



## ELABORACIÓN DE HOJAS DE ESTILO (UF1303)

---

Familia profesional: Informática y Comunicaciones

[Manual de contenidos](#)

*Reservados todos los derechos. El contenido de esta obra está protegido por la ley, que establece penas de prisión o multas, además de las correspondientes indemnizaciones por daños y perjuicios, para quienes reprodujeren, plagiaren, distribuyeren o comunicaren públicamente en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución artística fijada en cualquier tipo de soporte o comunicada a través de cualquier medio, sin la preceptiva autorización.*

*Todos los nombres propios de programas, sistemas operativos, equipos, hardware, programas de afiliación, páginas web, etc. que aparecen en esta publicación son marcas registradas de sus respectivas compañías, organizaciones y propietarios y tan solo se muestran a modo informativo.*

Conzepto Comunicación Creativa

## ÍNDICE

### 1. HOJAS DE ESTILO EN LA CONSTRUCCIÓN DE PÁGINAS WEB

- 1.1. Funciones y características.
  - 1.1.1. Descripción de estilos
  - 1.1.2. Utilización de estilos.
  - 1.1.3. Los estilos en el lenguaje de marcas.
  - 1.1.4. Los estilos con herramientas de edición web.
- 1.2. Hojas de estilo y accesibilidad.
  - 1.2.1. Adecuación de las hojas de estilos.
- 1.3. Tipos de estilo: incrustados, enlazados, importados, en línea.
  - 1.3.1. Descripción de los tipos de estilo.
  - 1.3.2. Enlazar una hoja de estilo externa a un documento HTML.
  - 1.3.3. Incrustar un estilo dentro de un documento HTML.
  - 1.3.4. Importar una hoja de estilo desde un documento HTML.
  - 1.3.5. Importar una hoja de estilo, a través de un archivo con estilos.
  - 1.3.6. Utilización y optimización de los tipos de estilos.
- 1.4. Selectores y reglas de estilo.
  - 1.4.1. Estructura de los estilos.
  - 1.4.2. Sintaxis básica de estilos.
  - 1.4.3. Utilización de elementos y seudoelementos.
  - 1.4.4. Utilización de clases y seudoclases.
- 1.5. Atributos de estilo para fuentes, color y fondo, texto y bloques (párrafos).
  - 1.5.1. Descripción de los atributos de estilo.
  - 1.5.2. Utilización de los atributos de estilo.
- 1.6. Creación de ficheros de estilo.
  - 1.6.1. Definición de los ficheros de estilo.
  - 1.6.2. Creación de ficheros de estilo genéricos.
  - 1.6.3. Adaptación de los ficheros de estilo para distintas páginas web.

### 2. DISEÑO, UBICACIÓN Y OPTIMIZACIÓN DE LOS CONTENIDOS DE UNA PÁGINA WEB.

- 2.1. Creación de un documento funcional.
  - 2.1.1. Descripción de los objetivos de la página web.
  - 2.1.2. Definición de los elementos funcionales de la página web.
  - 2.1.3. Descripción de cada elemento.
- 2.2. Diseño de los contenidos.
  - 2.2.1. Diseño de página de inicio.
  - 2.2.2. Página de fotografías.
  - 2.2.3. Página CV
  - 2.2.4. Página contacto
- 2.3. Identificación de la información a ubicar en la página web.
- 2.4. Selección de contenidos para cada elemento de la página.
- 2.5. Utilización del documento funcional para las especificaciones del diseño.

- 2.6. Tipos de página para la ubicación de contenidos.
- 2.7. Definición de los tipos de página en base a los contenidos y funcionalidades.
- 2.8. Selección de los tipos de página para la página web.
- 2.9 Utilización del documento funcional para las especificaciones del tipo de página.
- 2.10. Especificaciones de navegación.
- 2.11. Creación de un mapa de navegación de páginas.
- 2.12. Utilización del documento funcional para integrar el mapa de navegación.
- 2.13. Elementos utilizados para la navegación.
  - 2.13.1. Definición de los elementos utilizados para navegar.
  - 2.13.2. Utilización del documento funcional para especificar los elementos de navegación.
- 2.14. Elaboración de una guía de usuario.

## OBJETIVOS

### OBJETIVO GENERAL DEL MÓDULO

Crear y publicar páginas web que integren textos, imágenes y otros elementos, utilizando lenguajes de marcas y editores apropiados, según especificaciones y condiciones de "usabilidad" dadas y realizar los procedimientos de instalación y verificación de las mismas en el servidor correspondiente.

### OBJETIVOS ESPECÍFICOS

C1: Describir las características de las hojas de estilo para dar formato a las páginas web, y crear ficheros de estilo de acuerdo a un diseño especificado.

CE1.1 Identificar las características y ventajas que proporcionan las hojas de estilo para definir la forma de presentación de las páginas web según el diseño especificado.

CE1.2 Describir las etiquetas y los atributos que se utilizan para definir estilos para una página y para partes de la misma, teniendo en cuenta unas especificaciones de diseño.

CE1.3 Describir la sintaxis y los atributos que se utilizan para crear estilos, con el fin de crear ficheros con declaraciones de estilos.

CE1.4 Enlazar las páginas de un sitio web con una fichero de estilo, para homogeneizar el aspecto del sitio web según unas especificaciones recibidas.

CE1.5 Realizar un fichero con declaraciones de estilo para homogeneizar el formato de un sitio web, según un diseño especificado:

- Identificar la sintaxis a utilizar para asignar el estilo a las etiquetas.
- Enlazar las páginas web al fichero de estilos.
- Probar las páginas utilizando varios navegadores web.
- Aplicar criterios de «usabilidad» y accesibilidad.
- Documentar el fichero creado.

C2: Diseñar, ubicar y optimizar los contenidos de una página web, para adecuarla al formato de la misma, facilitar su manejo a los usuarios y optimizarla de acuerdo a un diseño especificado.

