ejemplo.com/ruta1/ruta2/pagina2.html
URL relativa	../pagina2.html

Cuando el navegador encuentra la URL relativa ../pagina2.html, sabe que para encontrar el recurso enlazado (pagina2.html) tiene que subir un nivel desde el lugar en el que se encuentra esa URL relativa. La página que incluye esa URL se encuentra en el directorio ruta1/ruta2/ruta3, por lo que subir un nivel equivale entrar en el directorio ruta1/ruta2.

De la misma forma, si el destino se encuentra un par de niveles por encima, se debe incluir ../ dos veces seguidas:

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada	Página web llamada pagina2.html y que se encuentra en el directorio superior llamado ruta1
URL absoluta	http://www.ejemplo.com/ruta1/pagina2.html
URL relativa	../../pagina2.html

Además de subir niveles, también se puede entrar en otros directorios para obtener los recursos:

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada	Página web llamada pagina2.html y que se encuentra en un directorio llamado ruta4 que se encuentra en la raíz del servidor
URL absoluta	http://www.ejemplo.com/ruta4/pagina2.html
URL relativa	../../../ruta4/pagina2.html

Si se intentan subir más niveles de los que es posible, el navegador ignora todos los ../ sobrantes. Si la página que tiene el enlace es http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html y la URL relativa que se incluye es ../../../../../pagina2.html, el navegador realmente la interpreta como ../../../pagina2.html.
Como el objetivo de las URL relativas es crear URL más cortas y sencillas que las URL absolutas, este método sólo se puede utilizar cuando el origen y el destino se encuentran cerca, porque de otro modo la URL relativa se complica demasiado.

3) **El destino del enlace se encuentra cerca de su origen y en un nivel inferior**
Este caso es muy similar al anterior, pero más sencillo. Si el recurso enlazado se encuentra en algún directorio inferior al que se encuentra el origen, sólo es necesario indicar el nombre de los directorios a los que debe entrar el navegador.

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada	Página web llamada pagina2.html y que se encuentra en un directorio inferior llamado ruta4
URL absoluta	http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/pagina2.html
URL relativa	ruta4/pagina2.html

De la misma forma, se pueden indicar varios directorios seguidos para que el navegador descienda jerárquicamente por la estructura de directorios:

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada	Página web llamada pagina2.html y que se encuentra en un directorio inferior llamado ruta6 que está dentro del directorio ruta5 y que a su vez está dentro del directorio ruta4
URL absoluta	http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/ruta5/ruta6/pagina2.html
URL relativa	ruta4/ruta5/ruta6/pagina2.html

4) El origen y el destino del enlace se encuentran muy alejados
Cuando el origen y el destino de un enlace se encuentran muy alejados (pero en el mismo servidor) las URL relativas se pueden complicar en exceso. Aunque es posible utilizar ../ para subir por la jerarquía de directorios y se puede entrar en cualquier directorio indicando su nombre, las URL relativas que se obtienen son demasiado largas y complicadas.
En estos casos, lo más sencillo es indicar la ruta completa hasta el recurso enlazado comenzando desde la raíz del servidor web. Por lo tanto, estas URL relativas sólo omiten el protocolo y el nombre del servidor.

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada	Página web llamada pagina2.html y que se guarda en un directorio llamado ruta7 que se encuentra en la raíz del servidor
URL absoluta	http://www.ejemplo.com/ruta7/pagina2.html
URL relativa	/ruta7/pagina2.html

Cuando la URL relativa comienza por /, el navegador considera que es la ruta completa comenzando desde la raíz del servidor, por lo que sólo le añade el protocolo y el nombre del servidor origen.

- A continuación se resumen los cuatro posibles casos de URL relativas y el procedimiento que sigue el navegador para convertirlas en URL absolutas:

Si la URL relativa...	El navegador la transforma en URL absoluta...
...sólo consiste en el nombre de un recurso	...añadiendo el protocolo, servidor y ruta completa del origen del enlace
...comienza por ../	...añadiendo el protocolo y servidor del origen del enlace, subiendo un nivel en la jerarquía de directorios y añadiendo el resto de la ruta incluida en la URL relativa
...comienza por /	...añadiendo el protocolo y servidor del origen del enlace
En cualquier otro caso	...añadiendo el protocolo, servidor y ruta completa del origen del enlace, a la que se añade la ruta incluida en la URL relativa

- **La etiqueta <a>**
  - **Enlaces**
    En el código fuente XHTML los enlaces se identifican mediante la etiqueta <a> y su atributo href, que contiene la URL del destino al que conduce el enlace. De forma predeterminada, los enlaces se muestran en los navegadores de color azul y subrayados. En caso de que un enlace contenga una imagen, la imagen muestra un borde azul.
    Enlace a una página Web
  - **Destinos**
    Si el destino es un archivo, es suficiente con conocer su dirección (URL). Si la dirección es un directorio, el servidor puede estar configurado para mostrar un archivo determinado (index.html, home.html, index.php, etc.), mostrar el contenido del directorio, o incluso rechazar la petición.
    Pero si el destino es un elemento situado en el interior del archivo (por ejemplo, un párrafo en el interior de una página web o una imagen insertada en una página), es necesario identificar el punto de destino. Lógicamente, no puede haber dos destinos con el mismo nombre en la misma página web.
    La forma de identificar los destinos en las páginas web ha ido cambiando a lo largo del tiempo:

- En la versión HTML 3.2 y anteriores, los destinos se identificaban mediante la etiqueta <a> y su atributo name.

```
<p>Este es el <a href="#Este">enlace al párrafo
siguiente</a>.</p>

<p name="Este">Este párrafo es el destino del enlace
anterior.</p>
```

El atributo href contiene la dirección de la página, el carácter almohadilla (#) y el atributo name del destino. En caso de ser un enlace dentro de la misma página sólo aparece el carácter almohadilla (#) y el atributo name del destino.

- En las versiones HTML 4.X y XHTML 1.0, los destinos se podían seguir identificando mediante la etiqueta <a> y su atributo name, pero además se introdujo el atributo id. El atributo id se puede añadir a cualquier elemento, así que cualquier elemento de una página web con id puede ser el destino de un enlace aunque no tenga la etiqueta <a>.

```
<p>Este es el <a href="#Este1">enlace al párrafo
siguiente</a>.</p>

<p id="Este1">Este párrafo es el destino del enlace
anterior. </p>
```

Este es el enlace a una pregunta de la FAQ .

El atributo href contiene la dirección de la página, el carácter almohadilla (#) y el atributo id del destino. En caso de ser un enlace dentro de la misma página sólo aparece el carácter almohadilla (#) y el atributo id del destino.

- En la versión XHTML 1.1, el atributo name desaparece y sólo se puede gastar el atributo id para identifcar los destinos.
  Se aconseja identificar los destinos con el atributo id. La identificación de los destinos mediante la etiqueta <a> tenía menos problemas de compatibilidad con navegadores muy antiguos, que ya no se utilizan.
  Si en el atributo href se escribe simplemente el carácter almohadilla (#), casi todos los navegadores se desplazan al principio de la página (aunque este comportamiento no esté definido en ninguna recomendación).

- **Enlaces no HTTP**
  Los navegadores son capaces de gestionar otros protocolos distintos a HTTP, por ejemplo el protocolo FTP:

```
<a href="ftp://servidor.es/ruta1/ruta2/">Enlace a un servidor FTP</a>
```

Un enlace con el atributo href con el valor mailto:dirección_de_correo_electrónico abre automáticamente el cliente de correo electrónico (Outlook, Thunderbird, Evolution, etc.) con un correo dirigido a la dirección indicada. El uso cada vez más extendido del correo web hace que esta opción sea un engorro más que una ayuda.

Atributos de los enlaces ### Atributos de los enlaces.pdf ###

Etiquetas	Atributos	Valor	Descripción
<a>			Define un vínculo
	href	URL	Dirección URL a conectar.
	hreflang	código de lenguaje	Especifica el lenguaje de la URL.
	name	nombre de sección	Para crear un marcapáginas dentro de un documento.
	rel	alternate	Especifica la relación entre el documento actual y el destino del vínculo.
		designates	
		stylesheet	
		start	
		next	
		prev	
		contents	
		index	
		glossary	
		copyright	
		chapter	
		section	
		subsection	
		apendix	
		help	
		bookmark	
		nofollow	
	rev	alternate	Especifica la relación entre el destino del vínculo y el documento actual(vínculo inverso).
		designates	
		stylesheet	

		start	
		next	
		prev	
		contents	
		index	
		glossary	
		copyright	
		chapter	
		section	
		subsection	
		apendix	
		help	
		bookmark	
	coords	coordenadas	Especifica las coordenadas de la superficie que contiene el enlace.
	shape		Define la forma del área.
		rect	Usamos coords="izquierda, arriba, derecha, abajo"
		rectangle	
		circ	Usamos coords="centro x, centro y, radio"
		circle	
		poly	Usamos coords="x1, y1, x2, y2, .., xn, yn"
		polygon	
	target		Indica donde abrir el URL.
		_blank	El URL se abrirá en una nueva ventana.
		_parent	El URL se abrirá en el frameset padre.
		_self	El URL se abrirá en el mismo frame donde fue apretado.
		_top	El URL se abriró en una ventana de tamaño completo.
	type	tipo de contenido	Especifica el tipo de contenido a conectar.

Imágenes.

