

Diseño de la interfaz web. estilos

Montserrat Rovira Marco

índice

<u>Introducción</u>	5
<u>Resultados de aprendizaje</u>	7
<u>1 Planificación y diseño de interfaces web. HTML5</u>	9
<u>1.1 Diseño web centrado en el usuario</u>	9
<u>1.2 Elementos de diseño: percepción visual</u>	12
<u>1.2.1 Factores humanos</u>	12
<u>1.2.2 Elementos de percepción visual: colores</u>	13
<u>1.2.3 Elementos de percepción visual: tipografía</u>	14
<u>1.2.4 Elementos de percepción visual: iconos</u>	18
<u>1.2.5 Elementos de percepción visual: composición del sitio web</u>	19
<u>1.3 Lenguajes y aplicaciones de diseño web</u>	23
<u>1.3.1 Lenguajes de desarrollo web</u>	24
<u>1.3.2 Herramientas de desarrollo. Editores HTML</u>	26
<u>1.3.3 Navegadores web</u>	30
<u>1.4 Diseño web con HTML5</u>	32
<u>1.4.1 Formato de HTML5</u>	33
<u>1.4.2 Estructura del código HTML5</u>	34
<u>1.4.3 Elementos de HTML5</u>	36
<u>1.4.4 Formularios con HTML5</u>	49
<u>1.4.5 Visualización de elementos con HTML5</u>	56
<u>1.4.6 Estructura semántica de la página web</u>	57
<u>1.4.7 Validación HTML5</u>	61
<u>1.4.8 Seguridad en HTML5</u>	61
<u>2 Creación y aplicación de estilos a la interfaz web. CSS3</u>	63
<u>2.1 Hojas de estilos: CSS3</u>	63
<u>2.1.1 Incluir estilos CSS en un documento HTML</u>	64
<u>2.1.2 Características de CSS: cascada y herencia</u>	66
<u>2.1.3 Formato de una regla CSS</u>	69

2.1.4 Selectores	70
2.1.5 UnitatsdemesuraCSS	79
2.1.6 PropietatsCSS	81
2.1.7 Elmodelcaixa	84
2.1.8 Reglesarrova	98
2.1.9 TestivalidacióCSS	99
2.2 Dissenywebadaptatiu	102
2.2.1 Tècniquespercrearundissenyadaptatiu	103
2.2.2 'Mediaqueries'	104
2.2.3 Desenvoluparun'grid'fluid	108
2.2.4 Desenvoluparun'grid'flexible	113
2.2.5 Contingutsiimatgesflexibles	114

Introducción

El éxito de un sitio web no depende solamente de los contenidos que formen parte, sino que es fundamental que esté bien diseñado, con un aspecto atractivo y con una información fácilmente accesible. Es por ello que, a la hora de diseñar una web, hay que tener en cuenta no sólo la usabilidad y la accesibilidad a toda la información, sino también su estética, es decir, una interfaz que sea suficientemente amigable para resultar agradable acceder al sitio web.

En esta unidad se trabajan los conceptos básicos de diseño de una interfaz web, desde el que son los elementos básicos que conforman la web, como lenguajes de implementación, hasta las herramientas informáticas y los editores que facilitan su desarrollo.

Así, el apartado "Planificación y diseño de interfaces web. HTML 5 "se inicia con los conceptos de interfaz web y experiencia de usuario como concepto fundamental para centrar el diseño de un sitio web en su objetivo. También se explican las características de los elementos de diseño básicos, como son el color y la tipografía. A continuación se lleva a cabo el desarrollo del lenguaje de programación HTML, en su versión HTML5, y la implementación de este mediante diferentes editores.

El apartado "Creación y aplicación de estilos a la interfaz web. CSS3 "se centra, básicamente, en las hojas de estilo CSS que, entre otros, aportan una mejora significativa a la presentación de documentos web con HTML. Se trabajan todos los conceptos básicos de CSS3, así como también la implementación para diseño web adaptativo, es decir, la visualización de la web desde cualquier dispositivo. Finalmente, se explican las características básicas del *framework* Bootstrap que permiten el desarrollo de una web con HTML5 y CSS3.

En el apartado "Generación de sitios web. Prototipos y plantillas" se explican las fases de desarrollo de un proyecto web incidiendo en la parte de diseño, correspondiente a la realización de prototipos, y también en la fase de desarrollo, en el que correspondería a la creación de una web desde plantillas prediseñadas, así como también las herramientas informáticas que se utilizan en la actualidad. Seguidamente, se explica qué son los sistemas de gestión de contenidos y algunas de sus características para la creación de sitios web. Finalmente, se aborda la importancia del posicionamiento SEO, así como algunas técnicas y pautas a tener en cuenta para mejorar el posicionamiento de la web.

Al finalizar esta unidad debería poder crear un sitio web con HTML5 y CSS3 utilizando el *framework* Bootstrap y siguiendo los estándares generales de diseño de interfaces web.

1. Planificación y diseño de interfaces web. HTML5

Hoy en día es frecuente que cualquier usuario haga uso de dispositivos conectados a Internet para llevar a cabo cualquier tipo de tareas cotidianas, desde hacer la compra *online* de un determinado producto por medio de su web hasta reservar las entradas de un espectáculo o sencillamente acceder a las noticias de un periódico digital.

De igual manera, las empresas son conscientes de la importancia de su presencia en Internet y, así, tener un sitio web bien diseñado y accesible tiene una influencia directa no sólo en la imagen de la empresa, sino en el mismo negocio.

Pero de qué depende que una web de un periódico digital, por ejemplo, tenga más visitas que otro o que una reserva se haga desde un portal web u otro? obviamente, una parte fundamental será el contenido o el producto final, pero en muchas ocasiones también dependerá de la facilidad y amigabilidad que el usuario encuentre en acceder a la información de la web y de la simplicidad en hacer las gestiones correspondientes.

Por este motivo es de vital importancia no sólo cuál es el contenido y como está organizada la web, sino también qué aspecto presenta al usuario. este aspecto es lo que se denominará *interfaz web*.

Es imprescindible diseñar y desarrollar la interfaz de una web de tal manera que el usuario final pueda acceder fácilmente al servicio que se ofrece y, además, que este acceso le sea agradable hasta el punto de volver a acceder en un futuro.

Así pues, a fin de crear la interfaz de un sitio web será necesario disponer de unos parámetros estándar para diseñar la información a mostrar al usuario y utilizar las

técnicas y los lenguajes de programación web adecuados, tales como el HTML5 y los hojas de estilo CSS3.

1.1 Diseño web centrado en el usuario

Cuando un usuario accede una web, la interfaz es la parte que visualiza y que el permite comunicarse con el sistema, es decir, la que hace que el usuario entre en contacto, física y cognitivamente con éste.

Una **interfaz web** es el conjunto de todos aquellos elementos gráficos que ofrecen al usuario la presentación y la navegación por un sitio web con la finalidad de llevar a cabo unas acciones determinadas.

Dado que la interacción entre el usuario y el sistema se produce por medio de la interfaz, esta es necesario que refleje cómo se hace esta comunicación.

La **interacción persona-ordenador (IPO)** es una disciplina dedicada a estudiar la relación interactiva entre las personas y la tecnología. El objetivo principal es conseguir productos interactivos fáciles de usar, satisfactorios y que resulten realmente útiles.

Así pues, se pueden identificar los tres elementos clave de la IPO: tecnología, personas y diseño:

- La IPO se dedica a estudiar toda la **tecnología** que permita la interacción y el uso de esta con el usuario. Esta tecnología no se limita exclusivamente ordenadores, sino a todos los dispositivos que se utilizan cotidianamente, como ahora móviles y tabletas.
- La IPO intenta mejorar la relación interactiva entre las **personas** y la tecnología. Este ámbito estudia el comportamiento de las personas, sus capacidades y limitaciones, como reaccionan, como resuelven los problemas o como toman decisiones. Aún más, también tiene en cuenta dimensión social de las personas, es decir, cómo influye el uso de la tecnología en su contexto sociocultural y cómo afecta esta tecnología el entorno del usuario.
- El último de los aspectos clave de la IPO es el **diseño** , entendido como el proceso de idear soluciones a problemas de interacción, apoyándose para ello en los estudios elaborados sobre la tecnología y las personas. Cuando se diseña un producto interactivo lo que se hace es delimitar las posibles maneras de usarlo,

y este diseño es el que determina si la interacción resulta satisfactoria o no.

Paralelamente a la IPO aparece la ingeniería de la usabilidad (EU), lo intenta trasladar al entorno profesional el conocimiento teórico y metodológico de la IPO. Para algunos autores, IPO y EU representan dos caras de la misma moneda.

A diferencia de la IPO, la EU se ocupa menos de saber por qué funciona un diseño y se focaliza en la obtención de resultados, es decir, a demostrar si un diseño realmente funciona.

El concepto de **usabilidad** se refiere a la facilidad de uso de una aplicación interactiva o, como define el estándar ISO 9241: "Usabilidad es la medida en que un producto puede ser utilizado para determinados usuarios para conseguir unos objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso definido".

Si un usuario no encuentra como comprar un producto en un tiempo razonable es muy probable que intente comprarlo en otro lugar.

Un sitio usable intenta buscar que el usuario no tenga que pensar cómo está organizada la información, de tal manera que incluso se optimizan el número de clics a hacer para completar una tarea.

Otro aspecto a tener en cuenta a la hora de hacer el diseño de una interfaz web es la accesibilidad que ofrece. El objetivo es facilitar el acceso del sitio web a todas las

personas, independientemente del tipo de dispositivo, idioma, cultura, localización geográfica y capacidades físicas o cognitivas de los usuarios.

La **accesibilidad** trata el diseño de interfaces de usuario para que sean fáciles de percibir, operativas y comprensibles para personas con un abanico amplio de habilidades o con circunstancias, entornos y condiciones diferentes. En definitiva, la accesibilidad puede ser entendida como la usabilidad para todos.

Actualmente, en el diseño de interfaz utilizan metodologías que orientan el producto a sus usuarios finales, donde cada decisión tomada se basa en las necesidades, los objetivos, las expectativas, las motivaciones y las capacidades de los usuarios.

Así pues, el diseño de un sitio web no dependerá sólo de factores como el dispositivo desde el que se accede, la usabilidad, la accesibilidad, el diseño gráfico, la calidad de los contenidos o la utilidad, sino que también hace referencia a aquellos aspectos relativos a las emociones, los sentimientos ya la percepción de confianza en el producto o marca corporativa de la web, entre otros aspectos.

La **experiencia de usuario**, en inglés User Experience (UX), es una filosofía de diseño que tiene por objetivo la creación de interfaces que resuelven necesidades

concretas de sus usuarios finales, consiguiendo así una mayor satisfacción de estos, así como también una mejor experiencia de uso con el mínimo esfuerzo.

De este modo, el diseño basado en la experiencia de usuario utiliza metodologías de diferentes campos para estudiar la interacción entre el hombre y la máquina, como son la psicología, la sociología, la etnología, la antropología, la ciencia cognitiva y la estadística.

Véase en la figura [1.1](#) el diseño de la interfaz de un sitio web que tiene en cuenta aspectos sobre la experiencia de usuario. Se puede apreciar la construcción de interacciones digitales significativas gracias a la utilización de colores adecuados, movimientos que captan la atención pero que no distraen y que, en consecuencia, tienen como resultado una interfaz amigable. Sin embargo, cabe destacar que esta percepción de amigabilidad tiene una parte subjetiva.

F. . diseño web

Diseño de la interfaz de la página web del Ayuntamiento de Barcelona

El concepto de diseño

Diseñar es mucho más que simplemente acoplar u ordenar los elementos; es agregar valor y significado. Es decir, iluminar, simplificar, clarificar, persuadir y tal vez, incluso, entretener.

En este sentido, el **diseñador web** comprende perfectamente que un sitio web no debería estar hecho con el objetivo de "estar en Internet", sino con el ánimo de poner a disposición una herramienta que se corresponde con las necesidades del usuario que accede.

A menudo, algunos diseñadores web formarán parte de equipos de proyecto donde diferentes aspectos del proceso de diseño serán responsabilidad de varias personas, pero en otras ocasiones el diseñador web las deberá cubrir todas. Por ejemplo: diseñadores gráficos (logotipos, diseños y botones), especialistas de marketing y posicionamiento en Internet, redactores de Internet y diseñadores UX. Además, el diseñador web también debe estar informado de las últimas tendencias en usabilidad y accesibilidad.