CE2.1 Seleccionar los contenidos de la página web para adecuarlos al formato de la misma de acuerdo con las especificaciones recibidas.

CE2.2 Ubicar y distribuir los elementos para facilitar el manejo de la página por parte de los usuarios.  
CE2.3 Optimizar las páginas web teniendo en cuenta sus elementos, contenidos y sus funcionalidades.

## CONTENIDOS

### 1. HOJAS DE ESTILO EN LA CONSTRUCCIÓN DE PÁGINAS WEB

Una hoja de estilo es un archivo que suele acompañar al archivo principal donde se halla la estructura de la propia página web. Los archivos CSS tienen la extensión .css y se encargan de definir los diferentes aspectos relacionados con el formato de diseño de una web. Para ello, dentro de estos archivos se establecerán diversos parámetros donde se definirán las etiquetas utilizadas en el código fuente de la web.

En los archivos CSS es donde se almacena la información que utilizará el navegador para interpretar los estilos e la web, es decir, son las indicaciones de los colores de los elementos, el tamaño, la posición, el espaciado entre párrafos o entre letras, los estilos de los diferentes títulos, etc. Gracias a estas características los desarrolladores web puede darle una apariencia profesional y limpia a las páginas web. La apariencia de una web es muy importante, ya que, de ella dependerá el número de usuarios que se sientan atraídos por la misma. A nadie le gustaría entrar en una página web caótica y difícil de ver.

Además, este es uno de los ficheros que se utilizarán para crear la identidad de una marca y plasmarla en el entorno web. Para ello, se establecerán diversos parámetros estéticos, dependiendo de qué es lo que quiere ofrecer la marca al público.

Un documento CSS la gran ventaja que tiene es que se puede modificar de manera sencilla y rápida y todos los estilos se aplicarán de manera inmediata a todos los elementos HTML.

La principal utilidad de una hoja CSS es dar coherencia al diseño de una web y mantener ese estilo de manera uniforme y homogéneo a través de todas las páginas que compongan esa web.

Gracias a todo esto que ofrece CSS, las empresas pueden establecer un patrón determinado con el que se identificará su estilo propio para construir la web de la misma y, de este modo, conseguir una imagen corporativa profesional y de confianza que se fácil de vincular con la propia empresa. De esta forma, la imagen en Internet de la empresa se verá reforzada de manera gráfica y los usuarios la reconocerán inmediatamente.

#### 1.1. Funciones y características

Puede hablar de hojas de estilo u hojas CSS o Cascading Style Sheets. Las hojas de estilo permiten manipular y controlar las propiedades visuales de los elementos que hay en una página web. Actualmente, utiliza el HTML para crear la estructura de la página web y el CSS para crear el estilo o apariencia de la misma.

El código CSS es el encargado de modificar los elementos que están codificados en HTML mediante la modificación de las propiedades de los propios elementos como, por ejemplo, el color, el tamaño de borde, el posicionamiento y tamaño del propio elemento, etc. Gracias a esto, los desarrolladores

tendrán un control total de cada uno de los elementos de la web. Por lo que podrán crear la interfaz al milímetro.

Además de las diversas propiedades visuales que puede aportar a la página el CSS, también debe agradecer la gran contribución que brinda el CSS al propio desarrollo web, ya que, CSS permite que separe el código de estilo del contenido. Esto quiere decir que, si realiza un cambio en el estilo de su página web, se va a aplicar en todos los elementos que hayan sido asignados con ese estilo, por lo que, esta funcionalidad del CSS ahorrará muchísimo tiempo al no tener que estar cambiando el estilo de cada uno de los elementos uno por uno. Otra de las cosas que permite hacer CSS es crear colecciones de estilos, es decir, unir varias hojas CSS que puede utilizar para darle estilo a un documento HTML, de este modo, puede utilizar los mismos estilos para varias páginas a la vez.



Antes de que existiera el código CSS, los desarrolladores tenían que codificar una y otra vez las mismas estructuras de código como, por ejemplo, el color del texto y el tamaño del mismo. Además, el código HTML como tal no ofrecía una gran cantidad de posibilidades para cambiar la apariencia de una web, tan solo se podía cambiar el color y la tipografía de una fuente y poco más.

Por ello, el uso de las hojas CSS es tan importante, porque brinda una infinidad de posibilidades que el HTML jamás podrá ofrecer. Cuando utiliza CSS, lo que hace es codificar un mayor número de características para un mismo o varios elementos, por lo que, esas características quedan establecidas e inamovibles. Por eso, solo debe codificarlas una vez sin tener que estar repitiendo constantemente las mismas estructuras. A continuación, va a ver un ejemplo de CSS donde puede ver el estilo que se le da a la etiqueta <div>:

```
Div { Font-family: Verdana;
```

```
Font-size: 12px;
```

```
Color: #000
```

```
}
```

Como ha dicho anteriormente, una vez establecido el estilo para un elemento no tiene que volver a repetir la misma estructura de código. Por lo tanto, este código CSS establece el estilo para los elementos <div>, eso quiere decir que todos los elementos <div> de la página web a la que le aplique este código CSS, tendrán el mismo estilo. Debe recordar que todas las características que ofrece CSS no serían posibles utilizarlas con HTML.