- **Formatos de imagen para web**
  - **Formatos de imagen:**
    Las imágenes pueden ser de muchos formatos diferentes: bmp, gif, jpg, png, tiff, etc. Pero no todos estos formatos son adecuados para una web, debido a que pueden ocupar mucha memoria o a que no son compatibles con algunos navegadores.
    Los formatos más utilizados para web son el GIF, el PNG y el JPG, que a pesar de ser imágenes de menor calidad que las imágenes BMP, son más recomendables debido a que ocupan menos memoria. Vamos a ver un poco más sobre estos formatos:

1. **Formato GIF:**
   Utilizan un máximo de 256 colores, y son recomendables para dibujos con grandes áreas de un mismo color o de tonos no continuos.
   También si se muestra texto.
   Suelen utilizarse con gran frecuencia, ya que permiten contener transparencias y animación.
   En cambio, no están recomendados para fotografías, ya que se perderían colores, y al no tener áreas de color continuo, el archivo final sería mayor que por ejemplo un JPG.
2. **Formato JPG:**
   Estas imágenes pueden contener millones de colores, en un archivo comprimido de tamaño razonable. Por ejemplo, las imágenes que obtenemos de una cámara digital suelen estar en este formato.
   Por tanto, son especialmente indicadas para fotografías, o gráficos complejos, obteniendo mejores resultados que el GIF.
   En cambio, en gráficos con pocos colores y continuos, generará un archivo mayor que el GIF, y podremos apreciar pérdida de calidad.
3. **Formato PNG:**
   Se trata de un formato de compresión sin perdida. Tiene varias versiones:
   - PNG 8 es un formato de 256 colores muy similar al GIF, que en teoría obtiene archivos algo menores. También admite transparencias.
   - PNG 24 y PNG 36. Es un formato de color verdadero (34 o 36 bits), lo que hace que sea un archivo de tamaño algo mayor. Admite canal de transparencia alfa, lo que quiere decir que puede obtener distintos niveles de transparencia, a diferencia de PNG 8 o GIF que pueden ser totalmente transparentes o no. El resultado es el de mayor calidad, pero también de mayor tamaño. Este formato es el más adecuado cuando necesitamos distintos niveles de transparencia, o requerimos que una imagen muestre correctamente todos sus colores y detalles, evitando la pérdida de calidad que puede producir JPG.

Lo habitual es utilizar GIF o PNG para pequeños gráficos, normalmente elementos del diseño o imágenes simples, y JPG para fotografías.

Nos limitaremos al uso de estos formatos. Ya que aunque algunos navegadores soportan otros, no lo hacen todos. Y hemos de asegurarnos de que cualquier visitante de nuestra página pueda ver las imágenes.

- **Leyes de copyright sobre las imágenes en Internet**
  Según la Oficina de Derechos del Autor de los Estados Unidos, el dueño de una obra con derechos reservados puede entablar una demanda de infracción en contra tuyo si violas los derechos de autor del dueño. Las leyes vigentes de derechos de autor que se aplican a los medios tradicionales, como las impresiones y vídeos, también se aplican en las obras de Internet. Esto significa que las leyes de derechos de autor protegen los textos, vídeos musicales e imágenes en línea, incluso si no se indica el copyright en la página web.

- **Imágenes de dominio público**

  No todas las imágenes en Internet están protegidas con copyright. Las imágenes de dominio público se pueden utilizar con libertad sin tener que solicitar permiso. Las imágenes de dominio público usualmente se encuentran allí porque los derechos de autor han vencido. La Oficina de Derechos del Autor recomienda verificar que una imagen sea de dominio público antes de utilizarla en Internet.

- **Imágenes de "uso razonable"**

  La ley de "uso razonable" excluye la aplicación de copyright en algunas imágenes en la red. Por ejemplo, las leyes de copyright tal vez no protegen las fotografías e imágenes utilizadas en las organizaciones de noticias, universidades y sitios de críticas en línea. Los factores que determinan el "uso razonable" de la imagen incluyen la intención del usuario y el efecto que el uso de la imagen puede tener sobre el precio de mercado. La línea entre una infracción de copyright y el "uso razonable" no está bien definida.

- **Enlaces externos**

  Muchas páginas en Internet están enlazados a imágenes que existen en otros sitios. En ocasiones, los dueños de las páginas proveen permiso para que otros creen un enlace a sus imágenes. En otras ocasiones, tal vez no saben que otros sitios utilizan su trabajo. Un juicio federal del 2000 dictaminó que los hipervínculos no constituyen una infracción al copyright debido a que no se transfiere de manera física la información de la imagen de un sitio a otro cuando un usuario observa una imagen con hipervínculo. Muchos sitios y negocios motivan el uso de los hipervínculos a sus imágenes para aumentar el tráfico a su sitio.

- **Modificaciones de imágenes**

  ¿Es legal cambiar una imagen existente y utilizarla en tu página web? Según la Oficina de Derechos del Autor, puedes cambiar una imagen como lo desees. No obstante, el copyright seguirá perteneciendo al creador original de la imagen. Debes obtener el permiso del dueño del copyright antes de reclamar el derecho de autor sobre tu creación.

- **Licencias Creative Commons**

  Las licencias Creative Commons son varias licencias de copyright (derechos de autor) publicadas el 16 de diciembre de 2002 por Creative Commons, una corporación sin fines de lucro de los Estados Unidos fundada en 2001.

  Las licencias Creative Commons están al momento disponibles en 43 jurisdicciones diferentes de todo el mundo, junto con otras 19 más en desarrollo.1 Las licencias para jurisdicciones fuera de los Estados Unidos están bajo la competencia de Creative Commons International. Véase:
  https://creativecommons.org/licenses/?lang=es_ES

- Bancos de imágenes:
  - **De pago:**

    www.bigstockphoto.es/
    www.shutterstock.com/es
    sp.depositphotos.com/Librería

  - **Gratuitas/dominio público:** No significa que siempre sea posible utilizar estas imágenes en nuestros diseños. Buscar bien cualquier información sobre protección de derechos asociada a las imágenes que deseemos utilizar y, en caso de duda, no utilizar o contactar con el autor para solicitar su permiso.

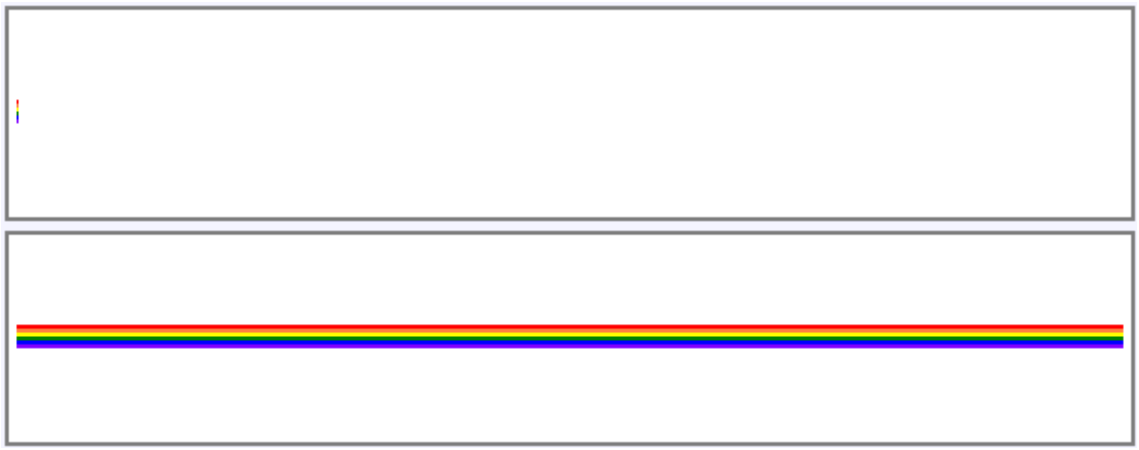
    http://www.publicdomainpictures.net/
    http://www.freestockphotos.biz/
    http://search.creativecommons.org/
    http://openphoto.net/
    http://photorack.net/
    http://www.public-domain-photos.com/
    http://www.stockvault.net/
- **Imágenes de mapa de bits en una página web**
  Las imágenes de mapa de bits se insertan en una página web mediante la etiqueta <img />. Las imágenes no forman parte del documento (como ocurre en los procesadores de texto), sino que se mantienen como archivos aparte.
- **Atributos de img**
  - **Atributo src**
    El atributo src contiene el camino absoluto o relativo a la imagen desde la página web. Para que el navegador pueda mostrar la imagen, el archivo referenciado debe estar disponible. Si los archivos de las páginas web o de las imágenes se cambian de carpeta o de nombre, hay que actualizar los atributos src para que apunten a la dirección correcta.

```
<img alt="Logotipo del W3C en el sitio web del W3C"
src="http://www.w3.org/Icons/w3c_main.png" />
```

  - **Atributos alt y title**
    El atributo alt contiene el texto que deben mostrar los navegadores si la imagen no está disponible. El atributo alt es obligatorio (en el XHTML).
    El atributo title contiene el texto que se muestra en forma de "tip" (cuadrito amarillo que aparece cuando se sitúa el ratón encima de la imagen). El atributo title es optativo.
  - **Atributos width y height**
    Los atributos width y height establecen la anchura y altura de la imagen. Si estos atributos no están definidos, la imagen se muestra con su tamaño original, pero si los atributos están definidos, la imagen se adapta a esos valores. Los valores numéricos se interpretan como píxeles.