1.2 Elementos de diseño: percepción visual

Los diferentes contenidos que compondrán el sitio pueden ser presentados y clasificados de múltiples formas. Por lo tanto, a la hora de escoger una habrá que atender una serie de consideraciones referentes a factores productivos, tecnológicos, perceptuales, culturales, artísticos y de innovación, entre otros.

En este sentido, el diseñador debe evaluar una serie de aspectos relacionados con factores humanos, así como también debe tratar de conseguir una composición equilibrada, utilizando técnicas básicas sobre la utilización del color, de la tipografía y de los iconos e imágenes a incluir en la web.

1.2.1 Factores humanos

Para el diseño de la interfaz web es importante considerar aspectos culturales y de el entorno del usuario final, para que las interfaces web se adecuen.

Así pues, por una parte, los factores culturales pueden influir en la percepción que un usuario pueda tener de un sitio web. Y por otra, es necesario que el diseñador muestre los contenidos utilizando la metáfora computacional correctamente y que no induzca a errores. Asimismo, debe tener en cuenta lo que los usuarios esperan encontrar en un tipo de diseño para cada servicio.

Así, puede ocurrir que una misma web tenga un diseño u otro, en función de los usuarios que, presumiblemente, la han de visitar. Por ejemplo, la biblioteca de Nueva York tiene un diseño diferente y especial para los niños que para los adultos:

- www.nypl.org
- kids.nypl.org

Y por otro lado, a la hora de decidir qué elementos formarán parte del diseño web es importante pensar desde donde se accederá. Es decir, si será desde un ambiente

metáfora

La función y la utilidad principal de las metáforas es favorecer la comprensión de una situación desconocida sobre similitudes que pueden establecerse respecto a una realidad conocida.

público o privado, si es posible que haya mucha luz o bien que sea probable que el usuario se conecte desde un ambiente con ruido.

Entonces resulta importante elegir una buena gama de colores para que sea visible en ambientes con mucha o poca luz, o decidir si es necesario que haya audio en la web o otros tipos de elementos interactivos que pueden influenciar en la atención del usuario.

Otro aspecto importante serán las posibles limitaciones físicas o cognitivas que el usuario pueda tener, y que condicionarán la accesibilidad en la web.

1.2.2 Elementos de percepción visual: colores

El color es uno de los recursos que tienen más impacto en la estética y el atractivo de una interfaz. Así pues, es una pieza clave en la percepción final del usuario.

Sin embargo, el color no tiene exclusivamente una función estética o emocional, sino que también tiene una función comunicativa importante. Además, cuando es usado de manera incorrecta puede ocasionar problemas graves de usabilidad en la interfaz.

La experiencia humana y la cultural nos dice que se asocian ciertos colores a emociones o sensaciones muy variadas. Por ejemplo, si se piensa en naturaleza seguramente el primer color que viene a la mente es el verde, pero de hecho habría miles de colores y tonalidades a escoger.

Además, no es recomendable utilizar un número excesivo de colores diferentes. Es aconsejable, pues, limitar el número a los que una persona pueda procesar de un solo vistazo; en función de la complejidad del diseño, este número giraría alrededor de cinco colores.

Para escoger los colores que pueden ser más adecuados para el diseño de un sitio web se pueden utilizar algunas técnicas, como el reconocimiento de colores, la combinación de diferentes colores o la utilización de colores complementarios.

Reconocimiento de colores

El color se utiliza como un recurso para reforzar el significado de ciertos elementos de la interfaz, sobre todo teniendo en cuenta el sentido de que determinadas culturas asocian a determinados colores. Así pues, la recomendación lógica es no incurrir en contradicciones que puedan interferir en la interpretación intuitiva del mensaje.

La figura [1.2](#) representa una mala utilización del color en el diseño de botones de una interfaz gráfica:

F. . Utilización del color

La utilización equivocada del color puede inducir al usuario a error.

Combinaciones de colores

Los diferentes colores del círculo cromático se pueden distinguir entre:

- **Color primario** : son el cian, el magenta y el amarillo.
- **Color secundario** : en medio de los tres colores primarios tenemos los tres colores secundarios, que son el rojo, el verde y el azul
- **Color terciario** : para completar los doce colores de la rueda cromática, los

terciarios son los que se encuentran situados en medio de un de primario, y los resultantes de mezclar un color primario con uno secundario adyacente.

- **Color neutro** : el blanco y el negro no se suelen considerar colores y no aparecen el círculo cromático. El blanco es la presencia de todos los colores y el negro la ausencia total de color, y al combinarse el uno y el otro forman las diferentes tonalidades de gris.

círculo cromático

La teoría sobre percepción del color y, en concreto, la teoría sobre los procesos opoentes (Hurvich y Jamesonm 1957; Webster, 1996), dice que la visión canaliza el color en tres canales diferentes: un codifica la luminancia (blanco-negro) y los demás dos, el color (rojo-verde y azul-amarillo). En consecuencia, encontramos pares de colores que no se pueden ver a la vez, como el rojo con el verde, el azul con el amarillo y el negro con el blanco (acromático).

Combinación de colores primarios

Una aplicación de esta teoría es la recomendación general de evitar la combinación de colores incompatibles (rojo, verde, azul y amarillo), principalmente de los puros o saturados. Por ejemplo, poner letras en color azul sobre un fondo amarillo puede inhibir la visibilidad, o el texto en rojo sobre un fondo azul puede parecer que vibra (Cañas, Salmerón y Gámez, 2001).

A menudo nos interese saber qué es un color determinado que se ve por pantalla. Aunque existe *software* específico para reconocer colores, esta tarea también se puede llevar a cabo con un editor de imágenes, como puede ser el GIMP o Adobe Photoshop, o bien también podemos usar la aplicación *online* Color Schemer: www.colorschemer.com.

Además, para saber cuál sería la combinación de colores que utiliza una determinada imagen se puede utilizar la aplicación *online* siguiente: whatsitscolor.com.

Otra herramienta online que permite crear una combinación de 5 colores a partir de un de solo seleccionando diferentes tipos de combinaciones: análogos, elegida, monocromática, etc, es Adobe Color CC: color.adobe.com.

fente tipográfica

Es un estilo o una apariencia de un grupo completo de caracteres, números y signos regidos por unas características comunes.

1.2.3 Elementos de percepción visual: tipografía

La tipografía es el arte y la técnica de crear y combinar tipos de fuentes para comunicar un mensaje. También se ocupa del estudio y la clasificación de diferentes fuentes tipográficas.

La tipografía tiene un papel crucial en el diseño, ya que la mayor parte del contenido está expresado a través del texto. Sin embargo, hay que tener en cuenta que muchos de los usuarios que accederán a la web harán un barrido en lugar de leer la totalidad del texto.

En consecuencia, los diseñadores deben tener presente qué tipo de tipografía es adecuada para la imagen y personalidad del sitio web, así como también para su

fente digital

Es un conjunto de dibujos vectoriales que se pueden escalar sin pérdida de calidad. Se almacenan en archivos de

La **legibilidad** se refiere a la facilidad con que se puede leer el texto, factor que depende de la presentación visual, es decir, de la claridad de las letras individuales. La **inteligibilidad**, en cambio, se refiere a la facilidad con que se puede interpretar y comprender el mensaje, factor que tiene en cuenta la longitud de las palabras o de las frases.

Para elegir un tipo de tipografía u otra para el diseño del sitio web será necesario seguir unas determinadas pautas estándar.

Familia de la fuente

Se denomina *familia de fuente* el conjunto de caracteres basado en una misma fuente con algunas variaciones, como pueden ser el espesor o la anchura, pero siempre manteniendo características comunes.

Actualmente, los navegadores web pueden utilizar cualquier fuente, aunque muchos diseñadores siguen utilizando las fuentes más comunes. En concreto, se define una lista de fuentes que se utilizarán en el orden establecido; es decir, si la primera fuente no está disponible se utilizará la segunda, y así sucesivamente.

Los grupos de fuentes más usados son:

- Verdana, Geneva, sans serif
- Georgia, Times New Roman, Times, serif
- Courier New, Courier, monospace
- Arial, Helvetica, sans serif
- Tahoma, Geneva, sans serif
- Trebuchet MS, Arial, Helvetica, sans serif
- Arial Black, Gadget, sans serif
- Times New Roman, Times, serif
- Palatino Linotype, Book Antiqua, Palatino, serif
- Lucida Sans Unicode, Lucida Grande, sans serif
- MS Serif, New York, serif
- Comic Sans MS, cursive

Los tipos de fuente se pueden dividir entre las fuentes *serif*, o de gracia, y las fuentes *sans-serif*, o de palo seco. La diferencia es que las fuentes de gracia presentan trazos

Puede encontrar catálogos de fuentes en las direcciones siguientes:

- fonts.google.com
- typedia.com

decorativos en los extremos del trazo de cada letra y las de palo seco, no.

Hay autores que defienden que para mejorar la legibilidad es mejor representar los textos digitales en fuentes sin ornamentos, aunque actualmente, para compensar un diseño web minimalista, hay una tendencia creciente a utilizar tipografías originales y cuidadas como un elemento más y fundamental en el diseño de la página web.

Recomendaciones para la tipografía

El factor más importante de la tipografía es el público al que va dirigido, como también lo es que éste sea capaz de leerlo perfectamente sin ningún inconveniente. en este sentido, el diseñador debe tener en cuenta qué tipo de usuario lo leerá y desde qué dispositivo.

Los efectos web que se pueden aplicar a las tipografías siguen siendo bastante limitados. Por defecto, las características que se pueden aplicar en el texto de una página web para definir el aspecto que tendrá y mejorar la legibilidad son:

- **Tipo** de fuente, es decir, su aspecto.
- **Tamaño** de la fuente.
- **Estilo** , que puede ser, normal, negrita, itálico, subrayado ...
- **Mayúscula, minúscula o versalita** .
- **Espaciado** entre letras, palabras y párrafo; se refiere a la distancia que hay entre estos.
- **Interlineado** : hace referencia a la distancia entre líneas.
- **Alineación** : si es izquierda, centrada, derecha o justificada.
- **Contraste de color** entre el de la tipografía y el color de fondo.
- **Sombreados y gradientes** de color.

Aparte de elegir un tipo de fuente adecuada para el texto del diseño, también se pueden seguir las siguientes recomendaciones, entre otros:

- **El tipo de fuente** : una fuente demasiado ancha o demasiado condensada pierde legibilidad. Se puede usar el modo condensado para informaciones secundarias que han de ocupar poco espacio.
- **El tamaño de la fuente** , que normalmente se expresa en píxeles. A modo de orientación, un tamaño de 11 o 12 píxeles facilita la lectura de la mayoría de tipografías, y casi nunca se trabaja con cantidades inferiores a 8 o 9 píxeles. Opcionalmente, también se puede definir el tamaño como un porcentaje de la tipografía media.

- Con referencia **al estilo** del texto, se puede decir que el **itálico** se utiliza para remarcar una palabra dentro de un bloque de texto para que destaque del resto, al tiempo que un texto escrito todo en estilo **itálico** resulta pesado de leer. Del mismo modo, un texto en **negrita** aumenta su espesor y también se puede utilizar para destacar. Se debería evitar el uso del **subrayado**, ya que dificulta la lectura.
- Las **mayúsculas** tienen una alineación horizontal homogénea que uniformiza las letras. Las minúsculas con caracteres individualizados facilitan la lectura.
- **Espaciado** entre letras y palabras: un espaciado reducido puede percibirse como un montón de letras amontonadas, mientras que si es demasiado amplio se puede percibir como un texto fragmentado.
- Si el **interlineado** es demasiado estrecho obtenemos un texto abarrotado con difícil lectura. Si es demasiado ancho interrumpe continuamente la lectura, dado que obliga a atravesar espacios en blanco continuamente. Se recomienda incrementar el interlineado un 150% más que en papel, especialmente en tipografías sans-serif, que no poseen la guía horizontal que forman las *gracias* y que ayudan al ojo a seguir el flujo del texto.
- La **alineación** a la izquierda y la justificada son las más legibles, aunque se prefiere la primera, ya que la justificada puede provocar espacios entre palabras variables o demasiado grandes. Para textos cortos se puede usar la centrada o a la derecha.
- **Contraste de color** : la utilización de colores similares entre texto y fondo en dificultarán la lectura. Se debe buscar un contraste fuerte, colores claros con oscuros o colores muy saturados con poco saturados. Por ejemplo, los tipos negras sobre fondo blanco aportan mayor legibilidad; en cambio, el texto blanco sobre fondo negro nos hacen perder visibilidad.
- **Ancho de columna** : gracias a la tendencia del mercado del uso de pantallas panorámicas, la mayoría de webs tienen el doble de anchura del alcance adecuado del ojo, por lo que exige un esfuerzo extra para leer el texto. Un tamaño estándar sería entre 12 y 15 palabras por columna. una columna muy estrecha fragmenta el texto, y en una masa ancha el lector encuentra dificultad para encontrar la línea siguiente.