### 1.1.1. Descripción de estilos

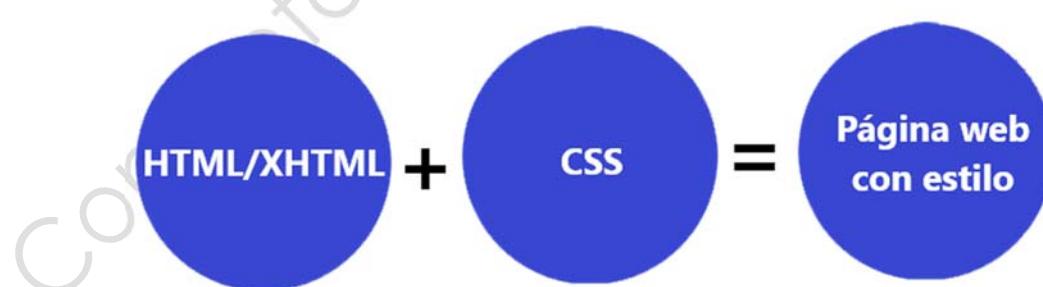
CSS se desarrolló como estándar para complementar al HTML, esto ofreció un control más estricto sobre la visualización de los elementos en las páginas web. Al principio, el CSS tenía un fin muy claro, mejorar la apariencia del HTML y dar estilo a los elementos, puesto que el HTML cuando se desarrolló solo le dio importancia a la estructura de una página web, pero se olvidaron de una de las características más importantes de una web, la apariencia visual. Es por ello, que gracias al CSS puede hacer una página mucho más atractiva al ojo de los usuarios, sin necesidad de modificar el contenido del propio documento HTML.

Puede obtener muchas ventajas si separa los contenidos de una web, de su aspecto, puesto que, obtendrá unos documentos llamados HTML/XHTML, los cuales están muy bien definidos y estructurados. A estos tipos de documentos también se les denomina documentos semánticos.

Además, la accesibilidad a este tipo de documentos es mejor, al igual que el mantenimiento de la web es mucho más sencillo de realizar. Una de las ventajas más importantes es que puede visualizar el documento en la mayoría de los dispositivos diferentes que utilizan los usuarios de Internet.

Al desarrollar una página web, lo primero que hará es utilizar un lenguaje de marcas HTML/XHTML para crear las estructuras de los contenidos y sus elementos, es decir, lo que hará será asignar a cada elemento de la web una función dentro de la misma. Por ejemplo, un título, un párrafo, una tabla, una lista, un texto destacado, entre otros.

Cuando ya ha terminado de crear este tipo de contenidos HTML/XHTML, es cuando utiliza el lenguaje CSS para poder darle estilo a cada uno de los elementos que ha creado anteriormente. Los estilos que les dará a estos elementos pueden variar dependiendo de lo que está buscando. Por ejemplo, con CSS puede cambiar el color, tamaño y tipo de letra en los textos, puede establecer una separación vertical u horizontal en cada uno de los elementos, puede posicionar cada uno de estos elementos dentro de la página en el lugar que quiera, etc.



### 1.1.2. Utilización de estilos

Anteriormente a que los desarrolladores web utilizaran código CSS, en su lugar, utilizaban etiquetas especiales de HTML para poder modificar la apariencia de los elementos que conformaban una página web. A continuación, va a ver un ejemplo donde puede ver una página HTML utilizando estas

etiquetas especiales de HTML para definir los estilos de los elementos de la página. Por ello, no se ha usado código CSS:

```
<!DOCTYPE html><html>

<head>

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

<title>Uso de estilos sin CSS</title>

</head>

<body>

<h1><font color="blue" face="Verdana" size="5">Título de la página</font></h1>

<p><font color="green" face="Arial" size="2">Esto es un párrafo de la página</font></p>

</body>
```

</html>Como puede ver en este ejemplo, ha utilizado la etiqueta `<Font>` que ofrece HTML para establecer las características de color, de fuente y de tamaño de los textos mediante sus atributos, color, face y size. Otra cosa que también puede observar es que para cada uno de los elementos que ha establecido en el ejemplo, ha tenido que repetir la misma estructura con las características de apariencia.

El gran problema de establecer los estilos con HTML de cada uno de los elementos es que si, por ejemplo, tiene 100 elementos, tendría que insertar 100 etiquetas `<Font>` para cada uno de ellos. Si, por ejemplo, el sitio web tuviese 150 páginas diferentes con 100 elementos en cada una de ellas, tendría que establecer 15000 etiquetas `<Font>` una por una.

Por otra parte, como los sitios web están en constante desarrollo, se deben modificar los elementos cada cierto tiempo para tenerlos actualizados. Por lo tanto, si toma el ejemplo que ha citado anteriormente, tendría que modificar unas 15000 etiquetas con sus respectivos 45000 atributos. Por lo que, sería un trabajo arduo y caótico.

Es por ello que se utiliza el código CSS, puesto que, si tiene que modificar miles de etiquetas y atributos, no supondría un gran esfuerzo, ni tomaría mucho tiempo para hacerlo. A continuación, vea el siguiente ejemplo de documento HTML que utiliza código CSS para establecer los estilos de los elementos:

```
<!DOCTYPE html>

<html>
```

```
<head>

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

<title>Uso de estilos con CSS</title>

<style type="text/css">

    h1{color:red; font-family: Verdana, Geneva, Tahoma, sans-serif; font-size: large;}

    p{color:green; font-family: Verdana, Geneva, Tahoma, sans-serif; font-size: medium;}

</style>

</head>

<body>

    <h1>Título de la página</h1>

    <p>Un párrafo de la página</p>

</body>

</html>
```

En este ejemplo, puede ver cómo se separa en el mismo archivo el código HTML con los contenidos de la página, del código CSS que es el que establecerá los estilos. Es por ello que, en este documento HTML, está dividido en una zona donde se aplican todo lo relacionado con los estilos. Para dividir esa zona, se aplican las etiquetas `<style>` y `</style>` que es donde se establecerá todo el código CSS del documento.

En este ejemplo, puede observar cómo en la zona donde se establece el código CSS se ha indicado que el contenido de todas las etiquetas `<h1>` del documento tengan el texto color rojo y la letra Verdana en un tamaño grande. En cambio, todas las etiquetas tipo párrafo `<p>` de este documento, tendrán el texto color verde, la tipografía Arial y la letra tendrá un tamaño medio.