Se pueden deformar imágenes para generar líneas horizontales de varios colores a partir de una imagen de 1 píxel de ancho, como muestra el siguiente ejemplo:

```
<img alt="Arco iris" src="arco_iris.png" />
<img alt="Arco iris" src="arco_iris.png" width="100%" height="12" />
```



La principal ventaja de establecer los atributos width y height es que así el navegador no necesita redibujar la página a medida que van llegando las imágenes. Cuando se recibe una página con imágenes, el texto de la página llega antes que las imágenes y el navegador no se espera a recibir todas las imágenes para mostrarla en la pantalla. Si no se indica el tamaño de las imágenes mediante los atributos width y height, el navegador les asigna un tamaño arbitrario pequeño o muestra el atributo alt. A medida que van llegando las imágenes, el navegador tiene que redibujar la página, lo que ocasiona dificultades en la lectura porque el texto se va desplazando. Si se indica el tamaño, el navegador reserva el espacio necesario desde el principio.

El inconveniente de establecer los atributos width y height es que si se cambia la imagen es necesario corregir los valores en la página web si no quiere verse la imagen deformada.

- **Imágenes flotantes**
  Las imágenes se insertan como elementos en línea, es decir, como un carácter más en un texto. Salvo que efectivamente se quiera que la imagen aparezca en mitad de una frase, lo normal es querer que la imagen esté a un lado, con el texto fluyendo en un lado de la imagen. Para ello, se debe utilizar la propiedad float en la hoja de estilo, como veremos más adelante.
- **Utilizar imágenes como enlaces**
  Para enlazar una imagen se debe combinar la etiqueta de enlace (<a> y) y la etiqueta de imagen (<img />) dentro de ella.
  Deberemos especificar los atributos obligatorios HREF, SRC y ALT.
  La sintaxis quedará de la siguiente manera.

```
<a href="destino"> <img src="ruta-de-la-imagen"
      alt="texto alternativo a la imagen" /> </a>
```

El destino, al igual que en los enlaces de texto, podrá ser cualquier recurso interno o externo: una página, una imagen o un archivo.

Además de los atributos obligatorios podremos utilizar atributos opcionales como title, target, etc:

```
<a href="http://www.google.es" target="_blank">
      <img src="logo-google.gif" alt="Logo de Google"
            title="Abrir Google en nueva ventana" /> </a>
```

- **Imágenes de fondo**

  Para poner una imagen de fondo con HTML es necesario agregar el atributo background al tag y hacer referencia a una imagen.

```
      <body background="fondo.jpg">
```

- **Mapas de imagen**

  Un mapa de imagen es una imagen con diferentes zonas interactivas, cada una de ellas puede ser activada por el usuario como si fuera un enlace normal y de esta manera vincular con otras páginas o recursos.
  El proceso de construcción de un mapa de imagen consiste en adjuntar el atributo usemap a img y darle un nombre personalizado junto a un carácter de almohadilla, posteriormente anidar un elemento area dentro de un elemento map. El elemento map debe tener un atributo name y otro id con igual nombre para que el código sea válido.
  Finalmente se pueden añadir tantos elementos area como sean necesarios. Estos elementos se encargan de situar las coordenadas y la forma de la zona interactiva.
  Realizar este proceso de definir las coordenadas de una forma manual puede ser una tarea muy pesada y complicada, por lo tanto es preferible utilizar software de diseño Web como Fireworks o Dreamweaver para la creación de mapas de imagen de una manera visual y copiar el código generado.
  Ejemplo XHTML:

```
<img src="mapa.jpg" alt="Ir a zona principal" usemap="#mapa" />

<map name="mapa" id="mapa">

<area shape="rect" coords="75,93,160,137" href="index.html"
      alt="Ir a index" /> </map>
```

Los mapas de imagen tienen ciertos problemas relacionados con la accesibilidad y la usabilidad. Si la imagen no muestra claramente donde hay que hacer los clics , el mapa de imagen puede pasar totalmente inadvertido por el usuario. Por otra parte si el usuario tiene desactivadas las imágenes no tendrá acceso a las capacidades del mapa de imagen, por lo que si el mapa de imagen fuera el único método de navegación entraría en un callejón sin salida.

Listas.

- **Tipos de listas:** Existen tres tipos de listas:
  - listas ordenadas (<ol> ...
  - listas no ordenadas (<ul> ...)
  - listas de definición (<dl> ... </dl>)

  Dentro de un elemento de lista se puede insertar cualquier otro elemento (texto, imágenes, divisiones, tablas, listas, etc).

  - **Listas ordenadas y no ordenadas:** Tanto las listas ordenadas (<ol>) como las listas desordenadas (<ul>) tienen la misma estructura:
    - las etiquetas <ul> ... o <ol> ... delimitan la lista completa.
    - las etiquetas <li> ... delimitan cada elemento de la lista

```
<ol>
    <li>Punto 1.</li>
    <li>Punto 2.</li>
    <li>Punto 3.</li>
    <li>Punto 4.</li>
</ol>
<ul>
    <li>Lista no ordenada.</li>
    <li>Otro ítem de la lista. </li>
    <li>Más elementos.</li>
    <li>Uno más.</li>
</ul>
```

    La diferencia entre ambos tipos de listas es que los navegadores numeran (con letras o números) los elementos de las listas ordenadas, mientras que en las listas no ordenadas se dibuja un símbolo gráfico.

  - **Listas de definición:** Las listas de definición tienen una estructura distinta:
    - las etiquetas <dl> ... </dl> delimitan la lista completa
    - las etiquetas <dt> ... </dt> delimitan los términos
    - las etiquetas <dd> ... </dd> delimitan las definiciones

```
<dl>
	<dt>software</dt>
		<dd>Conjunto de programas, instrucciones y reglas
		informáticas para ejecutar ciertas tareas en una
		computadora.</dd>
	<dt>libre</dt>
		<dd>Que tiene facultad para obrar o no obrar.</dd>
		<dd>Que no es esclavo.</dd>
		<dd>Que no está preso.</dd>
		<dd>Licencioso,</dd>
</dl>
```

Las etiquetas <dt> y <dd> pueden encontrarse en cualquier orden dentro de un lista de definición, aunque lo razonable es que aparezcan cada <dt> seguido por uno o varios<dd>.

- **Anidamiento:** Las listas pueden presentar cualquier nivel de anidamiento.

```
<ul>
	<li>Item 1
		<ol>
			<li>Item 1 de sublista
				<ul>
					<li>Item 1 de sub-sublista</li>
					<li>Item 2 de sub-sublista</li>
				</ul>
			</li>
			<li>Item 2 de sublista
				<ol>
					<li>Item 1 de sub-sublista</li>
					<li>Item 2 de sub-sublista</li>
				</ol>
			</li>
			<li>Item 3 de sublista</li>
			<li>Item 4 de sublista</li>
		</ol>
	</li>
	<li>Item 2
		<ul>
			<li>Item 1 de sublista</li>
			<li>Item 2 de sublista</li>
		</ul>
	</li>
</ul>
```

Los navegadores suelen utilizar estilos de marcadores distintos para las sublistas no ordenadas (hasta el tercer nivel), pero no en las listas ordenadas. Cada lista ordenada sigue su propia numeración, independiente de las demás.

Tablas.

- **Estructura de una tabla (<table>):** Una tabla XHTML (<table>) es una rejilla rectangular de celdas, formada por los siguientes elementos:
  - Leyenda (<caption>)
  - Cabecera de tabla (<thead>)
  - Cuerpos de tabla (<tbody>)
  - Pie de tabla (<tfoot>)

A su vez, tanto los cuerpos de tabla como la cabecera y el pie de tabla están formados por varias filas (<tr>) formadas por varias celdas (<td> o <th>). Todas las filas tienen el mismo número de celdas (aunque también se pueden unir celdas horizontal y verticalmente).

El código fuente de una tabla sencilla (con un único <tbody>) sería el siguiente:

Ejemplo de tabla

	A	B
1	A1	B1
2	A2	B2

```
<table border="1">
    <caption>Ejemplo de tabla</caption>
    <tbody>
        <tr>
            <td></td>
            <th>A</th>
            <th>B</th>
        </tr>
        <tr>
            <th>1</th>
            <td>A1</td>
            <td>B1</td>
        </tr>
        <tr>
            <th>2</th>
            <td>A2</td>
            <td>B2</td>
        </tr>
    </tbody>
</table>
```

- **Leyenda (<caption>):** La leyenda (<caption>) es texto explicativo opcional que se muestra fuera de la tabla (normalmente, arriba). La leyenda no puede incluir párrafos ni otros elementos de bloque, aunque sí etiquetas en línea (, imágenes, etc).

Los navegadores dan a la leyenda el mismo ancho que a la tabla, por lo que si una leyenda es larga y la tabla estrecha, la leyenda ocupará varias líneas, como muestra el ejemplo siguiente:

- **Cuerpos de tabla (<tbody>), encabezados (<thead>) y pies (<tfoot>):** El cuerpo de tabla (<tbody>) es obligatorio y puede haber tantos como se quiera.
- Tanto a cabecera de la tabla (<thead>) como el pie de tabla (<tfoot>) son opcionales y sólo puede haber uno de cada por tabla. En el código fuente, la etiqueta <tfoot> se encuentra situada antes del primer <tbody>, aunque los navegadores la muestran al final de la tabla.

```
<table border="1">
	<caption>Ejemplo de tabla</caption>
	<thead>
		<tr>
			<th>thead</th>
			<td>celda 1</td>
			<td>celda 2</td>
		</tr>
	</thead>
	<tfoot>
		<tr>
			<th>tfoot</th>
			<td>celda 1</td>
			<td>celda 2</td>
		</tr>
	</tfoot>
	<tbody>
		<tr>
			<th>tbody</th>
			<td>celda 1</td>
			<td>celda 2</td>
		</tr>
	</tbody>
</table>
```

Ejemplo de tabla

thead	celda 1	celda 2
tbody	celda 1	celda 2
tfoot	celda 1	celda 2

Al imprimir una tabla que ocupa varias páginas, Firefox repite al principio y al final de cada página las cabeceras <thead> y pies de tabla <tfoot>, mientras que Internet Explorer y Chrome sólo las incluyen al principio y al final de la tabla.