Otros aspectos técnicos

El texto web contiene muchas limitaciones, como por ejemplo que la mayoría de navegadores no pueden representar texto en 3D o con un ángulo de rotación. por otra parte, también presenta limitaciones a la hora de adaptarse automáticamente alrededor de bloques rectangulares que estén alineados a la derecha o a la izquierda.

En el diseño web, una solución sería representar el texto como una imagen, lo que nos permite crear cualquier efecto que queramos darle.

Aunque, esta técnica se podría utilizar para títulos, menús u otras partes de la web, se desaconseja su utilización por los siguientes motivos:

- Los buscadores no podrían indexar este texto. Si el texto fuera corto, un pequeño remedio sería poner un texto alternativo detrás de la imagen.
- La presentación de este texto se convertiría estática, no se podría adaptar automáticamente en el espacio que haya disponible.
- Si la página coge el texto de una base de datos sería muy complicado utilizar imágenes para representar este texto.
- El usuario no puede aumentar el texto, y en algunos casos puede tener problemas de legibilidad.

Por lo tanto, hay que tener en cuenta todos estos aspectos a la hora de escoger un texto con una tipografía u otra.

El color de la tipografía

Conocer y comprender qué transmiten los colores es importante y básico. Así pues, para conseguir una buena legibilidad tendremos que equilibrar cuidadosamente las tres propiedades del color: tono, valor e intensidad, sin olvidar que el contraste entre las letras y su fondo deberá ser adecuado.

Las sensaciones que determinados grupos de colores nos pueden transmitir son:

- Los colores cálidos, que van del rojo al amarillo, son colores muy impactantes, ya que destacan mucho sobre un fondo. Por este motivo se utilizan para llamar la atención.
- Los colores fríos son los verdes y la gama de azules, y tienen la característica que son relajantes.
- El blanco, el gris y el negro no pueden incluirse ni en el grupo de cálidos ni en el de fríos.
- Los marrones, el crema y el ocre representan añoranza y se utilizan para representar productos naturales y clásicos.
- Los colores primarios (magenta, cian y amarillo) se utilizan para productos infantiles, porque aportan alegría y juventud.
- Las tipografías doradas o plateadas sobre fondo oscuro denotan elegancia y sofisticación.

Es importante destacar que, a primera vista, la tipografía y el color dicen más que el mismo texto.

1.2.4 Elementos de percepción visual: iconos

mejora la estética y el atractivo visual de las interfaces. El uso de iconos se basa en representaciones equivalentes del mundo físico o convenciones para explicar un concepto o una idea. Además, puede mejorar la usabilidad del producto, pero cuidado: cuando están mal diseñadas también pueden provocar problemas graves en la interacción con el usuario.

En la figura [1.3](#) puede ver diferentes ejemplos de iconos para diferentes acciones cotidianas, como pueden ser, imprimir, salir, buscar, etc.

F. . Ejemplos de iconos cotidianas: imprimir,
buscar, correo electrónico, salir, información, teléfono

A pesar de que se piense que los iconos se reconocen más rápidamente que el texto, puede que no sea así; incluso con experiencia de uso, la velocidad de reconocimiento de iconos es la misma que la de textos.

La **efectividad** de un icono se puede medir con la facilidad con que el usuario en puede interpretar el significado.

A menudo, puede ocurrir que usuarios no expertos no utilicen algunas de las opciones que presenta la interfaz para que no consiguen interpretar correctamente el significado de los iconos. Por ello, en algunas ocasiones es necesario etiquetar iconos con un texto. Ver un ejemplo en la figura [1.4](#).

F. . Ejemplo de iconos con una etiqueta junto

Fuente: vucling.com

1.2.5 Elementos de percepción visual: composición del sitio web

A la hora de empezar una composición, lo primero que hay que saber es el espacio de que se dispone y elegir qué elementos gráficos y textuales usarán. Es en este momento cuando se deben definir qué partes de la pantalla deben estar ocupadas y

para qué elementos; hay que tener en cuenta aspectos como los siguientes:

- Si se dejan muchos espacios vacíos la composición puede resultar descoordinada, ya que puede ser complicado establecer relaciones globales entre los elementos o grupos de elementos.

- Si el número de elementos es elevado podemos incurrir en un diseño sobregato y difícil de entender, en el que no se distingue con claridad qué es cada cosa y cuál es el mensaje que se quiere transmitir.
- También hay que tener en cuenta la agrupación lógica de los elementos.

La **composición** se define como una distribución equilibrada de todos los elementos que se incluyen en un diseño. En este sentido, la composición del diseño de un sitio web debe ser estética y funcional.

Un mapa de calor es una representación gráfica que da información sobre los focos de atención estimados los usuarios.

Si seguimos las directrices que han analizado la manera que tiene un usuario de recorrer una página web, se deduce que hay zonas más destacadas que otras. para visualizar esta información podemos recurrir a lo que se denomina **mapa de calor** . También, la composición de diseño que un documento web puede seguir lo que se denomina **patrón basado en F** .

Véase un ejemplo en la figura [1.5](#) y la figura [1.6](#).

F. . Mapa de calor

F. . patrón

También se puede optar por otros patrones de diseño, como el **patrón basado en Z**, también utilizado en muchos sitios web. En muestra un ejemplo en la figura [1.7](#).

F. . Patrón de diseño Z

Fuente: <http://webdesign.tutplus.com/articles/understanding-the-z-layout-in-web-design-webdesign-28>

Para distribuir el espacio de nuestra página web de manera adecuada y dar equilibrio de proporciones a los elementos se pueden utilizar técnicas básicas de equilibrio visual, así como también se puede tener en cuenta la **divina proporción** y la **regla de los tercios**.

equilibrio visual

Para conseguir un diseño equilibrado se pueden seguir unas normas básicas basadas en:

- **La colocación** de los diferentes elementos que conformarán el diseño. así podemos obtener un diseño con equilibrio simétrico, el cual nos transmite una sensación de orden, o bien un equilibrio asimétrico, que nos transmite agitación, tensión, dinamismo, alegría y vitalidad.
- **El contraste** de tono o color de los elementos del diseño. Los colores más oscuros,

puros y saturados tienen más peso visual que los colores secundarios o neutros, y estos mayor que los terciarios o poco saturados. Así pues, si se quiere que un elemento pierda protagonismo necesario disminuir la intensidad de su tono y también su tamaño, para que así no se pierda equilibrio en la composición.

- **La proporción** de los elementos, de tal forma que los elementos más grandes, altos o anchos tienen una carga visual superior a la de los más pequeños, bajos o incluso, lo que crea zonas de atracción más intensas.

Ejemplo de utilización del contraste de
to

Por lo tanto, hay que definir una jerarquía visual clara, de tal forma que los elementos o los contenidos más relevantes deben estar enfatizados y deben destacar sobre la resto de la interfaz. Con ello se refuerza el mensaje, ya que se orienta y se facilita la exploración visual del usuario.

divina proporción

La divina proporción está basada en la regla áurea (figura [1.8](#)), La cual nos puede ayudar a decidir las medidas de los espacios que conformarán nuestra web para proporcionar armonía estructural.

F. . regla áurea

La **ratio** de la divina proporción se establece en 1,618.

Así, por ejemplo, si se quiere crear un diseño de ancho fijo, 960 px, y se quiere dividir el espacio en dos columnas, se dejará un tamaño más grande para el contenido y otra de más pequeña para la barra lateral.

Si quiere asegurarse de que el ratio entre las dos columnas es 1,618, se puede calcular la anchura de cada una de la siguiente manera:

- Dividir el tamaño 960 px entre 1,618. El resultado es **593 px** , aprox.
- Restar 960 px y 593 px. El resultado es **367 px** , aprox.

Véase la figura [1.9](#) el resultado del uso de la propiedad de la divina proporción.

Regla de los tercios

Generalmente, para colocar correctamente el mensaje más importante o la funcionalidad del lugar se puede utilizar **la regla de los tercios** .

Esta regla se aplica dividiendo la composición en 9 partes iguales, de tal manera que se divide la composición mediante dos líneas horizontales y dos verticales equidistantes. Los cuatro puntos que forman la intersección de estas líneas se pueden

usar para situar los elementos más importantes. Por lo tanto, alinear una composición de acuerdo con la regla de los tres tercios crea más tensión, energía e interés.

A menudo puede resultar imposible usar los cuatro puntos para remarcar el contenido o las funciones más importantes, pero podemos emplear alguno. De hecho, el punto de arriba a la izquierda es el más relevante, ya que los usuarios solemos inspeccionar las páginas web de acuerdo con el patrón basado en F.

Veamos en la figura [1:10](#) cuáles son los puntos calientes siguiendo la regla de los tercios.

F. . Aplicación de la regla de los tercios en una composición

1.3 Lenguajes y aplicaciones de diseño web

El desarrollo de un sitio web no es sólo la creación de una página web, sino que en este proceso intervienen diversos especialistas, como diseñadores, programadores, comerciales, *community managers*, redactores de contenidos, especialistas SEO, gestores de servidores, etc.

Un **sitio web** (en inglés *Website*) es una colección de páginas web, imágenes, vídeos y otros elementos digitales servidos por un único dominio o subdominio en el World Wide Web. Una **página web** es un documento HTML / XHTML que es accesible generalmente mediante el protocolo HTTP de Internet.

La creación de páginas web requiere conocer diferentes lenguajes de programación web como HTML5, CSS3 y JavaScript, así como también conocimientos de algunas normas básicas de usabilidad y accesibilidad.

Search Engine Optimization, SEO

Proceso que tiene por objetivo aumentar la visibilidad de un sitio web a los motores de búsqueda, incrementando su posición en la página de resultados.

HyperText Transfer Protocol, HTTP

El protocolo de transferencia de hipertexto establece como efectúa el intercambio de documentos de hipertexto y multimedia en la web. dispone de una variante cifrada mediante SSL llamada HTTPS.

1.3.1 Lenguajes de desarrollo web

Los lenguajes de desarrollo web son:

- HTML
- CSS
- JavaScript

HTML

Desde la aparición del World Wide Web, a principios de los años noventa, el HTML ha evolucionado hasta convertirse en un lenguaje de marcas relativamente eficiente, el cual, cuando se utiliza junto con elementos JavaScript y CSS, puede utilizarse para crear sitios web y aplicaciones interactivas visualmente sorprendentes.

Para poder transferir un documento entre equipos, en este caso una página web, es necesario codificar la información para su envío. Como la mayoría de información en enviar es texto, se utiliza la codificación ASCII, que relaciona cada carácter con una combinación numérica en código binario.

En un documento HTML, se envía junto con la información el formato correspondiente de

la misma, como atributos de color, tamaño, alineación, etc. El lenguaje HTML propone una serie de marcas o etiquetas que insertadas junto con la información establecen su formato.

El lenguaje HTML es el resultado del trabajo de W3C (World Wide Web Consortium). Sus trabajos se publican en diferentes etapas:

- Los **borradores**, Working Draft, se publican para compartir los avances de las investigaciones con el resto.
- Las **especificaciones** se publican cuando el lenguaje está casi terminado y listo para ser usado.
- Las **recomendaciones** se publican cuando el lenguaje está oficialmente aprobado y acabado.

Logotipo de W3 Consortium.

DHTML, Dynamic HTML
Designa el conjunto de técnicas que permiten crear espacios web interactivos y animados utilizando la combinación de lenguaje HTML estático, lenguaje interpretado por parte del cliente (como el JavaScript), hojas de estilo en cascada (CSS) y el Documento Object Model (DOM).

El Document Object Model (DOM) representa el documento creado a partir de las etiquetas HTML y los estilos CSS que el navegador mostrará en la pantalla.

El estándar de W3C que actualmente utilizan los desarrolladores para la creación páginas web es el HTML5.

La evolución del lenguaje HTML

El origen de HTML se remonta a finales de la década de 1980, cuando el físico Tim Berners-Lee, investigador del CERN (Organización Europea para la Investigación Nuclear) proponía un nuevo sistema "hipertexto" para compartir documentos.

A partir de 1996, los estándares de HTML los publica otro organismo de estandarización, el W3C (World Wide Web Consortium). Su primera recomendación fue la versión HTML 3.2, que se publicó el 14 de enero de 1997.