Como ha podido observar, al utilizar CSS puede establecer los mismos estilos que ha establecido anteriormente con HTML, pero de manera más sencillas, ya que, no tendrá que repetir la misma estructura de código por cada elemento y tampoco ensuciará el código HTML con la etiqueta `<Font>`. Como ya verá más delante de este tema, en la etiqueta `<style>` podrá crear una zona especial en donde incluirá las reglas CSS que se establecerán en toda la página web.

Utilizar código CSS aporta muchísimas ventajas, ya que, puede crear los mismos estilos que con HTML y más, sin tener que aplicar ningún esfuerzo mayor a la creación del código.

Además, no se ensuciará el código HTML con múltiples etiquetas <Font> y, por ello, podrá visualizar el código mucho más claramente, ya que, el código CSS estará ubicado en la sección correspondiente dentro de sus etiquetas <style> y </style>.

Al comparar ambos ejemplos con los que se obtiene el mismo resultado, puede darse cuenta de que, al hacer uso de las hojas de estilo, puede ahorrar tiempo sin tener que estar modificando miles de etiquetas y de atributos. Sin embargo, aunque es una solución ventajosa, no es del todo cómoda.

Cuando vaya avanzando en la materia, verá que los estilos CSS solo se establecen en el documento en el que están incluidos, es por ello que verá algunas soluciones para este problema donde podrá utilizar el mismo documento CSS para diversos archivos al mismo tiempo, por lo que, no va a tener que copiar una y otra vez las mismas reglas CSS para todos los documentos en los que quiera utilizarlas.

### **1.1.3. Los estilos en el lenguaje de marcas**

La funcionalidad de los estilos es dar formato al contenido de una página. Los estilos están contenidos en archivos, al igual que el código HTML, la diferencia es que la extensión de estos archivos es .css en vez de .html. Es por ello, que puede desarrollar los estilos en un editor de código al igual que haría con los documentos HTML.

El lenguaje CSS permite establecer definiciones de estilos a los bloques con los cuales se compone la página, es decir, las tablas, textos, títulos, imágenes, párrafos, etc. Cada uno de estos elementos se pueden modificar con código CSS.

Por poner un ejemplo, puede decir que el HTML es el esqueleto de una página web y el código CSS sería su apariencia, que puede ser más o menos atractiva, dependiendo de sus gustos o necesidades.

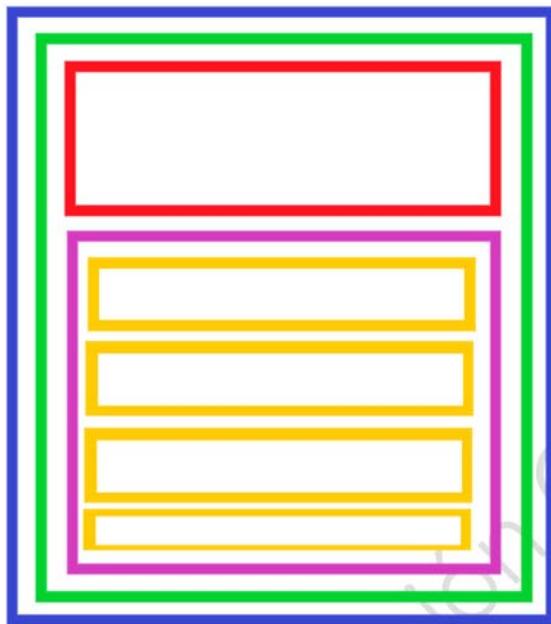
CSS son sigla en inglés que significan Hojas de estilo en Cascada. La palabra cascada significa que los estilos están aplicados en cascada, esto quiere decir que se aplican uno detrás de otro.

Otra de las cosas que debe saber es que el contenido de un documento HTML está distribuido en bloques. Por ejemplo, la página en sí es un bloque que se distribuye dentro de las etiquetas <html> y </html>, a su vez, la cabecera es otro bloque que está contenido en las etiquetas <head> y </head>, el body es otro bloque que está contenido en las etiquetas <body> y </body>, y así sucesivamente hasta llegar a las estructuras más simples como, por ejemplo, los párrafos, tablas, títulos, etc.

Los bloques en HTML funcionan como si fueran cajas contenedoras donde el código CSS establece los efectos y el estilo de cada uno de esos contenidos.

En la teoría, cada caja contiene otras cajas de manera en que unas van encajando con otras. En informática, se hace referencia a este tipo de relaciones entre contenidos como elementos padres y elementos hijos. Un bloque padre es aquel que contiene a otros bloques hijos, mientras que un bloque hijo es aquel que está contenido dentro de un bloque padre. Casi todos los bloques pueden ser padre de otros bloques y simultáneamente hijos de otros bloques. A continuación, va a ver una

Imagen donde podrá entender mejor estos conceptos observando de manera gráfica las diversas partes de un documento HTML dividido por bloques que mantienen relaciones unos con otros:



Como puede observar, el bloque verde es hijo del bloque azul, y a su vez, el bloque verde es padre de los bloques rojo y morado e, indirectamente, de los bloques amarillos que están contenidos en el bloque morado.

El bloque azul es el body, el cual no es hijo de ningún otro bloque, sin embargo, es el padre de todos los demás bloques. Si utiliza CSS para aplicar un estilo al body, tales estilos se aplicarán a todos los demás bloques, desde los más grandes hasta los más pequeños. Por eso se le llama hojas de estilo en cascada. Por otra parte, en informática, conoce este hecho como herencia, por ejemplo, en el caso de los estilos, se aplicarán siempre de padres a hijos, pero nunca de hijos a padres.

Por último, se debe preguntar por si un estilo siempre se aplica de padres a hijos, cómo podría hacer que un hijo tuviera su propio estilo. Esta cuestión la verá más adelante en profundidad. Sin embargo, puede mencionar que la cascada de los estilos tiene una regla de precedencia, es decir, el navegador siempre ignorará un estilo heredado de un parent si, en su caso, existiera un estilo aplicado directamente al hijo. Es por ello que, es posible definir un estilo para todo el documento y después ir definiendo pequeños estilos a cada uno de los bloques que quiera que tengan una apariencia diferente al resto.