- **Celdas de datos (<td>) y celdas de cabecera (<th>):** Cada celda de la tabla está marcada con la etiqueta <td> (celda de datos), aunque también se pueden marcar con la etiqueta <th> (celda de cabecera). Las celdas <th> están pensadas para utilizarse en las celdas que sirven de cabecera para la fila o columna, por lo que los navegadores las muestran resaltadas (normalmente, en negrita y centradas en horizontal), aunque se pueden utilizar en cualquier celda.
- Columnas (<col />) y grupos de columnas (<colgroup>): Aunque las celdas de una tabla (<td> y <th>) estén organizadas en filas (<tr>) y grupos de filas (<tbody>, <thead> y <tfoot>), también existen dos etiquetas que permiten hacer referencia a las columnas de una tabla: las etiquetas <col /> (columna) y <colgroup> (grupo de columnas).
  La etiqueta <col /> permite hacer referencia a una columna y la etiqueta <colgroup> permite definir grupos de columnas (de manera similar a como la etiqueta <tbody> define grupos de filas).
  Las etiquetas <col /> y <colgroup> se encuentran situadas al principio de la tabla, después de la etiqueta <caption>.
  El ejemplo siguiente muestra la situación de las estiquetas <col /> y <colgroup> en una tabla.

```
<table border="1" >
	<caption>Leyenda</caption>
	<colgroup><col /><col /></colgroup>
	<colgroup><col /></colgroup>
	<tbody>
		<tr>
			<td>1</td>
			<td>2</td>
			<td>3</td>
		</tr>
	</tbody>
</table>
```

Cada etiqueta <col /> corresponde a una columna, en el mismo orden (la primera etiqueta
<col /> corresponde a la primera columna, y así sucesivamente). En el ejemplo anterior, hay definidas dos grupos de columnas, el primero de los cuales abarca dos columnas y el segundo una columna.

Puede haber menos etiquetas <col /> que columnas en la tabla (también puede haber más, pero serían innecesarias).

En una tabla puede haber únicamente etiquetas <col /> (es decir, puede no haber etiquetas <colgroup />), pero si hay una etiqueta <colgroup /> ya no puede haber etiquetas <col /> que no estén dentro de una etiqueta <colgroup>.

- **Atributos de tablas:**
  - Atributos de <table>
    - El atributo width
      El atributo width establece el ancho de la tabla, en porcentaje o en píxeles. En el siguiente ejemplo, al modificar el tamaño de la ventana del navegador, se puede observar cómo se modifica el tamaño de la primera tabla mientras que la segunda permanece fija.

      ```
      <table width="50%">
      ```

    - El atributo border
      El atributo border establece el grosor del borde de la tabla. El valor se interpreta en píxeles y no admite unidades. Si el valor es 0 o el atributo no está presente, los navegadores no dibujan ni el borde exterior de la tabla ni los bordes interiores de las celdas.

      ```
      <table border="1">
      ```

      Si incorrectamente se escribe una unidad en el atributo border (px, em, cm, etc.), los navegadores muestran el borde mínimo, aunque el validador del W3C no lo considera inválido.

      Si en la hoja de estilo se establece un borde para un elemento de la tabla, este borde se muestra aunque el atributo border sea cero.

    - El atributo frame
      El atributo frame establece qué lados del borde exterior de la tabla son visibles. Los valores posibles son void (sin borde), above (borde superior), below (borde inferior), hside(bordes horizontales), lhs (borde izquierdo), rhs (borde derecho), vsides (bordes verticales), box (los cuatro bordes) y border (los cuatro bordes).

      ```
      <table frame="void" border="10">
      ```

    - El atributo rules
      El atributo rules establece qué lados de los bordes interiores de la tabla son visibles. Los valores posibles son none (ningún borde), all (todos los bordes), rows (los bordes de cada fila), cols (los bordes de cada columna) y groups (los bordes de los <tbody> y de los <colgroup>). Cuando se establece el atributo rules, los navegadores muestran la tabla en el modo de bordes colapsado.

      ```
      <table rules="none" border="1">
      ```

    - El atributo cellspacing
      El atributo cellspacing establece la separación entre celdas y entre las celdas y el borde (como si fuera un margin de los <td>). El valor se

interpreta en píxeles, por lo que no deben escribirse unidades. También se pueden escribir porcentajes, aunque los navegadores lo muestran como si fueran píxeles.

```
<table cellspacing="10">
```

- El atributo cellpadding
  El atributo cellpadding establece la separación entre el borde las celdas y el contenido (como si fuera un padding de los <td>). El valor se interpreta en píxeles, por lo que no deben escribirse unidades. También se pueden escribir porcentajes, aunque los navegadores lo muestran como si fueran píxeles.

```
<table cellpadding="10">
```

- Atributos de <tbody>, <thead> y <tfoot>
  - El atributo valign
  - El atributo valign establece la alineación vertical en todas las celdas del <tbody>. Los valores posibles son top (arriba), middle (centrado), bottom (abajo) y baseline (línea base).
  - Si el atributo valign no está definido, los navegadores centran verticalmente el contenido.

```
<tbody valign="top">
```

La diferencia entre top y baseline es que top alinea la parte superior de la primera línea de texto y baseline alinea la línea base de la primera línea de texto.

- Los atributos align y char
  El atributo align establece la alineación horizontal en todas las celdas del <tbody>. Los valores posibles son left (izquierda), center (centrado), right (derecha), justify (justificado) y char (alineación en carácter).

```
<tbody align="left">
```

Los atributos char y align de <tbody> deberían permitir alinear los valores de una columna a un carácter. El atributo align con el valor char establecen que la alineación es en un carácter determinado y el valor del atributo char establece el carácter por el cual se alinean las celdas.

- Atributos de <tr>
  - Los atributos align, char y valign
  - Los atributos se comportan como con las demás etiquetas (por ejemplo, <table> o <tbody>), ya comentadas en esta misma página.

- Atributos de <td> y <th>
  - Los atributos colspan y rowspan
    Los atributos colspan y rowspan permiten unir una celda con las celdas contiguas, tanto horizontal como verticalmente. El valor de colspan indica la cantidad de celdas unidas en horizontal y el valor de rowspan indica la cantidad de celdas unidas en vertical.

```
<table border="1">
	<caption>Esto es una tabla</caption>
	<tbody>
		<tr>
			<td colspan="2">celdas a1 y b1 unidas</td>
		</tr>
		<tr>
			<td>celda a2</td>
			<td>celda b2</td>
		</tr>
	</tbody>
</table>
```

  - Los atributos align, char y valign
    Los atributos se comportan como con las demás etiquetas (por ejemplo, <table> o <tbody>), ya comentadas en esta misma página.
- Atributos de <col />
  - El atributo span
    El atributo span permite que una etiqueta <col /> haga referencia a varias columnas seguidas. El valor del atributo span indica el número de columnas al hace referencia la etiqueta <col />.

Esto es la leyenda

Celda 1	Celda 2	Celda 3
Celda 4	Celda 5	Celda 6

```
<table border="1">
	<caption>Esto es la leyenda</caption>
	<col span="2" />
	<tbody>
		<tr>
			<td>Celda 1</td>
```

```
                <td>Celda 2</td>
                <td>Celda 3</td>
            </tr>
            <tr>
                <td>Celda 4</td>
                <td>Celda 5</td>
                <td>Celda 6</td>
            </tr>
        </tbody>
    </table>
```

- Los atributos align, valign y width
  Los atributos align, valign y width se comportan como en las demás etiquetas (por ejemplo, <table> o <tbody>), ya comentadas.

- Atributos de <colgroup>
  - El atributo span
    Para indicar el número de columnas que forman parte de un grupo de columnas, se pueden utilizar etiquetas <col /> (con o sin atributo span) o el atributo span de <colgroup>. Si se utiliza el atributo span, su valor indica el número de columnas que forman parte del grupo de columnas.
    Si no coincide el valor del atributo span con las etiquetas <col />, los navegadores hacen caso de las etiquetas <col />.
  - Los atributos align, valign y width
    Los atributos align, valign y width se comportan como en las demás etiquetas (por ejemplo, <table> o <tbody>), ya comentadas. El atributo width indica la anchura de cada columna del grupo.

Marcos (frames).

- FRAMES - MARCOS
  Amados u odiados, útiles o inútiles, excelentes o pésimos, los marcos son instrumentos que forman ya parte habitual del web y que los navegadores gestionan hoy día a la perfección. Los detractores de los marcos afirman la inutilidad de subdividir ulteriormente las páginas web, las cuales, en última instancia, pueden resultar poco legibles. Otros, consideran que los marcos pueden llegar a ser muy útiles ya que se evita cargar las mismas imágenes y se mantienen ordenados el contenido y la estructura del sitio. Naturalmente, abusar de los marcos puede producir como resultado pésimas impostaciones gráficas, obteniendo un efecto contrario al previsto.
  Un punto en contra de los marcos es, ciertamente, su incompatibilidad con los programas de navegación gráfica destinados a invidentes, los cuales se bloquean impidiendo la lectura de las páginas.
  Una buena solución es crear una versión con marco y una versión sin marco.
- ¿Cómo se crean los marcos?
  Antes de nada, repasemos rápidamente las marcas HTML de gestión de marcos.