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

XHTML 1.0 es una adaptación de HTML 4.01 al lenguaje XML, así que mantiene casi todas las etiquetas y características pero añade algunas restricciones y elementos propios de XML. La versión XHTML 2.0 nunca llegó a ser una recomendación para W3C, y ante el su fracaso, a finales de 2006, tomó como base las investigaciones de [WHATWG](#) para seguir trabajando con HTML5. Así que HTML5 ya es un estándar recomendado por la W3C desde el 6 de agosto de 2013. La última recomendación se puede consultar en www.w3.org/TR/html5.

En julio de 2012, ante las dificultades del trabajo en común y la diferencia de objetivos respecto de W3C, el WHATWG decide [separarse y seguir su propio camino de forma totalmente independiente](#). La consecuencia es que, a partir de entonces, el WHATWG propondrá una versión "viva" del HTML y el W3C llevará a cabo la normalización del lenguaje.

El WHATWG continúa con sus investigaciones independientemente del W3C, y al HTML se le llama [HTML Living Standard](#) para no confundirlo con el [HTML5](#) de W3C.

Esto complica la tarea de todos los desarrolladores web, ya que deben conocer los elementos HTML5 propuestos por WHATWG y comprobar si han sido validados por W3C y son reconocidos por los navegadores.

Mientras que HTML nos permite definir la estructura de una página web, las hojas de estilo en cascada (Cascading Style Sheets, CSS) son los que ofrecen la posibilidad de definir reglas y estilos que especifican cómo mostrar los diferentes elementos HTML en diferentes tipos de dispositivos, como son pantallas de ordenadores, móviles, tabletas, etc.

CSS está diseñado principalmente para permitir la separación de contenido del documento (escrito en HTML o en un lenguaje de marcas similar) de la presentación del documento, incluyendo elementos como la disposición, los colores y las fuentes.

Esta separación puede mejorar la accesibilidad al contenido, proporcionar más flexibilidad y control en la especificación de características de presentación y permitir que múltiples páginas compartan un formato común.

Así pues, las hojas de estilo permiten definir de manera eficiente como se debe mostrar una página web, y es uno de los conocimientos fundamentales que todo diseñador web debe conocer para llevar a cabo su trabajo.

La primera versión de CSS fue publicada por W3C a finales del 1996, y fue ganando popularidad hasta llegar al estándar actual CSS3.

JavaScript

JavaScript es un lenguaje de programación interpretado y orientado a objetos. Se utiliza junto al cliente, es decir, que se ejecuta en el navegador web, lo que permite mejoras en la interfaz de usuario. Su uso más común es escribir funciones que se incluyen en las páginas web y que interactuamos con el DOM de la página.

AJAX, Asynchronous
JavaScript And XML, es una
técnica de
desarrollo web para
crear aplicaciones
interactivas.

Tradicionalmente se utilizaba en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Hoy en día es ampliamente utilizado para enviar y recibir información del servidor con la ayuda de otras tecnologías como AJAX.

JavaScript fue desarrollado originalmente por Brendan Eich de Netscape el diciembre de 1995, y en 1997 fue adoptado como un estándar ISO.

1.3.2 Herramientas de desarrollo. editores HTML

Con el fin de desarrollar un sitio web es útil usar un **editor o gestor de HTML**. Este tipo de software ofrece facilidades para la edición de documentos HTML, y además también puede proporcionar funcionalidades avanzadas para gestionar el código y administrar el sitio web.

Editores WYSIWYG (What
you see is what you get)

Permiten generar el código de los documentos HTML a partir de una interfaz gráfica con la que se dibuja la página resultante. son muy sencillos de usar, pero pueden generar mucho código no deseado (Etiquetas vacías, saltos de línea, etc.).

Los snippets son partes de código fuente reusables. así se minimiza el uso de código repetido que es común en varias funciones.

Existen varias opciones para la edición del código de un sitio web: editores de texto, editores en línea o bien IDE. Sin embargo, las funcionalidades más importantes que estos incluyen, y que por ello son adecuados para el desarrollo web, son:

- Creación de documentos nuevos de una manera rápida y cómoda.
- Gestión de plantillas de varios documentos.
- Sintaxis coloreada para una mejor legibilidad de los documentos.
- Validación del código.
- Acceso rápido al código con la ayuda de marcadores.
- Configuración del sangrado y del estilo del código.
- automático del código con cierre automático de etiquetas.
- Inserción de etiquetas y atributos mediante auxiliares.
- Gestión de fragmentos de código reusable, en inglés *snippets* .
- Gestión de proyectos.
- Conectores adicionales que dan más funcionalidades al gestor.

Posteriormente, para la visualización de la página web desarrollada utiliza el **navegador web** . Asimismo, este incorpora una serie de herramientas para los desarrollos dorso que ayudarán en la depuración del código de la página.

editores HTML

Hoy en día hay una gran cantidad de editores de HTML que permiten el desarrollo de un sitio web, todos los cuales tienen características comunes. algunos de los

editores de HTML con los que podemos trabajar actualmente son Brackets, Sublime Texto, Atom, etc.

A continuación veremos algunas de las características más relevante de estos:

1. Brackets : permite ver dos o más archivos a la vez mediante la función *Inline*

Editores . También incorpora la funcionalidad *Live Preview* , que permite visualizar el aspecto de la página web a medida que se va desarrollando. Además, facilita la manera de encontrar, añadir o modificar las correlaciones entre documentos HTML y CSS gracias a la funcionalidad *Preprocessor Support* . Es de código abierto, compatible con GitHub, y se encuentra disponible para Windows, Linux y Mac OS. el podemos descargar en: brackets.io.

En la figura [1:11](#) vemos una captura de pantalla de su interfaz.

Logotipo del editor Brackets

Git

Git es un software de sistema de control de versiones pensado en la eficiencia y confiabilidad de

Github

Plataforma de desarrollo
colaborativo para alojar
proyectos utilizando el sistema de
control de versiones lluvias.

2. Sublime Text : dispone de muchos pedidos de teclado que agilizan el acceso a las funciones y las características que ofrece. Asimismo, también cuenta con *plugins* y complementos que están disponibles para lenguajes de programación específicos. Su interfaz es en pestañas y muestra una vista en miniatura para visualizar en qué parte del código está trabajando. Se distribuye como un *software* de evaluación, se encuentra disponible para Windows, Linux y Mac OS X y es compatible con GitHub. Se puede descargar en: www.sublimetext.com.

En la figura [01:12](#) podemos ver una captura de pantalla de su interfaz.

F. . Interfaz de trabajo del editor Sublime Text

3. Atom : proporciona las funciones más importantes para el desarrollo web, está integrado con herramientas de navegador y también alplugitHub. Es de código abierto y está disponible para Windows, Linux y Mac OS X. El podemos descargar en: atom.io.

En la figura [1:13](#) se puede ver una captura de pantalla de su interfaz.

F. . Interfaz de trabajo del editor Atom

IDE, Integrated Development Environment

Aplicación informática que proporciona servicios integrales para facilitar el desarrollo de *software*. normalmente incorpora un editor de código fuente, herramientas de construcción automáticas y un depurador.

También se puede optar por desarrollar el sitio web en un IDE, el cual normalmente ofrece un entorno de desarrollo que agrupa diferentes funciones: editor de código, compilador, depurador, etc. Se aleja del concepto de editor ofreciendo una solución completa especializada en ciertos lenguajes de programación, entre los que los lenguajes de programación web. Algunos ejemplos de IDE que permiten la elaboración de páginas web son Netbeans, disponible en netbeans.org; Eclipse, disponible en eclipse.org/downloads, O CodeLite, disponible en codelite.org.

En esta unidad se puede utilizar cualquiera de las herramientas mencionadas, ya que las Sus características hacen que la tarea de diseño web sea mucho más agradable y productiva.

Editores en línea

A la hora de realizar el diseño de una página web se puede optar por el uso de herramientas en línea que permiten escribir el código directamente desde el navegador, como pueden ser CODEP, Online HTML Editor o Real-Time HTML Editor, entre muchos otros. Sus características principales son las siguientes:

1. CODEP : esta herramienta está enfocada a crear un prototipo rápido y probar de fragmentos de código. Así como, también permite que varios usuarios colaboren en un mismo proyecto y que compartan fragmentos de código. La encontramos disponible en: codepen.io/pen.

CODEP ofrece una interfaz de usuario y funcionalidades bastante intuitivas y fáciles de usar. Permite actualizar rápidamente el código HTML, CSS o JavaScript y obtener resultados inmediatos, así como también guardar y compartir fragmentos de código a través de enlaces personalizados, e incentiva la colaboración ofreciendo la oportunidad de contribuir con otros desarrolladores.

En la figura [1:14](#) se puede apreciar un ejemplo de utilización de CODEP.

2. Online HTML Editor : este editor permite generar el código de los documentos HTML a partir de una interfaz gráfica. Consta de dos partes, una para crear la página web mediante un interfaz gráfica (a la izquierda), la que permite dar formato al texto, añadir imágenes, etc., y otra para escribir directamente el código HTML (a la derecha). La encontramos disponible en: html-online.com/editor.

En la figura [1:14](#) se puede apreciar un ejemplo de utilización de este editor.

F. . Utilización de Online HTML Editor

3. BDS Editor Online : permite visualizar diferentes paneles donde se puede incluir código HTML, JavaScript o CSS. También permite una previsualización de la página web. Está disponible en: jseditor.proyectosbds.com.

En la figura [01:16](#) se puede apreciar un ejemplo de utilización de este editor.

1.3.3 Navegadores web

Para acceder a un determinado sitio web necesitamos un determinado software que nos permita el acceso y que se llama navegador web.

Logotipos de los navegadores más importantes

Un **navegador web**, en inglés *browser*, es un software informático que proporciona al usuario la visualización de documentos de hipertexto junto con otros tipos de archivos multimedia, como son imágenes, vídeos, sonidos y animaciones.

Actualmente, los navegadores más utilizados son Internet Explorer, Mozilla Firefox, Google Chrome, Safari y Opera.

Hay que tener en cuenta que no todos los navegadores interpretan el código HTML y CSS del mismo modo, y que entre ellos existen variaciones en la visualización de la página web que hacen que el resultado no sea el mismo con uno que con otro. Además, algunas de estas diferencias son tan importantes que pueden hacer que algunas partes de la página no se vean o no funcionen.

Elementos como los títulos o los párrafos pueden tener un tamaño, un interlineado o un margen diferente en un navegador y en otro.

Algunas consideraciones a tener en cuenta para mejorar la compatibilidad de un sitio web con los diferentes navegadores son:

- **Validar el código del sitio web** : es muy recomendable validar el código del conjunto de las páginas que conforman un sitio web, ya que, si no está bien escrito, un navegador puede interpretar el código de manera diferente y en consecuencia el resultado puede no ser el esperado.
- **Inicializar los estilos CSS** : por mucho que no utilizamos estilos, cada elemento HTML tiene unos estilos CSS por defecto, y éstos no son exactamente iguales en todos los navegadores. Así, si se inicializan estos valores por defecto, se controla el aspecto de cada elemento de manera más efectiva. se pueden obtener algunos ejemplos de hojas de estilos para restablecer los estilos del navegador a: cssreset.com.

- **Usar técnicas soportadas** : los lenguajes HTML5 y CSS3 ofrecen muchas técnicas para mejorar el aspecto de las páginas web, pero no todas están soportadas por todos los navegadores.

Las características de los navegadores que se tienen en cuenta a la hora de compararlas entre sí son:

- uso de los recursos, memoria RAM y CPU
- tiempo de arranque, tanto en frío como en caliente
- tiempo de carga
- rendimiento JavaScript
- rendimiento gráfico
- cumplimiento de estándares

Actualmente, existen herramientas como Acid Test, que ponen a prueba los navegadores para saber si cumplen los estándares web. Puede encontrar más información en: acid3.acidtests.org.

Herramientas para el desarrollador web

Los navegadores más importantes ofrecen una serie de herramientas que facilitan el desarrollo de un sitio web. De este modo, Chrome ofrece las **Google Chrome Dev Tools** , y Firefox permite incluir la extensión **Firebug** .

Las funciones más importantes que ofrecen son:

- Depurar el código Javascript.
- Inspeccionar el DOM.
- Analizar el tiempo de ejecución de cada función para optimizar el rendimiento.