En este vídeo, de seis minutos de duración, se demuestra cómo agregar estilos CSS de múltiples maneras y utilizar la que más convenga en cada momento:

## RECURSO MULTIMEDIA



### 1.1.4. Los estilos con herramientas de estilos

El estilo de una página web, puede diseñarlo de diversas maneras. Puede diseñarlo en un archivo de texto normal y corriente, en cambio, también puede diseñar y aplicar estos estilos mediante otras herramientas de desarrollo web. Las herramientas que tienen como finalidad el desarrollo web, ofrecen muchísimas más ventajas y facilitan el trabajo de desarrollar cualquier tipo de código, en el caso de CSS puede encontrar herramientas que ayudan a escribir propiamente el código, y otras que ayudan a utilizarlo.

Vea a continuación algunos de los nombres de estas herramientas, sus características y cómo puede comenzar a utilizarlas:

#### Firebug

Esta herramienta es una de las imprescindibles para cualquier desarrollador web. Consiste en una extensión, gratuita, del navegador Firefox, aunque también puede encontrarla para otros navegadores.

Esta herramienta puede descargarla desde su [web](#) oficial. Tan solo debe pulsar en el botón Instalar Firebug que encuentre en la esquina superior derecha. Este botón descargará e instalará este *plug-in*. Cuando haya finalizado, debe reiniciar el explorador Firefox y ya lo tendrá funcionando.

En caso de que quiera instalar Firebug en otro navegador que no sea Firefox como, por ejemplo, en Opera o Safari, debe dirigirse al siguiente [enlace](#). Una vez este en el sitio web, tan solo debe seguir las instrucciones de instalación para poder habilitar la herramienta en su navegador, para poder utilizarla en una página web que este probando. También tiene la opción de añadirlo como marcador o bookmarklet a la barra de herramientas del navegador. Sin embargo, esta versión de Firebug no es tan completa como la original, aunque sigue siendo muy potente y de mucha utilidad.

Comentar todas las posibilidades que ofrece la herramienta Firebug es casi imposible, puesto que son muchísimas. Por ello, va a centrarse en las más importantes destacando su funcionalidad, aspecto y utilidad más relevantes.

Cuando abra un documento web en el navegador Firefox teniendo el *plug-in* Firebug instalado, puede observar que la página web se verá de manera diferente a lo habitual. Lo primero que va a mostrar es la página web, y justo debajo en el lado izquierdo de la propia página, podrá observar que

se muestra una interfaz donde está la estructura HTML del documento que está visualizando. Dentro de esta estructura HTML, podrá utilizar las diferentes flechas para expandir o contraer el subárbol del documento HTML.

Además, podrá visualizar que, al pasar el cursor del ratón por encima del nombre de un elemento HTML, se mostrará resaltado en la página donde se mostrará el área del contenido, los márgenes y el relleno del propio elemento que se establecerá con márgenes de colores diferentes.

Por otra parte, en el lado derecho de la propia ficha HTML, podrá visualizar el código CSS a dicho elemento que ha seleccionado, al pulsar la ficha de estilo. Vea qué cosas debe tener en cuenta cuando navega por la parte, ya mencionada, ficha de estilo:

Al navegar por la ficha de estilo, podrá ver propiedades que no ha especificado con anterioridad, por ejemplo, la propiedad -moz-background-clip. Inicialmente, puede ignorar este tipo de propiedades, a no ser que las haya declarado de manera explícita.

Las propiedades que haya utilizado de manera abreviada, aparecerán de manera individual.

Las reglas se establecerán por orden inverso de especificidad, por ejemplo, la primera regla será la más específica que se aplicará al elemento seleccionado anteriormente, la segunda regla será la siguiente más específica y, así, seguidamente.

Para inspeccionar cualquier elemento de la página, tan solo tendrá que pulsar el botón derecho del ratón sobre ese elemento y elegir la opción, que aparecerá en el menú contextual, Inspeccionar elemento.

Por otro lado, podrá cambiar al modo Inspección si pulsa en el pequeño icono representado por un puntero que señala a una caja, el cual se encuentra ubicado al lado del icono de Firebug. Este modo hará posible que los elementos queden resaltados a medida que vaya seleccionándolos y, de ese modo, pueda inspeccionarlos uno por uno.

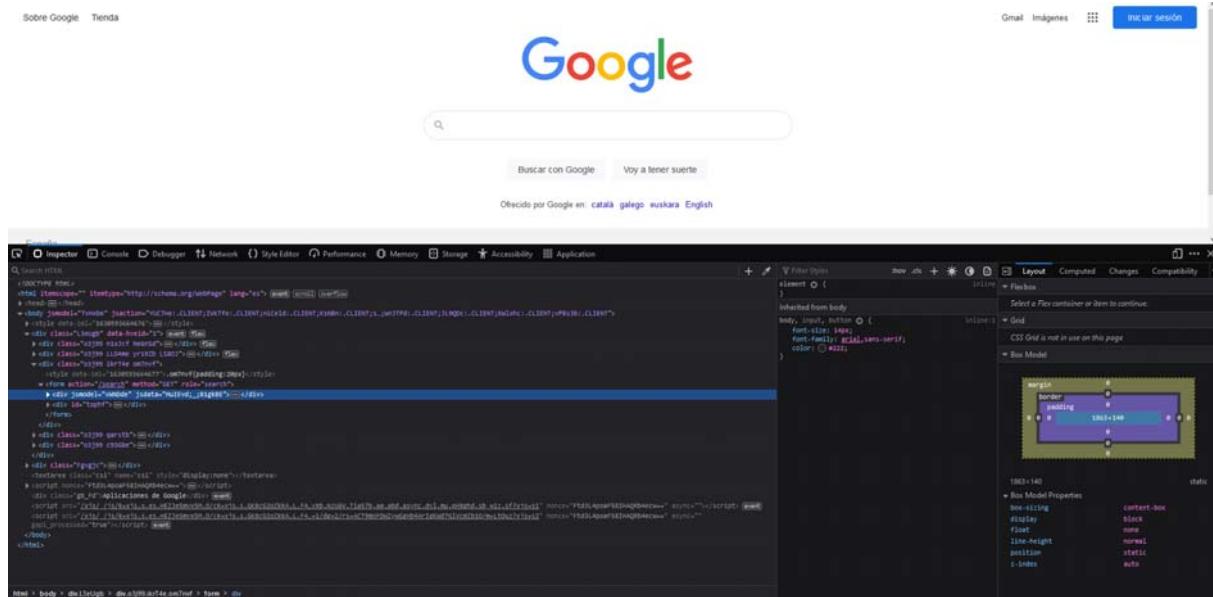
En caso de que desee desactivar cualquier declaración, tan solo debe pulsar en el lado izquierdo. Esto será muy útil cuando esté comprobando cómo funciona cualquiera de las propiedades de un elemento, de este modo podrá observar los efectos que se producen al activar o desactivar estas propiedades.