Documento Marco	<FRAMESET></FRAMESET>	(en lugar de<BODY>)
altura en filas	<FRAMESET ROWS=,,,></FRAMESET>	(píxel ó %)
altura en filas	<FRAMESET ROWS=*></FRAMESET>	(* = tamaño relativo)
anchura en columnas	<FRAMESET COLS=,,,></FRAMESET>	(píxel ó %)
anchura en columnas	<FRAMESET COLS=*></FRAMESET>	(* = tamaño relativo)
anchura del borde	<FRAMESET BORDER=?>	
borde	<FRAMESET FRAMEBORDER="yes|no">	
color del borde	<FRAMESET BORDERCOLOR="#$$$$$$">	
Definición del marco	<FRAME>	(contenido de cada uno de los recuadros)
documento que se debe mostrar	<FRAME SRC="URL">	
denominazione del frame	<FRAME NAME="***"|_blank|_self|_parent|_top>	
anchura de los márgenes	<FRAME MARGINWIDTH=?>	(margen izquierdo y derecho)
altura de los márgenes	<FRAME MARGINHEIGHT=?>	(margen superior e inferior)
barra de desplazamiento o no	<FRAME SCROLLING="YES|NO|AUTO">	
no redimensionable	<FRAME NORESIZE>	
borde	<FRAME FRAMEBORDER="yes|no">	
color del borde	<FRAME BORDERCOLOR="#$$$$$$">	
contenido en ausencia de marco	<NOFRAMES></NOFRAMES>	(para navegadores antiguos)

Para crear una página dividida en marcos, es necesario crear varios archivos HTML referidos a un archivo principal, que es el que permite su gestión. Así pues, antes de nada hace falta impostar este archivo "fuente", y, posteriormente, los demás archivos que componen el marco.

Imaginemos que debemos crear una ventana dividida en marcos como la de la figura, con un marco en la parte superior fijo (en el cual cargaremos el archivo "top.htm", que deberemos crear aparte) y un marco central (en el cual cargaremos el archivo "central.htm", que deberemos, asimismo, crear aparte) que cambiará según cual sea la

página que deba mostrar. Como hemos señalado antes, la gestión de estos dos marcos correrá a cargo de un tercer archivo, el cual deberá invocarlos asignándoles una parte de la página. He aquí el código de esta página:

```
<FRAMESET rows="80,*">
      <frame name="alto" src="top.htm">
      <frame name="central" src="central.htm">
</FRAMESET>
```

Como podemos ver, el código del marco está encerrado entre las marcas <FRAMESET></FRAMESET> que se comportan como las marcas usuales <HTML></HTML>.

El tamaño de los marcos, o mejor dicho, el espacio que cada uno de ellos debe ocupar en la página, queda establecido mediante la marca **rows="80,***, que significa que el marco alto (que en este caso es una fila, "row") debe tener 80 píxel, mientras que "*" significa que todo el resto debe asignarse al marco central. Asimismo, habríamos podido expresar el tamaño de los marcos en tantos por ciento de esta manera:

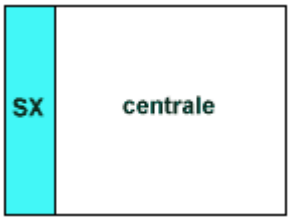
```
<FRAMESET rows="20%,*">
```

Una vez impostados los dos parámetros <FRAMESET></FRAMESET>, dentro de ellos se definen los nombres y los archivos que deberán invocarse en los dos marcos creados. Es necesario dar un nombre al marco (name="alto") e indicar el archivo HTML que deberá cargarse dentro del marco (SRC="top.htm"). Deben, por tanto, crearse previamente dos archivos de nombre "top.htm" y "central.htm".

Fíjate bien en lo importante que es la colocación dentro del código para una correcta interpretación por parte del navegador. Así, si se invirtiera el orden de esta manera:

```
<FRAMESET rows="80,*">
      <frame name="central" src="central.htm">
      <frame name="alto" src="top.htm">
</FRAMESET>
```

el navegador invertiría el orden de asignación y cargaría el archivo "central.htm" en el marco superior, y el archivo "top.htm" en el marco central.



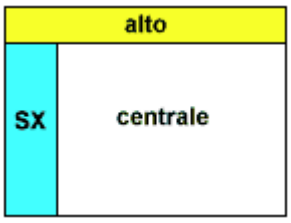
Para crear dos marcos verticales basta sustituir el término "rows" (filas) con el término "cols" (columnas):

```
<FRAMESET cols="100,*">

<frame name="sx" src="sx.htm">
<frame name="central" src="central.htm">

</FRAMESET>
```

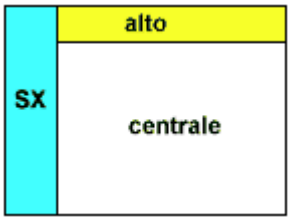
Los antiguos navegadores no soportaban los marcos por lo cual, dada la posibilidad de que se use uno de estos viejos programas para visualizar las páginas, es útil insertar un código que advierta de la presencia de marcos y de la imposibilidad de que el navegador los muestre. Éste es el código que debes incluir:

```
<noframe>

<HTML>
<body>

Atención. Tu navegador no soporta la opción de los marcos. Para ver estas
páginas es necesario descargar un navegador que soporte dicha opción.

</body>
</html>

</noframe>
```

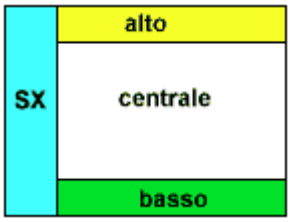
Es posible adoptar simultáneamente una división tanto en columnas como en filas, de manera que se cree una ventana dividida en varios marcos. Veamos cómo debemos intervenir en el código HTML del documento según el número y la posición de los marcos que queremos crear.



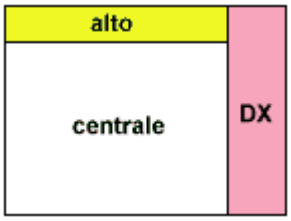

```
<frameset rows="100,*">
        <frame name="alto" src="top.htm">
<frameset cols="150,*">
        <frame name="sx" src="sx.htm.htm">
        <frame name="central" src="central.htm">
</frameset>

</frameset>
```



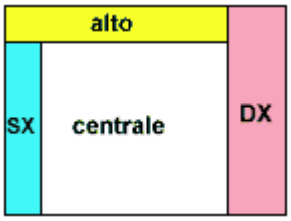

```
<frameset cols="120,*">
        <frame name="sx" src="sx.htm">
<frameset rows="100,*">
        <frame name="alto" src="top.htm">
        <frame name="central" src="central.htm">
</frameset>

</frameset>
```



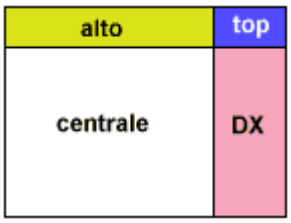

```
<frameset cols="120,*">
        <frame name="sx" src="sx.htm">
<frameset rows="20%,60%,20%,*">
        <frame name="alto" src="top.htm">
        <frame name="central" src="central.htm">
        <frame name="bajo" src="basso.htm">
</frameset>

</frameset>
```



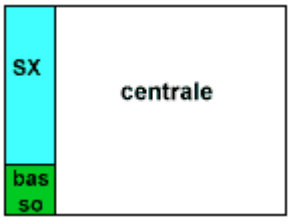

```
<frameset cols="75%,25%">

<frameset rows="20%,80%*">
        <frame name="alto" src="top.htm">
        <frame name="central" src="central.htm">
</frameset>
        <frame name="dx" src="dx.htm">
</frameset>
```



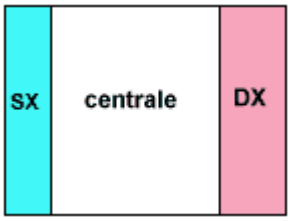

```
<frameset cols="75%,25%">

<frameset rows="20%,80%*">
        <frame name="alto" src="top.htm">
<frameset cols="20%,80%*">
        <frame name="sx" src="sx.htm">
        <frame name="central" src="central.htm">
</frameset> </frameset>
        <frame name="dx" src="dx.htm">
</frameset>
```




```
<frameset cols="75%,25%">

<frameset rows="20%,80%*">
        <frame name="alto" src="top.htm">
        <frame name="central" src="central.htm">
</frameset> <frameset rows="24%,76%">
        <frame name="top" src="top2.htm"> <frame name="dx"
        src="dx.htm">
</frameset> </frameset>
```




```
<frameset cols="25%,75%">

<frameset rows="80%,20%">
        <frame name="alto" src="top.htm">
        <frame name="bajo" src="basso.htm">
</frameset>
        <frame name="central" src="central.htm">
</frameset>
```




```
<frameset cols="20%,60%,20%">
        <frame name="sx" src="sx.htm">
        <frame name="central" src="central.htm">
        <frame name="dx" src="dx.htm">
</frameset>
```

Para eliminar el borde gris de los marcos, se debe insertar el siguiente código:

```
<frameset cols="20%,60%,20%" border=0>
```

Para impedir que los marcos sean redimensionados por el visitante:

```
<frame name="alto" src="top.htm" noresize>
```

Para eliminar siempre las barras de desplazamiento (scrollbars):

```
<frame name="alto" src="top.htm" scrolling="no">
```

Para mostrarlas siempre:

```
<frame name="alto" src="top.htm" scrolling="yes">
```

Para mostrarlas sólo cuando son necesarias:

```
<frame name="alto" src="top.htm" scrolling="auto">
```

Para regular la distancia del contenido del marco al margen superior (marginheight) y a los márgenes izquierdo y derecho (marginwidth):

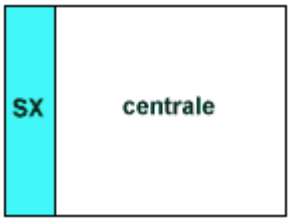
```
<frame name="alto" src="top.htm" marginheight=2 marginwidth=5>
```

Por lo que se refiere a los enlaces dentro de los marcos (es decir, cómo cargar una página en un marco diverso de aquél en que se encuentra el enlace) hay que hacer referencia al nombre que hemos asignado a los diferentes marcos en la fase de realización. Este nombre no se refiere al archivo sino a lo que aparece escrito después de "name=". Por ejemplo, en este caso:

<frame **name="alto"** src="top.htm">

el nombre asignado es "alto".