Para acceder se selecciona el icono situado en la esquina superior derecha del navegador y luego se hace clic en las herramientas para los desarrolladores. Entonces aparece la aplicación para inspeccionar el contenido de la página web en una ventana lateral o debajo.

Véase en la figura [01:17](#) y la figura [1:18](#) la visualización de estas herramientas en los navegadores Chrome y Firefox, respectivamente.

F. . Vista del inspector en el navegador Chrome

F. . Vista del inspector en el navegador Firefox

Desde estas herramientas se puede seleccionar cualquier elemento HTML o estilo CSS y modificarlo para ver cómo queda. Hay que tener en cuenta que los cambios que se hagan no se guardan en el documento HTML.

1.4 Diseño web con HTML5

El lenguaje de marcas HTML5 es la versión actual del lenguaje HTML. es el lenguaje que se utiliza para el diseño de páginas web y hace referencia a un estándar desarrollado por W3C.

El estándar HTML5 contiene un conjunto amplio de tecnologías que permite crear sitios web, así como aplicaciones web de envergadura. algunas características de HTML5 agrupadas por su función son:

- **Semántica** : permite describir con mayor precisión cuál es su contenido.
- **Rendimiento e integración** : Proporciona una mayor optimización de la velocidad y un mayor uso del hardware a través de la introducción de un diseño

- **CSS3** : lenguaje de hojas de estilos que nos ofrece una gran variedad de opciones para hacer diseños más sofisticados
- **Multimedia** : nos proporciona un excelente soporte para utilizar contenido multimedia, como son audio y vídeo nadiuament.
- **Gráficos y efectos 2D / 3D** : proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la web, como son canvas 2D, WebGL, SVG, etc.

Lo que se llama *web semántica* es un concepto que propone la inclusión de metadatos semánticas en la web, para que agentes informáticos, como pueden ser los navegadores o robots de los motores de búsqueda, puedan procesar su contenido, razonar con este, combinarlo y realizar deducciones lógicas para resolver y dar una respuesta más acertada. Así pues, los elementos que la especificación HTML5 aporta para poder estructurar un documento web con semántica estandarizada son: `<header>` , `<footer>` , `<section>` , `<aside>` , `<nav>` , `<artículo>` , `<hgroup>` , `<time>` , etc.

El lenguaje HTML5 también permite realizar diseños web adaptativos, en inglés *Responsive Web Design* . Así pues, dispone de técnicas para que una web sea visible de manera óptima en todo tipo de dispositivos capaces de visualizar páginas web, es decir, desde móviles hasta ordenadores o tabletas. Para ello se utilizan estructuras con imágenes fluidas y hojas de estilos CSS.

'Web crawler' o robot web
Software que inspecciona páginas web de manera metódica y automatizada. Se utiliza para poder referenciar las páginas visitadas a través de un motor de investigación, como puede ser Google, Yahoo, Bing, etc.

1.4.1 Formato del HTML5

Los objetivos del HTML5 son la simplicidad y la permisividad. Así pues, HTML5 retoma la sintaxis permisiva de HTML4 y hace desaparecer las reglas estrictas de sintaxis de XHTML. Sin embargo, es recomendable seguir algunas de las normas de sintaxis de XHTML:

- Usar sólo minúsculas.
- Usar las comillas dobles para indicar el valor de los atributos.
- Cerrar todas las etiquetas.
- Sangrar el código para aumentar la legibilidad.

El **formato** de una etiqueta HTML es:

```

1  <NomEtiqueta ATRIBUT1 = "valor ATRIBUT1" ... atributN = "valor
    atributN ">
2      Elementos afectados por la etiqueta
3  </NomEtiqueta>

```

Cada etiqueta puede tener una serie de atributos que modifican el comportamiento de la misma, por ejemplo:

Ejemplo de formato de una etiqueta

```

1  <!-- En la siguiente línea definimos un párrafo -->
2  <P> Esta es la primera página web </P>
3
4  <!-- Fijamos - nos en el cierre de la siguiente etiqueta,
    directamente con la /
5  -->
6  <Br />
7
8  <!-- En la siguiente línea estamos definiendo un link -->
9  <A href = "https://ioc.xtec.cat/educacio/" target = "_blank"> Link IOC
    </A>

```

Donde a es el nombre de la etiqueta.

Donde href es el atributo a través del cual definimos donde ha de llevar el enlace.

Donde target permite especificar dónde se debe abrir el enlace; **_blank** significa que se abrirá en una ventana.

Donde Link IOC es el texto que hace de link y que se ve afectado por la etiqueta **<a>**

**Donde ** sirve para cerrar la etiqueta y marcar donde deja de tener efecto la misma.

Como se puede apreciar en el ejemplo anterior, los comentarios de HTML5 se ponen entre los caracteres **<!--** y **-->**.

1.4.2 Estructura del código HTML5

Generalmente, la estructura de una página web en HTML5 suele tener un patrón parecido.

Estructura de una página web en HTML5

```

1  <!DOCTYPE html>
2  <Html lang = "se">

```

```

3  < Head >
4      < Meta charset " = UTF" - 8 />
5      < Title > Primera Página HTML5 </ title >
6  </ Head >
7  < Body >
8      < H1 > Esto es HTML5 </ h1 >
9      < P > Aquí vemos el contenido de la página web. </ P >
10 </ Body >
11 </ Html >

```

En la figura 1:19 se puede apreciar como se visualiza una página web en el navegador.

F. . Visualización de una página web en un navegador

En el ejemplo anterior se puede apreciar que el documento `html` se diferencia en dos partes: *head* y *body* .

Al *head* está la información adicional de la página web, como puede ser el título que se visualiza en la pestaña del navegador a través del elemento `<title>` , así como también elementos `<meta>` , `<script>` y `<link>` , entre otros.

Al *body* está el contenido que se visualiza de la página web, el cual puede estar formado por multitud de elementos `html` que dan formato al documento.

Aunque en la especificación HTML5 se ha hecho un gran esfuerzo para relajar la sintaxis y que las etiquetas `<html>` , `<head>` y `<body>` no serían obligatorias, es del todo recomendable que estén.

Veamos algunos de los apartados que se pueden definir en una página web:

1. El tipo de documento : la primera línea de un documento HTML corresponde a la declaración del tipo de documento, el **doctype** , y sirve para indicar qué versión de HTML se ha utilizado en el documento.

DTD, Document Type Declaration, es una descripción de estructura y sintaxis de un documento, en este caso HTML. Se utiliza para validar el documento.

```

1  <! - Declaración del doctype en HTML5 - >
2  <! DOCTYPE html >
3
4  <! - Declaración del doctype en HTML4.01 transicional - >
5  <! DOCTYPE HTML PUBLIC " - // W3C // DTD HTML 4.01 Transitional // EN" "http://www.w3.
    org / TR /html4/loose.dtd ">
6
7  <! - Declaración del doctype en XHTML 1.0 estricto - >
8  <! DOCTYPE html PUBLIC " - // W3C // DTD XHTML 1.0 Strict // EN" "http://www.w3.org/ TR /

```


Así pues, en el ejemplo anterior se puede apreciar que uno de los objetivos de HTML5 es la simplicidad y sobre todo la compatibilidad en versiones anteriores, como también versiones futuras; por tanto, incluir el número de versión no tendría sentido.

2. El documento HTML : el elemento `html` indica el principio del contenido de la página HTML.

```

1  <!-- Sintaxis en HTML5 -->
2  <html lang="es">
3
4  <!-- Sintaxis en XHTML 1.0 -->
5  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">

```

3. La codificación de caracteres : se suele indicar que codificación de caracteres se ha utilizado en una página web.

```

1  <!-- Sintaxis en HTML5 -->
2  <meta charset="UTF-8" />
3
4  <!-- Sintaxis en HTML 04:01 transicional -->
5  <meta http-equiv="Content-Type" content="text/html" charset="UTF-8">
6
7  <!-- Sintaxis en XHTML 1.0 estricto -->
8  <meta http-equiv="Content-Type" content="text/html" charset="UTF-8">

```

4. Estilos CSS : para indicar los estilos que se utilizarán en la página html .

```

1  <!-- Sintaxis en HTML5 -->
2  <link href="elsMeusEstils.css" rel="stylesheet" />
3
4  <!-- Sintaxis en XHTML 1.0 estricto -->
5  <link href="elsMeusEstils.css" rel="stylesheet" type="text/css" />

```

5. Los scripts : en una página web se pueden incluir *scripts* .

```

1  <!-- Sintaxis en HTML5 -->
2  <script src="elsMeusScripts.js"></script>
3
4  <!-- Sintaxis en XHTML 1.0 estricto -->
5  <script src="elsMeusScripts.js" type="text/javascript"></script>

```

1.4.3 Elementos de HTML5

Todos los elementos del estándar HTML5 y sus atributos, tanto los obligatorios o requeridos como los que son opcionales, se pueden consultar, por ejemplo, en la web developer.mozilla.org/es/docs/HTML/HTML5/HTML5_lista_elementos.

En la especificación HTML5 también se determina qué elementos se han declarado obsoletos. HTML5 debe ser compatible con las versiones anteriores; así pues, los elementos obsoletos seguirán siendo interpretados correctamente por los navegadores.

Si se desea consultar todos elementos HTML estándares, los no estándares y los que están obsoletos y los que no, se puede acceder, por ejemplo, a [developer.mozilla.org/en-US / docs / Web / HTML / Elemento](http://developer.mozilla.org/en-US/docs/Web/HTML/Element).

encabezados

Los encabezamientos, *headings*, son muy importantes en HTML5. se definen con las etiquetas `<h1>` hasta `<h6>`, siendo `<h1>` la más importante, y `<h6>` definiría el encabezamiento de menos.

Automáticamente, los navegadores añadirían un espacio (margen) antes y después de cada encabezamiento.

En el apartado HTML de CODEP se recomienda sólo poner el contenido que iría al elemento `body` y, por tanto, no es necesario escribir toda la estructura de la página web.

Es mejor utilizar los encabezados para escribir títulos en lugar de utilizar, por ejemplo, las etiquetas `<bold>` o `<big>`, ya que los motores de búsqueda usan los encabezamientos para indexar la estructura y el contenido del sitio web.

Véase el resultado del ejemplo en la figura [1:20](#).

Ejemplo de visualización de diferentes encabezamientos

```
1  <!doctype html >
2
3  <Html lang = "es">
4  <Head >
5      <Meta charset = "UTF - 8" />
6      <Title > Primera Página Web </ title >
7  </ Head >
8  <Body >
9      <H1 > Este es el Encabezado 1 </ h1 >
10     <H2 > Este es el Encabezado 2 </ h2 >
11     <H3 > Este es el Encabezado 3 </ h3 >
12 </ Body >
13 </ Html >
```

F. . Visualización de diferentes encabezamientos en un navegador

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-M09/pen/xZIXV?editors=1000.

párrafos

Los párrafos se definen con el elemento `<p>`. Realmente no podemos estar seguros de cómo se visualizará un párrafo, ya que hay diferentes dimensiones de pantalla y el párrafo se verá utilizando más o menos líneas.

Con HTML no se puede cambiar la salida añadiendo espacios en blanco o saltos de línea, ya que el navegador abolirá estos caracteres extras para mostrar la página. De esta manera, cualquier número de espacios en blanco o de saltos de línea contará como un solo espacio en blanco.

Si se quiere que se tengan en cuenta los espacios en blanco y / o saltos de línea, podemos utilizar el elemento `<pre>`.

Ejemplo de visualización de párrafos

```
1  <!doctype html >
2
3  <Html lang = "es">
4  <Head >
5      <Meta charset = "UTF - 8" />
6      <Title > Primera Página Web </title >
7  </Head >
8  <Body >
9      <H1 > Este es el Encabezado 1 </h1 >
10     <P >
11         Este párrafo,
12         contiene varias líneas
13         en el código fuente,
14         pero el navegador
15         no las muestra.
16     </P >
17
18     <Pre >
19         este           párrafo,
20         contiene varias           líneas
21         en el           código fuente,
22         pero           el           navegador
23         no           las           muestra.
24     </Pre >
25 </Body >
26 </Html >
```

El resultado del ejemplo se puede visualizar en la figura [una y veintiuna](#).

F. . Visualización de párrafos en un navegador

Puede ver este ejemplo en el siguiente enlace: codepen.io/ios-daw-M09/pen/pgZWmk?editors=1000.

Colores en HTML5

Hay varias maneras de especificar los diferentes colores que se utilicen para mostrar el contenido del sitio web:

- con el nombre del color en inglés
- con su código RGB
- con su código hexadecimal

En HTML5, los colores se suelen representar a través de estilos en CSS; así, una manera de aplicar un color a un elemento es a través de su atributo `style`.