Además, también puede observar cómo al pulsar sobre un valor en concreto del panel izquierdo, aparecerá una caja en la ficha de estilo que mostrará el color que está definido en la declaración de color pertinente.

Por otro lado, si lo que quiera es saber los tamaños y las dimensiones exactas de los componentes del modelo de cajas de un elemento de la web, como, por ejemplo, el ancho, el alto, los márgenes o el relleno del mismo, tan solo debe pulsar sobre la ficha de Maquetación. En esta ficha podrá ver los colores en píxeles de los elementos y si pasa el puntero del ratón por encima del cuadro que aparece

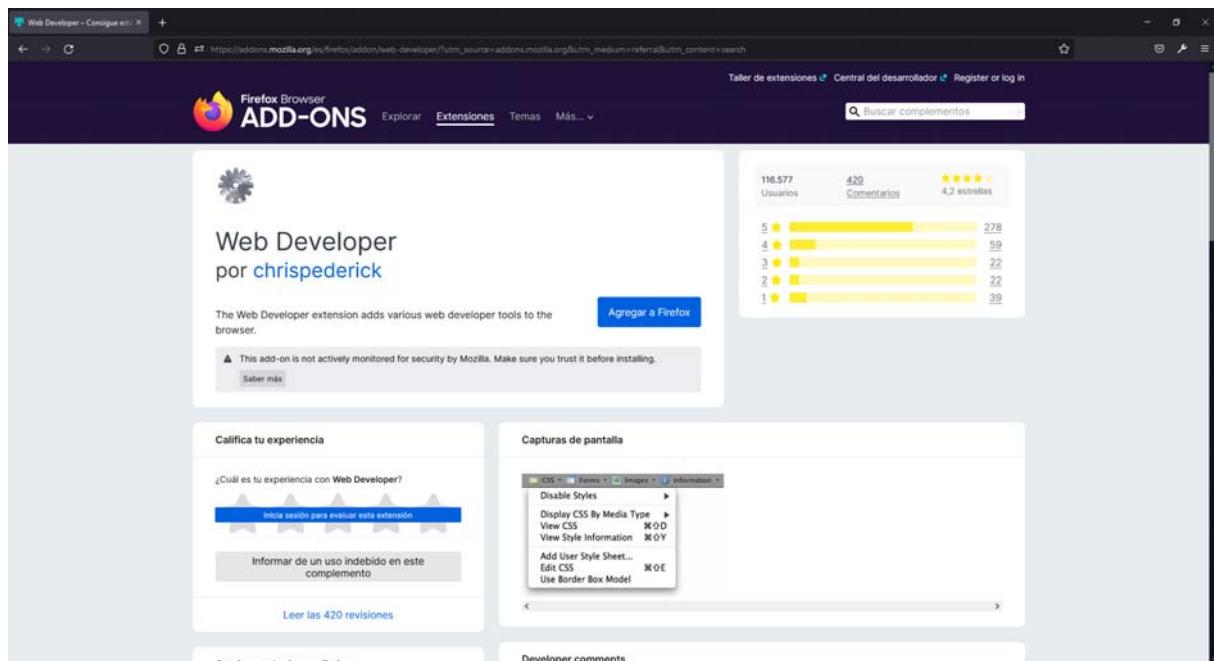
en dicho panel, se mostrarán las reglas en píxeles de la propia página web en sí, tanto en el lado superior, como como en el izquierdo del borde exterior del propio elemento.

Por último, debe saber que también puede modificar los valores e los atributos de los elementos, por ejemplo, elementos como class, también puede cambiar el contenido propio del elemento. Para ello, solo tendrá que añadir o editar los valores de las propiedades CSS.