Tomemos la siguiente página subdividida en marcos:




```
<frameset cols="20%,60%,20%">
        <frame name="sx" src="sx.htm">
        <frame name="central" src="central.htm">
</frameset>
```

Pongamos que de un enlace presente en "SX" tengamos que cargar otra página en el marco "central".

Si el enlace presente en el marco "SX", fuera simplemente:

<A HREF="nuova.htm">Haz clic</A>

la página se cargaría dentro del mismo marco (es decir, "SX") porque sin adecuadas marcas específicas el navegador interpreta que debe cargar la nueva página en el mismo marco en que está presente el enlace.

El código exacto sería:

```
<A HREF="nuova.htm" TARGET="central">Haz clic</A>
```

Otro uso fundamental de la marca <TARGET> es el de llamar un enlace a otra página, la cual se visualizará ocupando la pantalla completa y eliminando todos los marcos preexistentes.

Aquí está el código:

```
<A HREF="nuova.htm" TARGET="_parent">Haz clic</A>
```

Si insertas el código:

```
<base target="_top">
```

a la cabeza del documento HTML todos los enlaces presentes en las páginas eliminarán los marcos existentes, sin necesidad de ir enlace por enlace.

DIV y SPAN

- **La etiqueta **
  La etiqueta permite agrupar varios elementos en línea seguidos dentro de un mismo bloque (por ejemplo, varias palabras seguidas en un párrafo), para después darles formato con la hoja de estilo.

```
<p>El primer servidor web de la historia se instaló en el
CERN en <span>diciembre de 1990</span></p>
```

A menudo, la etiqueta se emplea para asignar clases a porciones de texto.

```
<p>El <span class="resaltar">primer servidor web de la
historia</span> se instaló en el CERN en
<span class="fecha">diciembre de 1990</span></p>
```

Las hojas de estilo por defecto de los navegadores no aplican ningún estilo a la etiqueta .

- **Qué es una división **
  La etiqueta define una división. Esta etiqueta permite agrupar varios elementos de bloque (párrafos, encabezados, listas, tablas, divisiones, etc). En principio, los navegadores no muestran nada especial cuando se crea una división, salvo que se dé formato a la división con la hoja de estilo.
  Nota: Hay gente y programas que llaman "capas" a las divisiones. Parece ser que ese término se debe a que Netscape 4.0 introdujo una etiqueta llamada <layer> (etiqueta que no formó parte de ninguna recomendación del W3C) que jugaba un papel similar a la etiqueta .

```
<div>
    <p>El hombre es fuego; la mujer, estopa; llega el
        diablo y sopla.</p>
    <p>Para el amor y la muerte, no hay cosa fuerte.</p>
    <p>Viejo el pajar, malo de encender y peor de
        apagar.</p>
</div>
```

Una división no puede insertarse dentro de una etiqueta en-línea (,, etc.) o de un bloque de texto (párrafo <p>, encabezado <h1> ... <h6>, dirección <address>, pre-formateado <pre>, lista, etc), pero si puede insertarse dentro de una tabla, de un bloque de cita <blockquote> o de una división .

- **Divisiones anidadas:**
  La etiqueta se puede anidar (es decir, que una división puede contener otras divisiones), por lo que se utiliza para estructurar en bloques el contenido de la página web.

```
<div>
    <div>
        <p>El hombre es fuego; la mujer, estopa; llega
        el diablo y sopla.</p>
        <p>Para el amor y la muerte, no hay cosa
        fuerte.</p>
        <p>Viejo el pajar, malo de encender y peor de
        apagar.</p>
    </div>
```

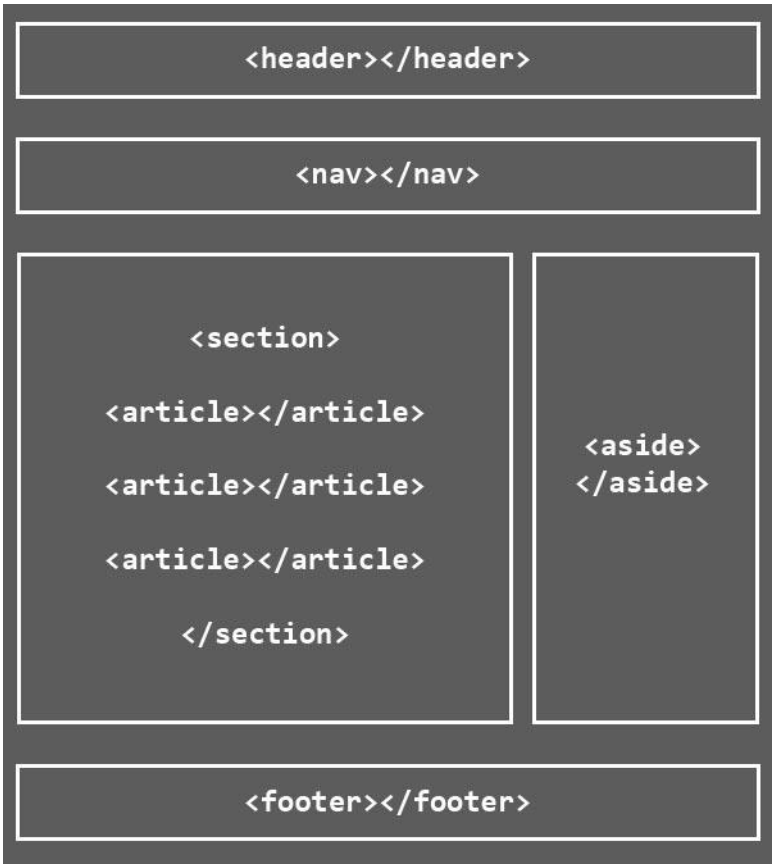
```
        <div>
            <p>¿Enseñas sin saber? Como no sea el culo, no sé qué.</p>
            <p>Practicar hace maestro; que no leer en el cuaderno.</p>
            <p>Lo que natura no da, Salamanca no presta</p>
        </div>
    </div>
```

En este ejemplo tenemos una división que contiene dos divisiones que a su vez contienen tres párrafos cada una.

- **Cuándo utilizar una división:**
  Las divisiones son elementos muy útiles, puesto que permiten agrupar elementos, pero hay que evitar las divisiones innecesarias. Por ejemplo, en general no suele ser necesario crear divisiones cuando sólo contienen un elemento.

Etiquetas semánticas de estructura HTML5

Estas nuevas etiquetas ayudan o sirven para agrupar cada tipo de contenido HTML de nuestra web. Antes de listarlas veamos en la siguiente imagen su jerarquía:



Ahora listamos las etiquetas para posteriormente hacer un breve resumen de cada una de ellas:

ETIQUETA HEADER HTML5

Contiene la agrupación de código HTML correspondiente a la cabecera de la web o referencia descriptiva de artículos. Su sintaxis es:

```
<header>
<!-- Código HTML -->
</header>
```

ETIQUETA NAV HTML5

Nos sirve para definir el conjunto de código HTML de un menú de navegación, solamente contiene enlaces a las diferentes secciones de la web. Su sintaxis es:

```
<nav>
<!-- Código HTML -->
</nav>
```

ETIQUETA SECTION HTML5

Sirve para definir en una sección la agrupación temática del contenido de una página web. Su sintaxis es:

```
<section>
<!-- Código HTML -->
</section>
```

ETIQUETA ASIDE HTML5

Define una sección de la web donde el contenido no tiene que estar necesariamente relacionado con el contenido principal de la web. Generalmente suelen ser barras laterales o sidebars de una web (Plugins sociales, publicidades…). Su sintaxis es:

```
<aside>
<!-- Código HTML -->
</aside>
```

ETIQUETA ARTICLE HTML5

Esta etiqueta nos ayuda a definir un contenido específico dentro de una web y que por si solo tiene sentido. Un código independiente y con sentido puede ser:

- Mensajes de un foro.
- Comentarios.
- Artículos de periódicos, revistas y blogs.
- Widget y similares.