Veamos con un poco más de detalle cada una de las opciones anteriores:

1. Con el nombre del color en inglés : es una manera sencilla de especificar un color, aunque este método limita la cantidad de colores representables.

El resultado del siguiente ejemplo lo puede visualizar en la figura [01:22](#).

Puede encontrar una lista de colores en la siguiente dirección: es.wikipedia.org/wiki/Colores_web.

Ejemplo de visualización de colores

```
1
2 <P> Colores por su nombre: </P>
3 <P style="color: red"> Texto de color rojo. </P>
4 <P style="color: orange"> Texto de color naranja. </P>
5 <P style="color: yellow"> Texto de color amarillo. </P>
6 <P style="color: cyan"> Texto de color cian. </P>
7 <P style="color: blue"> Texto de color azul. </P>
```

F. . Visualización de colores referenciados por su nombre en un navegador

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-M09/pen/vLzpPz?editores=1000.

Las siglas RGB significan Red, Green, Blue.

2. Con su código RGB : la paleta de colores RGB proviene de la combinación de tres colores primarios: rojo, verde y azul. Por lo tanto, para especificar el código de un color en RGB lo haremos a través de tres números referentes a la cantidad de rojo, verde y azul, respectivamente. De esta manera podemos especificar cualquier combinación de colores.

Ejemplo de colores definidos con su código RGB

```
1 <P> Colores con su código RGB: </p>
2 <P style="color: rgb(255,0,0)"> Texto de color rojo. </P>
3 <P style="color: rgb(255,165,0)"> Texto de color naranja. </P>
4 <P style="color: rgb(255,255,0)"> Texto de color amarillo. </P>
5 <P style="color: rgb(0,255,255)"> Texto de color cian. </P>
6 <P style="color: rgb(0,0,255)"> Texto de color azul. </P>
```

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-m09/pen/VeGQPq.

3. Con su código RGBA : es una extensión de los valores de color RGB añadiendo un valor *alpha* que especifica la opacidad del objeto.

Un valor de color RGBA especifica de la siguiente manera:

RGBA (rojo, verde, azul, alfa)

Donde el parámetro alfa es un número entre 0.0 (completamente transparente) y 1.0 (Totalmente opaco).

Ejemplo de colores RGBA

```
1 <P> Colores RGB con opacidad: </p>
2 <P style="background-color: RGBA(255, 0, 0, 0.3);"> Rojo </p>
3 <P style="background-color: RGBA(0, 255, 0, 0.3);"> Verde </p>
4 <P style="background-color: RGBA(0, 0, 255, 0.3);"> Azul </p>
```

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-m09/pen/RaVYya.

4. Con su código hexadecimal : el código RGB también se puede especificar en hexadecimal, en lugar de especificar tres números decimales. Este código se expresa

con una tira de caracteres que comienza con el carácter # seguido de seis caracteres hexadecimales, dos para cada color primario.

Ejemplo de colores definidos en hexadecimal

```
1 < P > Colores con su código HEX: </ p >
2 < P style = " color : # FF0000 "> Texto de color rojo. </ P >
3 < P style = " color : # FFA500 "> Texto de color naranja. </ P >
4 < P style = " color : # FFFF00 "> Texto de color amarillo. </ P >
5 < P style = " color : # 00FFFF "> Texto de color cian. </ P >
6 < P style = " color : # 0000ff "> Texto de color azul. </ P >
```

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-m09/pen/qbMxXM.

estilos

Para dar un formato de estilo a un elemento HTML usará el atributo `style` . por aplicar estilos a través de este atributo se hará de la siguiente manera:

`style = propiedad1: valor1; ... propietatN: valorN;`

Donde *propiedad* hace referencia a una propiedad CSS y *valor* hace referencia a un valor CSS.

El resultado del siguiente ejemplo puede verlo en la figura [01:23](#).

Ejemplo de atributo "style"

```
1 < H1 style = " font - family: Verdana; color : blue; "> Titol1 </ h1 >
2
3 < P style = " font - family: arial; color : red; font - size : 200%; ">
4 Párrafo con formato
5 </ P >
```

Formato del texto

Aparte de dar formato al texto a través del atributo `style`, el lenguaje HTML también proporciona otros elementos para hacerlo, como los que se pueden ver en la tabla [1.1](#).

T. . Elementos para dar formato al texto

elemento	Descripción
<code></code>	Define texto en negrita
<code></code>	Define texto enfatizado
<code><i></code>	Define texto en itálico, cursiva
<code><small></code>	Define texto más pequeño
<code></code>	Define texto importante
<code><sub></code>	Define texto que debe aparecer en el subíndice
<code><sup></code>	Define texto que debe aparecer en el superíndice
<code><ins></code>	Define texto insertado, subrayado
<code></code>	Define texto rayado
<code><mark></code>	Define texto resaltado

Formato negrita e itálica

```

1
2 <P> Este texto es normal. </P>
3 <P><b> Este texto está en negrita </b>. </P>
4 <P><strong> Este texto es importante </strong>. </P>
5 <P><i> Este texto está en itálica </i>. </P>
6 <P><em> Este texto está enfatizado </em>. </P>
```

El resultado del ejemplo se puede visualizar en la figura [01:24](#).

F. . Visualización de formatos de texto en negrita e itálica en un navegador

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-M09/pen/LGBOJb?editors=1000.

Fíjese en el ejemplo anterior que la visualización del texto a través de los elementos `` y `` es prácticamente igual, pero si se utiliza el elemento `` se añade significado semántico al texto y se está indicando que este es importante.

Puede apreciar el mismo entre los elementos `<i>` y `<I>`. De este modo, si se utiliza el elemento `<I>` también está añadiendo significado semántico al texto.

otros formatos

```
1
2  < H2 > HTML formato del texto < small > pequeño </ small > </ h2 >
3
4  < H2 > HTML formato del texto < mark > remarcado </ mark > </ h2 >
5
6  < P > Este texto es < del > obsoleto </ del >, </ P >
7  < P > Este texto es < ins > obligatorio </ ins >, </ P >
8
9  < P > Aquí ponemos un superíndice: E = m · c < sup > 2 </ sup > </ p >
10 < P > Aquí ponemos un subíndice: H < sub > 2 </ sub > O </ p >
```

El resultado del ejemplo se puede visualizar en la figura [01:25](#).

F. . Visualización de diferentes formatos de texto en un navegador

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-M09/pen/VeByKQ?editors=1000.

imágenes

En HTML, las imágenes se definen a través del elemento ``.

Ejemplo de imágenes


```

1  <img src = "images / img .jpg" alto = " texto alternativo" style = " width : 120
    px; height : 120px; "/>

```

Donde en el atributo **src** hay especificamos el URL o la ruta de la imagen.

Donde en el atributo **alt** hay especificamos un texto alternativo por si la imagen no puede ser mostrada.

Donde en el atributo **style** hay especificamos el estilo de la imagen; por ejemplo, las dimensiones de la imagen a mostrar.

El tamaño de la imagen también se puede especificar a través de los atributos **width** y **height** . Es recomendable, sin embargo, hacerlo a través del atributo **style** , para prevenir posibles hojas de estilos que puedan cambiar el tamaño de la imagen.

Visualización de una imagen

```

1  < Head >
2  < Meta charset = "utf - 8" />
3  < Title > Primera Página Web </ title >
4  < Style >
5      img {width: 100%;}
6  </ Style >
7  </ Head >
8  < Body >
9
10     < img src = "/" html5.png" alto = "Icono" style = " width : 128px; height
        : 128px; "/>
11     < img src = "/" html5.png" alto = "Icono" width = "128" height = "128" /
        >
12
13 </ Body >

```

El resultado del ejemplo se puede visualizar en la figura [uno y veintiséis](#).

F. . Visualización de una imagen en un navegador

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-M09/pen/EPporb?editors=1100.

Vemos, pues, que un estilo en CSS podría modificar la visualización de la imagen.

El navegador Internet Explorer no soporta los elementos `TM` ni los atributos `sr` `TM` `septiembre` ni `sizes` .

otra más pequeña en función del tamaño de la pantalla o de la resolución del dispositivo, así como en función del tamaño del navegador desde el que accedemos al sitio web.

En la unidad "Elementos multimedia en la web: creación e integración" trabajaréis como incluir imágenes en diseños adaptativos.

Por este motivo, HTML5 incluye el nuevo elemento `<picture>` y nuevos atributos, `srcset` y `sizes`, para el elemento ``.

Sin embargo, a la hora de visualizar una imagen hay que tener claros los conceptos de resolución de pantalla y de la imagen:

La **resolución de pantalla** se expresa en píxeles por pulgada (PPP) y relaciona el número total de píxeles de un monitor con su tamaño en pulgadas (*inches*), en la dirección horizontal y la vertical. En inglés, Pixels Per Inch, PPI.

La **resolución de una imagen** indica la cantidad de detalle que puede observarse en esta. Se expresa con dos números enteros, referentes a la cantidad de columnas de píxeles (número de píxeles de ancho) y la cantidad de filas de píxeles (número de píxeles de alto).

Evitaremos mostrar una **imagen grande**, es decir, con mucha resolución, en un espacio pequeño de la pantalla.

Para una correcta visualización y para reducir la cantidad de información descargada es importante intentar adaptar al máximo los píxeles en que se visualizará la imagen con los píxeles que realmente tiene la imagen.

Enlaces

Los enlaces son los elementos que permiten al usuario navegar de una página web a otra.

El elemento que permite definir un enlace en HTML es `<a>`.

Ver los atributos que se pueden especificar para este elemento en la tabla [mesa 1.2](#).

T. . Atributos de los enlaces

atributo	Descripción
href	Permite definir el destino del enlace.
target	Permite definir dónde abrir el documento enlazado.
id	Permite definir una marca dentro de la página.
style	Permite definir el estilo del link.

Ejemplo de enlaces

```
1
2  < A href = "http://www.google.es" target = "_ blank"> Enlace a Google <
   / A >
3  < Br />
4  < A href = "página. Html " style = " color : DARKBlue; texto - decoration:
   none "> Enlace a una página de nuestro sitio web </ a >
5  < Br />
```

```
6 < A href = "/imatgeGran.jpg"><img src = "/imatgePetita.jpg" alto = "
    img1 " style : " width : 30; height : 30 "> </ a >
```

El resultado del ejemplo se puede visualizar en la figura [1.27](#).

F. . Visualización de enlaces en un navegador

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-m09/pen/PZdBgO.

Véase algunas consideraciones sobre los ejemplos anteriores:

- El atributo `href` es obligatorio.
- Si el destino del enlace es una página externa en su sitio web, es necesario introducir la URL completa.
- Si el destino es una página de nuestro sitio web simplemente coloque la ruta de esta página, y en puede obviar el dominio y el protocolo.
- Si desea que el enlace permita la descarga de un archivo o recurso, en el destino del enlace será la ruta de este archivo descargable. Fijaos también que una imagen puede ser un enlace.
- El atributo `target` puede tomar por valor el nombre del elemento que desea que abra el enlace. El valor predefinido `_blank` significa que se abre en una nueva pestaña.
- Los enlaces, por defecto, son de color azul y aparecen subrayados, pero puede modificar su estilo mediante el atributo `style`.

tablas

Si se quiere crear una tabla en HTML5 se utilizarán los elementos de la tabla [1.3](#).

T. . Elementos para las tablas

elemento	Descripción
<code><table></code>	Permite definir una tabla.
<code><tr></code>	Permite definir una fila de la tabla.

T. (continuación)

elemento	Descripción
<code><TDB></code>	Permite definir una columna de una fila.
<code><THB></code>	Permite definir las cabeceras de la tabla.
<code><TM -ptionb></code>	Permite definir el título de una mesa.

No siempre es necesario utilizar todos los elementos para definir una tabla, pero verá como los elementos `<table>` , `<tr>` y `<td>` son indispensables para visualizarla correctamente.

Ejemplo de mesa

```

1  < Table id = "tabla1" border = "1" style = " width : 100%; ">
2  < Tr >
3      < Td > Concepto </ td >
4      < Td > Precio </ td >
5      < Td > Descripción </ td >
6  </ Tr >
7  < Tr >
8      < Td > Objecte1 </ td >
9      < Td > AC10 </ td >
10     < Td > Descripción del objeto 1 </ td >
11 </ Tr >
12 < Tr >
13     < Td > Objecte2 </ td >
14     < Td > AC20 </ td >
15     < Td > Descripción del objeto 2 </ td >
16 </ Tr >
17 </ Table >

```

El resultado del ejemplo se puede visualizar en la figura [01:28](#).