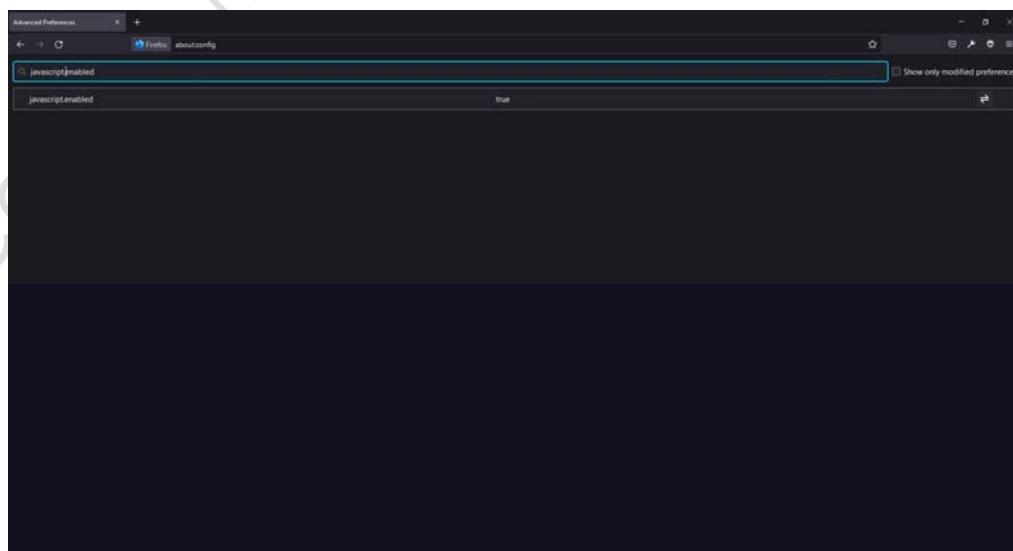
## Web Developer Toolbar

La barra de herramientas para desarrollo web o Web Developer Toolbar, se considera otros de los elementos fundamentales para cualquier desarrollo web. Esta herramienta, al igual que Firebug, también es totalmente gratuita y ha sido desarrollada por Firefox. Para poder obtener esta herramienta, puede dirigirse a la propia [web](#) de addons de Firefox. Una vez entre en esta web, tendrá que buscar en la barra buscadora Web Developer e instalarla desde la propia página siguiendo los pasos correspondientes. Vea el ejemplo en la siguiente imagen:



Esta herramienta ofrece una inmensa cantidad de opciones, al igual que lo hacía la herramienta Firebug, es por ello que no podrá nombrarlas todas. En cambio, se habla de las opciones y menús más importantes de esta herramienta. Por todo esto, es importante que cuando tenga la herramienta instalada en su navegador, investigue cada uno de los menús para aprender a utilizarla como un profesional y sacarle todo el rendimiento que realmente ofrece aprovechando todas las posibilidades que brinda.

Para hacer actualizaciones menores en una página web, se recomienda antes deshabilitar el almacenamiento temporal de la página, ya que, la caché del navegador va a soportar muchos archivos. En caso de que falle un *script* o no se abra el entorno de JavaScript, también, también puede deshabilitar JavaScript del navegador y así resultará mucho más sencillo dar con el error. Vea el siguiente ejemplo:

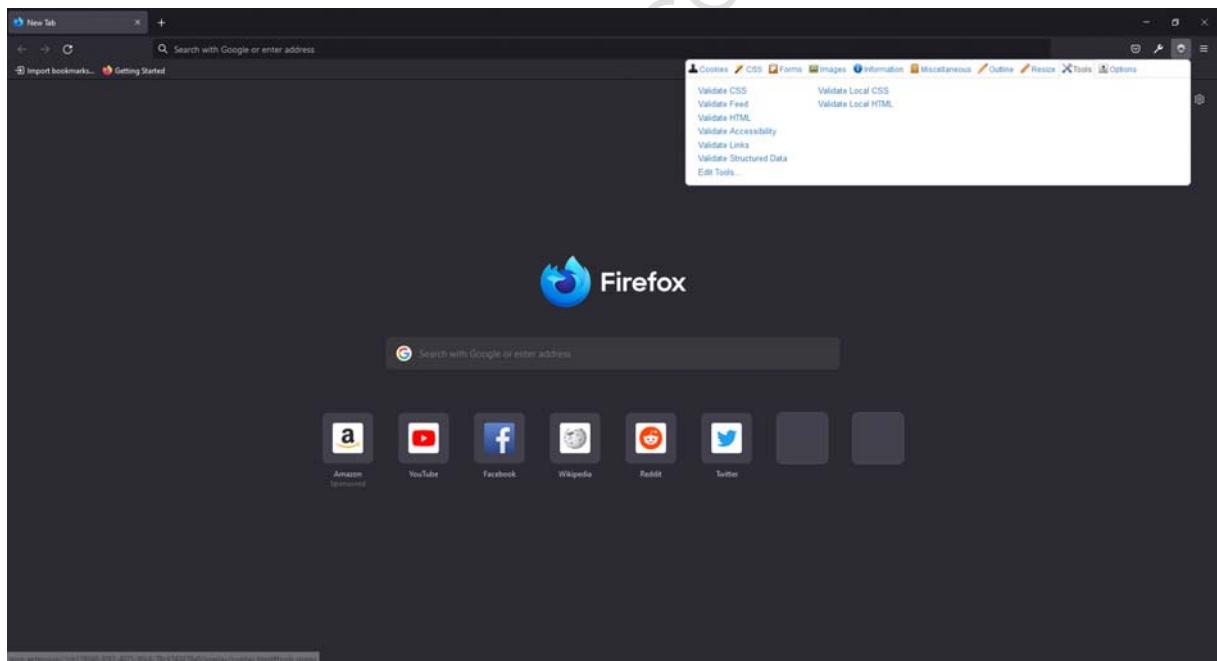


Por otra parte, desde el menú CSS de la herramienta Web Developer Toolbar, solo podrá desactivar los estilos incrustados y las hojas de estilo en línea o estilos enlazados. Además, puede desactivar los estilos por defecto que trae el propio navegador, aunque no se recomienda hacerlo.

Además, el menú Información ofrece la posibilidad de mostrar información sobre la clase e id del propio documento, también del orden de las capas o div de la página y un resumen de los colores que se utilizan en el documento. Esta herramienta también ayuda cuando quiera saber alguna información sobre los atributos y los valores de un elemento en concreto, además de su posición en la página, la información de la fuente que utiliza, y el elemento que hay anterior y posterior a él, entre otras opciones. Para ello, tan solo debe hacer clic en el elemento sobre el que quiera sacar esa información.

Por otro lado, puede resaltar las clases de los elementos como elementos de bloque, posicionados, tablas, etc. Además de poder indicar los elementos y colores propios de los mismos, en el menú Resaltar, ya que, al ser esta una herramienta de diseño y diagnóstico, podrá resaltar los elementos para trabajar con rapidez y de manera eficiente con la disposición que los elementos tengan unos con otros y, de esta manera, será mucho más sencillo encontrar los problemas.