Su sintaxis es:

```
<article>
<!-- Código HTML -->
</article>
```

ETIQUETA MAIN HTML5

Sirve para definir el contenido principal o más destacado de una url. Solo puede haber una etiqueta main.
Dentro de esta nueva etiqueta HTML5 tenemos que tener elementos tales como: cabecera de la web, menú de navegación, información legal de la web, pie de página, barras laterales… Su sintaxis es:

```
<main>
<!-- Código HTML -->
</main>
```

ETIQUETA FOOTER HTML5

Contiene la agrupación de código HTML correspondiente al pie de página que suelen tener habitualmente todas las webs. Muy parecida a la etiqueta header. Su sintaxis es:

```
<footer>
<!-- Código HTML -->
</footer>
```

Formularios

- **El formulario: <form>**
  Un formulario es un conjunto de controles (botones, cajas de texto, casillas de verificación, botones radio, etc) que permiten al usuario introducir datos y enviarlos al servidor web para su procesamiento.
  La etiqueta que delimita un formulario es la etiqueta <form> ... </form>. Los atributos más importantes de la etiqueta <form> son:
  - action: contiene el nombre del agente que procesará los datos remitidos al servidor (por ejemplo, un script de PHP).
  - method: define la manera de envíar los datos al servidor. Los valores posibles son:
    1. get: los valores enviados se añaden a la dirección indicada en el atributo action
    2. post: los valores se envían de forma separada

    Si el atributo method no está establecido, Internet Explorer y Firefox se comportan como si el valor fuera get.

  La etiqueta <form> es un elemento de bloque. En su interior puede haber cualquier elemento típico de una página web (párrafos, imágenes, divisiones, listas, tablas, etc.), además de las etiquetas que crean los controles.
  Los etiquetas que crean los controles en los formularios son <input />, <button>, <select>, <optgroup>, <option> y <textarea>. Además, se pueden estructurar los controles con las etiquetas <fieldset> y <legend>. Por útlimo, la etiqueta <label> permite mejorar la accesibilidad de los controles.
- **Atributos comunes de los controles: name, value, disabled, readonly, tabindex y accesskey**
  El atributo **name** identifica al control. El formulario únicamente envía al servidor los datos de los controles que tienen establecido el atributo name, por tanto si un control no tiene establecido su atributo name, el control simplemente se ignora. Si hubiera varios controles con el mismo atributo name, el formulario envía todos los datos.
  El atributo **value** permite establecer el valor inicial de un control, aunque cada control lo utiliza de una forma ligeramente distinta. El único control sin atributo value es el área de texto (<textarea>).
  El atributo **disabled** permite deshabilitar el control. Una vez deshabilitado, el control ni siquiera puede coger el foco.

```
<input type="submit" value="Enviar" disabled="disabled" />

<input type="text" name="texto" disabled="disabled" />
```

  El atributo readonly permite que el control no sea modificable, aunque el control puede coger el foco.

```
<input type="submit" value="Enviar" readonly="readonly" />

<input type="text" name="texto" value="¡A que no me cambias!"
                    readonly="readonly" />
```

El atributo **tabindex** permite controlar el orden en que el foco pasa de un elemento a otro mediante el tabulador (Tab para avanzar y Shift+Tab para retroceder). Los valores de tabindex pueden ser números naturales (incluido el cero), no necesariamente consecutivos. Si no está presente, los controles se visitan en el orden en que aparecen en el texto. Si está presente, los controles se visitan de menor a mayor.

```
<input type="text" name="texto1" value="Texto 1" />
<input type="text" name="texto2" value="Texto 2" />
<input type="text" name="texto3" value="Texto 3" />
<input type="text" name="texto4" value="Texto 4" />

<input type="text" name="texto1" value="Texto 1" tabindex="5" />
<input type="text" name="texto2" value="Texto 2" tabindex="7" />
<input type="text" name="texto3" value="Texto 3" tabindex="6" />
<input type="text" name="texto4" value="Texto 4" tabindex="8" />
```

El atributo accesskey permite definir teclas de acceso (atajos de teclado). El problema con las teclas de acceso es que a veces entran en conflicto con combinaciones de teclas ya definidas por el navegador o el sistema operativo.

```
<input type="text" name="texto1" value="Acceso con a" accesskey="a" />
<input type="text" name="texto2" value="Acceso con e" accesskey="e" />
<input type="text" name="texto3" value="Acceso con i" accesskey="i" />
<input type="text" name="texto4" value="Acceso con o" accesskey="o" />
```

- **Controles**
  - **Botones: <input /> y <button>**
    Los botones se crean mediante la etiqueta <button>, aunque los botones más usuales en un formulario (los botones Submit y Reset) se pueden crear también con la etiqueta<input />.
  - **Botones Submit y Reset mediante <input />**
    El botón Submit es el que permite al usuario remitir los datos al servidor. Se crea mediante una etiqueta <input /> cuyo atributo type tiene el valor submit. El texto que se muestra en el botón se define mediante el atributo value.

```
<input type="submit" value="Enviar" />
```

El botón Reset restablece los valores iniciales del formulario. Se crea mediante una etiqueta <input /> cuyo atributo type tiene el valor reset. El texto que se muestra en el botón se define mediante el atributo value.

```
<input type="reset" value="Reset" />
```

- **Botones Submit y Reset mediante <button>**

  La etiqueta <button> permite crear botones de tipo submit o reset o botones de tipo general que deben asociarse a scripts para hacer algo. Los botones submit o reset se crean mediante el atributo type con el valor submit o reset. El botón de uso general se crea mediante el atributo type con el valor button.

```
<button type="submit">Enviar</button>
```
Enviar

```
<button type="reset">Borrar todo</button>
```
Borrar todo

```
<button type="button">Botón</button>
```
Botón

  Los botones <button> pueden contener texto e imágenes (o estructuras más complejas, pero no mapas de imágenes).

- **Caja de texto: <input type="text" />, <input type="password" /> y <textarea>**

  Existen dos tipos de cajas de texto: de una sola línea y de varias líneas
  Las cajas de texto de una sola línea se crean mediante la etiqueta <input /> cuyo atributo type tiene el valor text.

```
<input type="text" name="texto" />
```

placeholder permite mostrar una pista de lo que hay que escribir en el cuadro de texto

  El atributo value (optativo) contiene el valor inicial de la caja de texto. El atributo size indica el tamaño en caracteres de la caja en la pantalla (por omisión, las cajas suelen tener 20 carácteres de tamaño). El atributo maxlength indica el número máximo de carácteres que puede escribir el usuario.

```
<input type="text" name="texto" value="Escribe algo" />

<input type="text" name="texto" size="10" />

<input type="text" name="texto" maxlength="5" />
```

  Existe una caja de texto de una sola línea especial para escribir contraseñas que se crea mediante la etiqueta <input /> cuyo atributo type tiene el valor password.

```
<input type="password" name="contrasena" />
```

  Al escribir en una caja de contraseña, en vez de letras aparecen puntos gruesos. Es importante señalar que estas cajas no proporcionan ninguna

seguridad en la transmisión, simplemente ocultan al usuario lo que este escribe.

- **Caja de texto de varias líneas: <textarea>**

  Las cajas de texto de varias líneas se crean mediante la etiqueta <textarea>. Los atributos obligatorios rows y cols establecen el número de filas y columnas de la caja. El atributo value (optativo) contiene el valor inicial de la caja de texto de varías líneas.

```
<textarea name="texto" rows="4" cols="20"></textarea>

<textarea name="texto" rows="3" cols="30"></textarea>

<textarea name="texto" rows="4" cols="40">
      Escribe algo</textarea>
```

- **Casilla de verificación: <input type="checkbox" />**

  Las casillas de verificación se crean mediante la etiqueta <input /> cuyo atributo type tiene el valor checkbox.

```
<input type="checkbox" name="casilla" />
```

  Si el atributo checked tiene el valor checked, la casilla aparece marcada.

```
<input type="checkbox" name="casilla" checked="checked" />
```

  Las casillas de verificación sólo se envían si se han marcado. El atributo value contiene el valor que envía el formulario si la casilla de verificación está marcada. Si el atributo value no está establecido, el formulario envía el valor on.

- **Botón radio: <input type="radio" />**

  Los botones radio se crean mediante la etiqueta <input /> cuyo atributo type tiene el valor radio.

```
<input type="radio" name="radio" value="1" />
```

  Lo botones radio que tienen el mismo atributo name forman un grupo, es decir, que si se marca uno de ellos se desmarca automáticamente el resto.

```
<input type="radio" name="radio" value="1" />Opción 1<br />
<input type="radio" name="radio" value="2" />
```

Los dos ejemplos anteriores, aunque estén separados, forman el mismo grupo de botones de radio ya que su atributo name tiene el mismo valor (en este caso "radio"). Se puede comprobar pulsando en uno de los ejemplos y observando cómo se desmarca el otro ejemplo. Para que fueran independientes, bastaría con que sus atributos name fueran distintos.

Si uno de los botones tiene el atributo checked con el valor checked, el botón aparece marcado.

```
<input type="radio" name="radio3" value="1" />Opción 1<br />
<input type="radio" name="radio3" value="2" checked="checked" />
```

Los botones radio sólo se envían si se han marcado. El atributo value contiene el valor que envía el formulario si el botón radio está marcado. Si el atributo value no está establecido, el formulario envía el valor on, así que para poder saber cuál ha sido la opción elegida por el usuario es necesario establecer con valores distintos los atributos value de todos los elementos de un botón radio .

- **Lista desplegable: <select>**

  Las listas desplegables se crean mediante la etiqueta <select>. Cada opción de la lista se define mediante la etiqueta <option>.

```
<select name="menu">
  <option selected="selected" value="1">Uno</option>
  <option value="2">Dos</option>
  <option value="3">Tres</option>
</select>
```

El atributo selected indica la opción por omisión. Si ningún elemento posee el atributo selected, tanto Firefox como Internet Explorer muestran la primera opción del menú.