F. . Vista de una mesa en un navegador

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-M09/pen/XXBZZB?editors=1000.

Aunque en el ejemplo anterior hemos utilizado el atributo `border` para especificar que se vea el borde de la mesa, todos los aspectos del formato de visualización de una tabla se suelen especificar en lenguaje CSS a través, por ejemplo, de su atributo `style` .

Para combinar celdas de una tabla se utilizan los atributos `colspan` y `rowspan` los elementos `<td>` o `<th>` .

Ejemplo de tabla con filas combinadas

```

1  < Table id = "Tabla2" style = " border : 1px solid black ">
2  < Caption > Notas Finales </ caption >
3  < Tr >

```

```

4      < Th > Alumno </ th >
5      < Th > Módulo </ th >
6      < Th > Nota </ th >
7      </ Tr >

```

```

8      < Tr >
9          < Td rowspan = "3" style = "vertical - align : texto - top;"> Nombre Alumne1 <
            / Td >
10         < Td > M1 Sistemas Operativos </ td >
11         < Td > 9 </ td >
12     </ Tr >
13     < Tr >
14         < Td > M2 Bases de Datos </ td >
15         < Td > 8 </ td >
16     </ Tr >
17     < Tr >
18         < Td > M3 Programación </ td >
19         < Td > 7 </ td >
20     </ Tr >
21
22     < Tr >
23         < Td rowspan = "3" style = "vertical - align : texto - top;"> Nombre Alumne2 <
            / Td >
24         < Td > M1 Sistemas Operativos </ td >
25         < Td > 7 </ td >
26     </ Tr >
27     < Tr >
28         < Td > M2 Bases de Datos </ td >
29         < Td > 6 </ td >
30     </ Tr >
31     < Tr >
32         < Td > M3 Programación </ td >
33         < Td > 5 </ td >
34     </ Tr >
35 </ Table >

```

El resultado del ejemplo se puede visualizar en la figura [01:29](#).

F. . Visualización de una tabla con filas combinadas en un navegador

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-M09/pen/eJjpp?editores=1000.

Como veis en el ejemplo anterior, en las cabeceras, en la fila 1 y en la fila 5 se definen tres columnas. En cambio, en las filas 2, 3, 6 y 7 sólo se definen dos columnas,

ya que la primera columna está ya ocupada.

El atributo `colspan` funciona de manera similar, y el resultado del ejemplo en el navegador se puede visualizar en la figura [1:30](#).

Ejemplo de tabla con celdas combinadas

```
1 < Table id = "taula3" border = "1">
```

```
2 < Caption > Notas Finales </ caption >
3 < Tr >
4 < Td rowspan = "2" style = "vertical - align : bottom;"> Alumno </ td >
5 < Td colspan = "3"> Módulo 1 Sistemas </ td >
6 </ Tr >
7 < Tr >
8 < Td > EAC </ td >
9 < Td > PAF </ td >
10 < Td > Final </ td >
11 </ Tr >
12 < Tr >
13 < Tr >
14 < Td > Nombre Alumne1 </ td >
15 < Td > 8 </ td >
16 < Td > 8 </ td >
17 < Td > 8 </ td >
18 </ Tr >
19
20 < Tr >
21 < Td > Nombre Alumne2 </ td >
22 < Td > 7 </ td >
23 < Td > 7 </ td >
24 < Td > 7 </ td >
25 </ Tr >
26
27 < Tr >
28 < Td > Nombre Alumne3 </ td >
29 < Td > 6 </ td >
30 < Td > 6 </ td >
31 < Td > 6 </ td >
32 </ Tr >
33
34 </ Table >
```

F. . Visualización de una tabla con celdas combinadas en un navegador

Puede ver este ejemplo en el siguiente enlace: [codepen.io/ioc-daw-M09/pen/MKBGOy?editors=1000](https://codepen.io/ioc-daw-/pen/MKBGOy?editors=1000).

Como veis en el ejemplo anterior, la celda "alumno" ocupa dos filas y la celda "Modulo1 Sistemas" ocupa tres columnas.

Logotipo o icono

Si desea ver el logotipo del sitio web o el icono en la barra de direcciones o en la pestaña del navegador debe crear el icono *.ico*, renombrar con el nombre *favicon.ico* y ponerla en la raíz de nuestro sitio web.

El motivo de que el archivo se diga *favicon.ico* y se ponga a la raíz del sitio es porque, por defecto, los navegadores como Firefox, Safari u Opera buscan en esta ubicación.

Para asegurarnos de que el navegador encuentra nuestro *favicon* es recomendable insertar la siguiente línea de código en la cabecera de nuestro documento HTML (entre las etiquetas `<head>` y `</head>`), indicando la ruta del *favicon*, que en este caso es la raíz del sitio:

```
1 <Link rel "Shortcut" Icon href "favicon." Ico type "image / x - " icon />
```

Para crear nuestro icono, primero hay que hacer una imagen cuadrada, por ejemplo 32x32 píxeles, en formato JPG, GIF o PNG.

Si se quiere que el icono tenga un fondo transparente, entonces esta imagen inicial ha de ser en formato PNG y con el fondo transparente.

Para convertir esta imagen a *.ico* se puede hacer a través de la web www.favicon.cc.

1.4.4 Formularios con HTML5

Los formularios se utilizan para que el usuario pueda introducir datos para luego ser enviadas y procesadas por el servidor.

El elemento HTML que permite crear formularios es `<form>`, y en su contenido se definen los diferentes campos que forman parte.

Ejemplo de formulario

```
1  <Form id = "ValidaUsuari" action = "mailto: nom@correu.com" method = "  
    post "autocomplete =" on ">  
2  
3  Nombre de usuario: < br />  
4  < Input type = " texto " name = "username" /> < br />  
5  Contraseña: < br />  
6  < Input type = "password" name = "contraseña" />  
7  < Br />  
8  < Br />  
9  < Input type = "reset" value = "Borrar" />  
10 < Input type = "submit" value = "Aceptar" />  
11  
12 </ Form >
```

El resultado del ejemplo se puede visualizar en la figura [1:31](#).

F .

navegador

Visualización de un formulario en un

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-M09/pen/NxBMQG?editors=1000.

Veamos, en el ejemplo, los atributos más importantes que tiene el elemento `<form>` :

- **id** : indica el identificador del formulario.
- **action** : indica la URL o el nombre de la página que se encarga de procesar los datos del formulario.
- **method** : indica el método de envío del formulario, que puede ser `get` o `post`. Si es `get` , la información del formulario se enviará a través de la URL al que se envía el formulario. Si es `post` , se envía encapsulada en el paquete HTTP.
- **autocomplete** : indica si el navegador completará automáticamente los valores de los campos del formulario basándose en valores que el usuario haya entrado con anterioridad. El valor de este atributo puede ser `donde` o `off` .
- **novalidate** : si este atributo está presente se desactiva la validación nativa

que HTML5 ofrece en los formularios.

- **name** : permite definir un nombre al formulario.
- **enctype** : permite definir el tipo de datos que se enviarán.

Los campos de un formulario se pueden definir a través de diferentes elementos HTML5:

- el elemento `<input>`
- el elemento `<textarea>`
- el elemento `<select>`
- el elemento `<button>`
- el elemento `<datalist>`

El elemento `<input>`

Mediante el elemento `<input>` , ya través de su atributo **type** , podemos insertar diferentes tipos de campo.

Veamos los diferentes tipos en la tabla [1.4](#).

T. . Valores del atributo "type" del elemento "input"

valor	Descripción
texto	Permite insertar un texto.
password	Permite insertar una contraseña, escondiendo caracteres.
hidden	Permite insertar un campo oculto para el usuario.
submit	Permite insertar un botón para enviar los datos al servidor.
image	Permite insertar una imagen como botón para enviar las datos.
button	Permite insertar un botón.
radio	Permite insertar un botón radio.
checkbox	Permite insertar un checkbox.
color	Permite seleccionar un color en el sistema de selección de colores del sistema operativo.
date	Permite insertar una fecha. Los navegadores propondrán una interfaz de selección de la fecha.
datetime	Permite insertar una fecha y hora con la diferencia horaria.
datetime-local	Permite insertar una fecha y hora sin la diferencia horaria.

time	Permite insertar solamente la hora.
month	Permite insertar el mes de una fecha.
week	Permite insertar la semana.
email	Permite insertar un correo electrónico. los dispositivos móviles adaptarán el teclado.
number	Permite insertar un número. Los dispositivos móviles adaptarán el teclado.
range	Mostrará una barra de selección con un cursor con el que podremos escoger un valor. No soportado en IE.
search	Permite insertar un campo de introducción de datos específico para las búsquedas.
tel	Permite insertar un número de teléfono. los dispositivos móviles adaptarán el teclado.
url	Permite insertar un URL. Los dispositivos móviles adaptarán el teclado.

Veamos que a través del elemento `input` se pueden crear diferentes tipos de campo. Y, de hecho, hay varios campos, como por ejemplo *email* , *number* , *url* , etc., que aparentemente son iguales, pero su utilización ayudará a la validación correcta de los datos introducidos en el formulario y que en dispositivos móviles su teclado adapte para una mejor utilización.

Aparte del atributo `type` , también podemos definir otros atributos para el elemento `<input>` . Veamos un recode en la tabla [1.5](#).

T. . Atributos del elemento "input"

atributo	Descripción
id	Permite definir un identificador.
name	Permite definir un nombre para identificar el campo.
value	Permite definir qué valor debe tomar el campo del formulario.
readonly	Permite definir que este campo sólo es de lectura.
disabled	Permite desactivar el campo.
selected	Permite definir que este campo esté seleccionado.
checked	Permite definir que un botón de radio o checkbox esté seleccionado.
size	Permite definir el tamaño del campo.
maxlength	Permite definir el tamaño máximo del valor del campo.
autocomplete	Permite activar la función <code>autocomplete</code> .
autofocus	Permite definir que estos campos es el que tiene el foco por defecto.
min y max	Permite definir un valor mínimo y máximo.
placeholder	Permite definir una descripción o ejemplo para el valor del campo.

No todos los atributos son aplicables a todos los tipos de campo; así por ejemplo, no tiene sentido aplicar el atributo **placeholder** para un campo de fecha o aplicar los atributos **min** y **max** en un campo de contraseña.

Veamos un ejemplo para comprobar la utilización de algunos tipos campo con sus atributos, cuyo resultado lo podemos ver en la figura [01:32](#).

Ejemplo de un formulario con diferentes campos de tipo <input>

```

1  < H3 > Datos del usuario </ h3 >
2  < Hr />
3  < Form method = "post" action = "mailto: usuari@correu.com"
    autocomplete>
4      < Input type = " texto " name = "nombre" size = "50" maxlength = "10"
        placeholder = "Escribe el nombre" required autofocus /> < br />
5      < Input type = " texto " name = "apellido" size = "50" maxlength = "20"
        placeholder = "Escribe el apellido" required /> < br />
6      < Input type = "email" name = "email" size = "50" placeholder = "Escribe
        el correo electrónico "/> < br />
7      < Input type = "password" name = "contraseña" size = "50" placeholder
        = "Escribe la contraseña" disabled /> < br />
8      Fecha de nacimiento: < br />
9      < Input type = "date" name = "dataNaix" />
10     < Hr />
11     Género: < br />
12     < Input type = "radio" name = "gender" value = "masculino" checked />
        Home < br />
13     < Input type = "radio" name = "gender" value = "femenino" /> Mujer < br />
14     < Hr />

```

```

15     Aficiones: < br />
16     < Input type = "checkbox" name = "aficio1" value = "deporte" /> Deporte <
        br />
17     < Input type = "checkbox" name = "aficio2" value = "cine" /> Cine <
        br />
18     < Input type = "checkbox" name = "aficio3" value = "leer" /> Lectura
        < Br />
19     < Input type = "checkbox" name = "aficio4" value = "otros" /> Otros <
        br />
20     Selecciona tu color favorito:
21     < Input type = " color " name = "colorPreferit" /> < br />
22     < Hr />
23     foto:
24     < Input type = "file" name = "foto" />
25     < Hr />
26     < Input type = " button " value = "Haz clic!" onclick = "alert ( 'HOLA
        !!! '); "/>
27     < Input type = "reset" value = "Borrar" />
28     < Input type = "submit" value = "Enviar" />
29     < Hr />
30
31 </ Form >

```

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-M09/pen/YwjRzy?editors=1000.