Para evitar malentendidos y forzar al usuario a elegir un valor, se suele incluir una opción en blanco al principio de los menús

```
<select name="menu">
  <option value="1" selected="selected"></option>
  <option value="2">Uno</option>
  <option value="3">Dos</option>
  <option value="4">Tres</option>
</select>
```

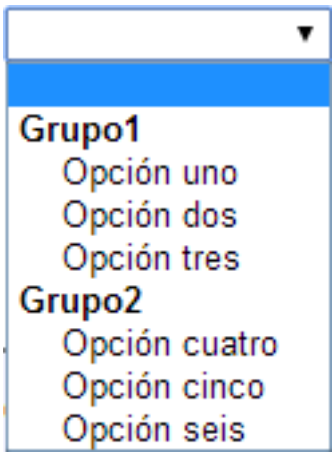
El atributo size permite definir la altura del control

```
<select name="menu" size="3">
  <option value="1" selected="selected">Uno</option>
  <option value="2">Dos</option>
  <option value="3">Tres</option>
  <option value="4">Cuatro</option>
</select>
```

El atributo multiple permite elegir varias opciones simultáneamente (con ayuda de la tecla Control o Mayúsculas)

```
<select name="menu" size="4" multiple="multiple">
  <option value="1" selected="selected">Uno</option>
  <option value="2">Dos</option>
  <option value="3">Tres</option>
  <option value="4">Cuatro</option>
</select>
```

Se pueden agrupar opciones utilizando la etiqueta <optgroup>.



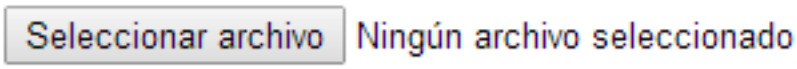

```
<select name="menu">
      <option selected="selected"></option>
      <optgroup label="Grupo1">
            <option value="1">Opción uno</option>
            <option value="2">Opción dos</option>
            <option value="3">Opción tres</option>
      </optgroup>
```

```
        <optgroup label="Grupo2">
            <option value="4">Opción cuatro</option>
            <option value="5">Opción cinco</option>
            <option value="6">Opción seis</option>
        </optgroup>
    </select>
```

El atributo value de cada opción contiene el valor que envia el formulario si la opción está elegida. Si no se define el atributo value, el formulario envía como valor el texto que aparece en el menú.

- **Selector de archivo: <input type="file" />**
- El selector de archivo se crea mediante la etiqueta <input /> cuyo atributo type tiene el valor file.

```
<input type="file" name="archivo" />
```



- **Botón de imagen: <input type="image" />**
  El control de tipo imagen inserta una imagen que funciona como un botón (aunque los navegadores no le dan relieve como a los botones). Al hacer clic en un punto de la imagen se envía el formulario (como si se hubiera pulsado un botón submit) y se envían las coordenadas del punto en el que se ha hecho clic (junto con los valores de los otros controles del formulario).

```
<input type="image" name="GNU" alt="GNU" src="gnu.jpg" />
```

Si se define el atributo value, el formulario envía también el nombre del control con el valor del atributo.

- **Control oculto: <input type="hidden" />**
  El control oculto se crea mediante la etiqueta <input /> cuyo atributo type tiene el valor hidden. Lógicamente, los navegadores no muestran estos controles en la pantalla (aunque pueden verse en el código fuente). Se utilizan para almacenar información que de otro modo se perdería (por ejemplo, cuando hay varios formularios encadenados).

```
<input type="hidden" name="nombre" value="pepito" />
```

- **Grupos de controles: <fieldset>**
  La etiqueta <fieldset> permite agrupar un conjunto de controles. Los navegadores muestran una caja alrededor de cada grupo de controles.
  La etiqueta <legend> permite añadir una leyenda al <fieldset>. Los navegadores muestran la leyenda sobre el borde que rodea el grupo de controles.

```
<fieldset>
	<legend>Formulario</legend>
	<input type="text" name="texto"/><br/>
	<select name="lista">
		<option>Uno</option>
		<option>Dos</option>
		<option selected="selected">Tres</option>
	</select>
</fieldset>
```

- **Accesibilidad: <label>**
  La etiqueta <label> permite asociar un control con un texto, de manera que mejore la accesibilidad de los formularios.
  La asociación entre el control y la etiqueta <label> puede ser implícita o explícita. Se dice que la relación es implícita cuando el control se encuentra en el interior de la etiqueta. Se dice que la relación es explícita cuando la etiqueta <label> contiene el atributo for, que indica el control afectado (el control tiene entonces que tener establecido el atributo id).
  Por ejemplo, en el caso de una casilla de verificación, la etiqueta <label> permite que la casilla se marque o desmarque haciendo clic en el texto, como se muestra en los ejemplos siguientes:

```
<input type="checkbox" name="casilla" />Casilla 1

<label><input type="checkbox" name="casilla" />
	Casilla 1</label>

<input type="checkbox" name="casilla" id="casilla1" />

<label for="casilla1">Casilla 1</label>
```

  En el caso de la cajas de texto <input type="text" />, la etiqueta <label> permite que el cursor se sitúe en la caja de texto haciendo clic en el texto, como se muestra en los ejemplos siguientes:

```
Nombre: <input type="text" name="casilla" />

Nombre:
<label>Nombre: <input type="text" name="casilla" /></label>

Nombre:
<label for="text1">Nombre:</label><input type="text"
name="casilla" id="text1" />
```

Elementos en desuso (deprecated).

- **Etiquetas en desuso u obsoletas de html**
  La web evoluciona a diario y por lo tanto los lenguajes de programación también, en esta oportunidad veremos algunas etiquetas declaradas en desuso u obsoletas, pero por que en desuso? muy fácil, con el tiempo aparecen mejores técnicas para aplicar el mismo efecto o formato al contenido de una web, la mayoría se aplica con CSS.
    - Etiqueta Applet.
      Esta etiqueta sirve para cargar un un applet de java, a sido sustituida por las etiquetas < objetc> < /object>.
    - Etiqueta Basefont.
      Esta etiqueta se utilizaba para poder asignarle formato al texto, como el color, tamaño o tipo de fuente, actualmente se utiliza codigo CSS.
    - Etiqueta Center.
      Esta etiqueta nos sirve para alinear el texto o un objeto al centro de la página web o del elemento en el que este, actualmente se utiliza código CSS.
    - Etiqueta Font.
      Esta etiqueta nos sirve para asignarle formato al texto (color, tamaño, tipo de letra) que este entre la etiqueta < font> y < /font>, actualmente se utiliza código CSS.
    - Etiqueta Menu.
      Esta etiqueta nos muestra una lista de elementos, actualmente se utilizan las etiquetas < ul> < /ul>.
    - Etiqueta S o Strike.
      Estas etiqueta nos sirve para tachar el texto, actualmente se utiliza codigo CSS.
    - Etiqueta U.
      Esta etiqueta nos sirve para subrayar el texto, actualmente se utiliza código CSS.

Bien, ahora ya conocemos algunas etiquetas obsoletas, tal vez aun te funcionen en los navegadores, pero con el tiempo lo dejaran de hacer, así que si aun tienes algunas de estas etiquetas empieza a actualizar el cogido de tu sitio.

- Texto parpadeante
  Podemos hacer que el texto parpadee así utilizando la etiqueta <blink>. Usadlo con cuidado porque es una de las cosas que menos gustan a los navegantes. He aqui un ejemplo:

```
<html>
        <head>
                <title>ejemplo de texto parpadeante</title>
        </head>
        <body>
                <p>Esta línea es normal.</p>
                <blink>Esta en cambio parpadea.</blink>
        </body>
</html>
```

- Marquesinas:
  Marquesina= una marquesina son una(s) palabra(s) que se mueve o sea que digamos: un texto aparece en la derecha de la pantalla y se va moviendo a la izquierda, y cuando llega al final de su rumbo el texto empieza a desaparecer. Y aquí explicaremos absolutamente todo sobre como funcionan las marquesinas.
  Bueno, en primer lugar enseñaremos las 2 etiquetas basicas de marquesina, la marquesina normal(de derecha a izquierda) y la marquesina que rebota.

  -Marquesina normal:

  ```
  <marquee>Texto aquí</marquee>
  ```

  -Marquesina que rebota:

  ```
  <marquee behavior=alternate>Texto aquí</marquee>
  ```

  -Como controlar la cantidad de espacios que se salta la marquesina:

  ```
  <marquee scrollAmount="1">Texto aquí</marquee>
  ```

  -Controlar el tiempo en el que se salta esos espacios ("100" son los milisegundos.):

  ```
  <marquee scrollAmount="1" scrolldelay="100">
       Texto aquí</marquee>
  ```

-Controlar desde donde hasta donde la marquesina:

```
<marquee width=50% height=80 ALIGN=BOTTOM>
      Texto aquí</marquee>
```

P.D= este codigo se puede colocar los numeros en % o pixeles(el "80" es el pixele.)

-Colocar un fondo a color a la marquesina:

```
<marquee bgcolor="AA0000">Texto aquí</marquee>
```

El efecto se ve usando IE, con Mozila no se aprecia

-Controlar dirección:

```
<marquee direction="up">este texto está en dirección
      hacia arriba</marquee>
```

P.D= las direcciones tienen que estar en ingles.