Algunas consideraciones sobre el ejemplo serían:

- Para que los botones de *radio* se discriminen entre sí deben tener el mismo nombre, al contrario que los botones *checkbox*, que cada uno debe tener un nombre diferente.
- Cada campo tiene un nombre, el cual se utilizará para que el servidor que recibe los datos pueda acceder a cada uno de los datos recibidos. También se puede especificar un identificador a través del atributo `id`, y el servidor también podría acceder a través de este. Pero fijémonos que los botones de *radio* necesitan especificar el atributo `name` para su funcionamiento.

- Fijémonos también en la diferencia de los atributos `maxlength` y `size` : el primero hace referencia a la medida del valor que se le dará al campo y el segundo, a la medida en que se visualiza el campo de texto.
- También en algunos campos se han utilizado los atributos `disabled` , `placeholder` y `required` .

El elemento `<textarea>`

Este elemento permite definir un campo de texto de varias líneas.

Ejemplo de campo de texto de varias líneas

```
1 Observaciones: < br />
2 < Textarea id = "observaciones" rows = "10" cols = "30">
3 Aquí el usuario escribe su observación.
4 </ Textarea >
```

El resultado del ejemplo se puede visualizar en la figura [01-33](#).

F. . Visualización del elemento `<textarea>` en un navegador

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-M09/pen/LGJYPw?editores=1000.

El elemento `<select>`

Permite definir una lista de opciones seleccionables. Cada opción se especifica con el elemento `<option>` .

Ejemplo de utilización del elemento `<select>`

```
1 Qué Sistema Operativo utilizas? < Br />
2 < Select id = "SO" name = "SO">
3 < Option value = "windows"> Windows </ option >
4 < Option value = "linux"> Linux </ option >
5 < Option value = "mac"> Mac OS </ option >
6 < Option value = "otros"> Otros </ option >
7 </ Select >
```

El resultado del ejemplo se puede visualizar en la figura [01:34](#).

F. . Visualización de una lista desplegable
<Select> en un navegador

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-M09/pen/gPdOrg?editors=1000.

El elemento <button>

Con este elemento se puede insertar un botón en el formulario, y es muy similar a <input type = button> . La diferencia es que a través de este se puede modificar no tan sólo el texto del botón, sino que por ejemplo se podría especificar una imagen.

Ejemplo de utilización del elemento <button>

```
1 < Button type = " button " onclick = "alert ( 'Hola a todos!')"> < Img src = "
  fletxa.png "/> </ button >
```

Fijémonos que a través del atributo onclick especificamos la acción a realizar en pulsar el botón. Generalmente, suele ser una función en JavaScript.

JavaScript

Es un lenguaje *script* que ejecuta en la parte del cliente, es decir, en el navegador. permite mejoras en la interfaz de usuario de aplicaciones y páginas web dinámicas.

El elemento <datalist>

Con este elemento podemos especificar una lista de opciones para sugerir al usuario algunas opciones predefinidas para un elemento <input> . Los usuarios verán una lista desplegable con las opciones predefinidas a medida que vayan introduciendo el texto, pero pueden acabar escribiendo otro texto.

Ejemplo de utilización del elemento <datalist>

```
1 < Input list = "navegadores">
2 <datalist id = "navegadores">
3   < Option value = "Internet Explorer">
4   < Option value = "Firefox">
5   < Option value = "Chrome">
6   < Option value = "Opera">
7 </ Datalist>
```

El resultado del ejemplo se puede visualizar en la figura [01:35](#).

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-M09/pen/pgOoRZ?editors=1000.

1.4.5 Visualización de elementos con HTML5

Cada elemento HTML tiene un valor de visualización predeterminada. En la mayoría de elementos, este valor es *block* o *inline*.

Un elemento con visualización de ***block*** siempre empieza a visualizarse en una nueva línea, y ocupa toda la anchura disponible. Un elemento con visualización ***inline*** no empieza en una nueva línea y sólo coge el ancho necesaria.

La vista predeterminada los elementos HTML tanto tipo block como de tipo inline se puede modificar mediante propiedades CSS para conseguir un diseño adaptativo o, en inglés, Responsive Web Design.

Ejemplo de elemento "block" y "en línea"

```
1 < Div >
2 < P >
3 Vivamus commodo porttitor ullamcorper. Donec Convallaria, risus ac
  condimentum vulputate,
4 < Span > Contenido inline </ span >
5 enim Turpin tincidunt ligulas, in eleifend nibh este elementum
  augue. Sed Sagitta nisl velita,
6 te luctus ligulas posuere rutrum. Donec a sem nunc. Integer te
  facilisimo ligulas. Fusce in suscipit Orci.
7 Sed vitae ultrices nisi. Curabitur te porttitor nibh.7
8 </ P >
9 </ Div >
```

El resultado lo podéis ver en las figura [01:36](#) y figura [una y treinta y siete](#).

Podemos apreciar el contenido de tipo 'block' remarcado.

F. . Vista de una caja de tipo "en línea"

Podemos apreciar el contenido de tipo "en línea" remarcado.

1.4.6 Estructura semántica de la página web

En contraposición a la manera de estructurar el contenido de las páginas web que utilizaba HTML4 a través de casi únicamente elemento `<div>` , con el nuevo estándar HTML5 llegan nuevos elementos que aportan estructura **semántica** .

Se puede estructurar una página web con tablas o marcos, pero actualmente estas técnicas se consideran obsoletas y han caído en desuso.

Destacamos los siguientes elementos que aportan semántica a la estructura de la página (*Page layout*):

- Elemento `<header>` : permite insertar una zona de visualización para las cabeceras.
- Elemento `<footer>` : permite insertar una zona de visualización para los pies de página.
- Elemento `<nav>` : permite insertar una zona de visualización para la navegación con vínculos hipertexto.
- Elemento `<section>` : permite agrupar elementos que tengan la misma temática.
- Elemento `<artículo>` : permite insertar un contenido autónomo.

- Elemento `<aside>` : permite mostrar un contenido relacionado con el contenido

al que esté vinculado.

- Elemento `<main>` : permite representar el contenido principal de la página.

Cabe mencionar que en HTML5 sigue siendo de mucha utilidad el uso de las cajas `<div>` .

Puede apreciar en las figura [1:38](#) y figura [01:39](#) la estructura del contenido de una página web en HTML4 y HTML5, respectivamente.

F. . Estructura de los contenidos en HTML4

F. . Estructura de los contenidos en HTML5

Veamos un ejemplo de página web utilizando estas etiquetas con sentido semántico.

Estructura del contenido de una página web en HTML5

```
1  <! DOCTYPE html >
2  < Html lang = "es">
3  < Head >< meta charset = "utf - 8" /> < title > Primera Página Web </ title >
4  < Style >
5  # Capa1 {
6      width: 800px;
7      margin: 0 auto;
8  }
9  header {
10     background - color: # 0000ee;
11     color: white;
```

```

12         texto - align: center;
13         padding: 5px;
14     }
15     nav {
16         line - height: 30px;
17         background - color: #eeeeee;
18         height: 515px;
19         width: 100px;
20         float: left;
21         padding: 5px;
22     }
23     section {
24         background - color: #aaaaaa;
25         float: left;
26         padding: 10px;
27         width: 560px;
28     }
29     articulo {
30         background - color: #ffffff;
31         padding: 10px;
32     }
33     aside {
34         line - height: 30px;
35         background - color: #eeeeee;
36         height: 515px;
37         width: 100px;
38         float: right;
39         padding: 5px;
40     }
41     footer {
42         background - color: # 0000ee;
43         color: white;
44         clear: both;
45         texto - align: center;
46         padding: 5px;
47     }
48 </ Style >
49
50 </ Head >
51 < Body >
52 < Div id = "capa1">
53     <Header>
54         < H1 > Estudios Grado Superior Informática </ h1 >
55     </ Header>
56
57     <Nav>
58         < P > ASIX </ p >
59         < P > DAM </ p >
60         < P > DAW </ p >
61     </ Nav>
62
63     <Section>
64         <Articulo>
65             < H2 > M1. Sistemas Informáticos </ h2 >
66             < P > Bloque 1: 118 horas (8 horas semanales) </ p >
67             < P > Bloque 2: 80 horas (5 horas semanales) </ p >
68         </ Articulo>
69
70         <Articulo>
71             < H2 > M2. Bases de datos </ h2 >
72             < P > Bloque 1: 132 horas (9 horas semanales). </ P >
73             < P > Bloque 2: 99 horas (7 horas semanales) incluyendo - hay
              trabajo en equipo y práctica virtual obligatoria.
              </ P >
74         </ Articulo>
75
76         <Articulo>

```

```

77         < H2 > M3. Programación </ h2 >
78         < P > Bloque 1: 85 horas (5 horas semanales). incluye
              trabajo en equipo y prácticas virtuales obligatoria
              rías </ p >
79
80         < P > Bloque 2: 80 horas (5 horas semanales). incluye
              trabajo en equipo y prácticas virtuales obligatoria
              rías </ p >
81     </ Artículo>
82
83     </ Section>
84
85     <Aside>
86         < P > Links de interés </ p >
87     </ Aside>
88
89     <Footer>
90         < P > IOC - Instituto Abierto de Cataluña </ p >
91     </ Footer>
92
93 </ Div >
94 </ Body >
95 </ Html >

```

El resultado del código se puede ver en la figura [01:40.](#)

F. . Visualización de una página web con estructura semántica en un navegador

Puede ver este ejemplo en el siguiente enlace: codepen.io/ioc-daw-M09/pen/pgOoQG?editors=1100.

Para la visualización de cada elemento se utiliza un estilo en lenguaje CSS. así,

vemos que el tamaño, la posición y el color de cada uno de estos elementos está definido a través de un estilo CSS al *head* de esta página, y por tanto, en el *body* sólo encontramos el contenido.

1.4.7 Validación HTML5

Resulta de gran utilidad poder verificar los errores que puede haber en las páginas.

La herramienta en línea accesible desde validator.w3.org permite validar el código HTML de diferentes maneras:

- Indicando la URL completa de la página que deseamos validar.
- Subiendo el documento HTML.
- Escribiendo el código HTML directamente.

Una vez indicado el código HTML en validar, pulsar la opción *More Options* , y allí indique el *Document Type* , en su caso HTML5.

Finalmente, haga clic en el botón *Check* para que se ejecute la validación, tal como se muestra en la figura [una y cuarenta y uno](#).

F. . Validación de HTML

En unos instantes tendrá el listado y los detalles de errores y alertas, si es que el validador en detecta.

1.4.8 Seguridad HTML5

Actualmente existen muchas aplicaciones web disponibles en HTML5, pero este lenguaje tiene potenciales vulnerabilidades de seguridad. Algunas de ellas provienen los elementos del estándar en sí, y otras provienen de las implementaciones de cada navegador.

Algunas de las vulnerabilidades más importantes de HTML5 son:

- Acceso a los datos de otros dominios: HTML5 integra una API llamada `postMessage` que permite a un *script* de un dominio pasar datos a un *script* que esté en otro dominio. Para asegurar que estas peticiones no son maliciosas, `postMessage` incluye propiedades para que el programador pueda verificar el origen de la petición. Pero estas comprobaciones no son obligatorias, y por tanto depende del programador. Puede consultar más información sobre esta vulnerabilidad a: code.google.com/archive/p/html5security/wikis/CrossOriginRequestSecurity.wiki.
- Algunos elementos HTML5 permiten la ejecución de código JavaScript, y esto lo podría aprovechar un atacante para inyectar código JavaScript y así acceder el control de la navegabilidad de la web. Puede consultar más información sobre esta vulnerabilidad a: code.google.com/archive/p/html5security/wikis/RedirectionMethods.wiki.
- Almacenamiento local: HTML5 permite el almacenamiento *offline*, y incluso se puede acceder a una base de datos local con JavaScript. De esta manera, las aplicaciones web pueden funcionar más rápidamente, pero este sistema también presenta vulnerabilidades, así que sólo se debería utilizar el almacenamiento local en datos que no sean sensibles y que sean accesibles para la totalidad de los usuarios. Puede consultar más información sobre esta vulnerabilidad a: [code.google.com/archive/p/html5security/wikis / WebSQLDatabaseSecurity.wiki](http://code.google.com/archive/p/html5security/wikis/WebSQLDatabaseSecurity.wiki) o bien, a [developer.mozilla.org/es / Docs / Web / HTML / Recursos_offline_en_firefox](http://developer.mozilla.org/es/docs/Web/HTML/Recursos_offline_en_firefox).