Por último, en el menú de Herramientas encuentre un acceso rápido a un montón de herramientas validadores, comprobadoras de errores y consolas de depuración. Vea el ejemplo:



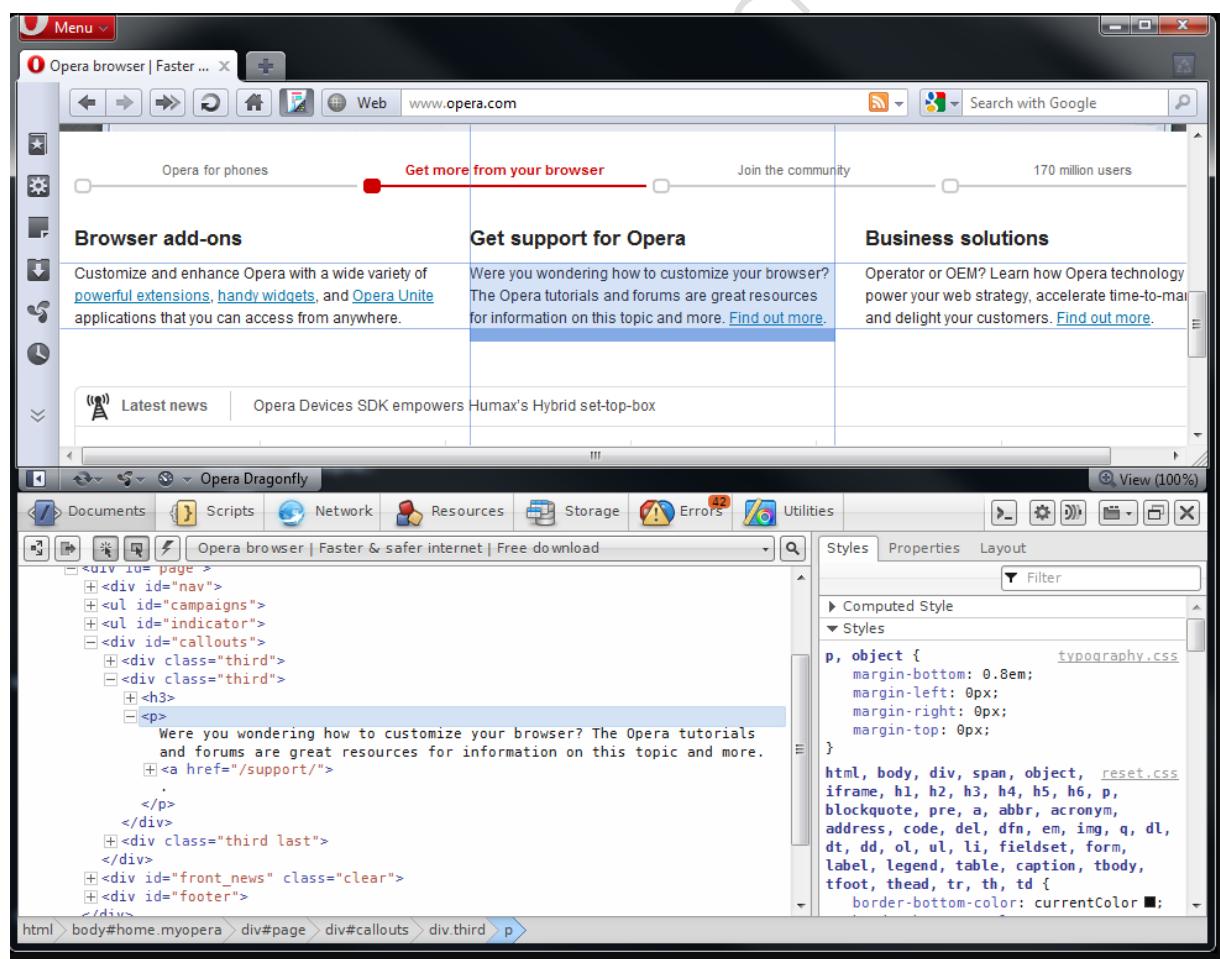
### Dragonfly

Si utiliza como navegador principal el navegador Opera, puede utilizar esta herramienta llamada Dragonfly. Esta herramienta es un entorno de desarrollo que se integra en los navegadores Opera desde la versión 9.5 hacia delante. En su página web podrá conocer mejor sus características.

Además, desde su página principal puede instalar un menú de depuración al mismo tiempo que descargar el navegador Opera. Cuando haya instalado todo, es cuando tendrá acceso total a la herramienta Dragonfly y a todas las posibilidades que ofrece, tan solo debe ir desde el menú a Página, Herramientas de desarrollo, Opera Dragonfly. En caso de que prefiera utilizar un atajo de teclado, pulsar las teclas Control, Mayús y I.

Debe recordar que al utilizar el atajo de teclado solo va a funcionar para la activación de la herramienta. Es decir, en caso de que la herramienta Dragonfly ya estuviera abierta, utilizar el atajo de teclado no hará que se cierre. Para cerrar la herramienta debe utilizar el ratón o, en caso contrario, utilizar el atajo de teclado pulsando las teclas Control y W. Este comando funcionará correctamente si la herramienta Dragonfly se encuentra abierta en una ventana aparte. Sin embargo, en caso de que tenga la herramienta Dragonfly abierta en una pestaña del navegador, solo funcionará si selecciona esa pestaña o tiene seleccionado esa ventana del navegador. Si no selecciona la herramienta Dragonfly antes de utilizar el atajo de teclado, se cerrará todo el navegador.

Si quiera activar otros modos de diseño en el navegador Opera, tan solo, debe ir a la opción Ver, Barra de herramientas, Personalizar, Decorado.

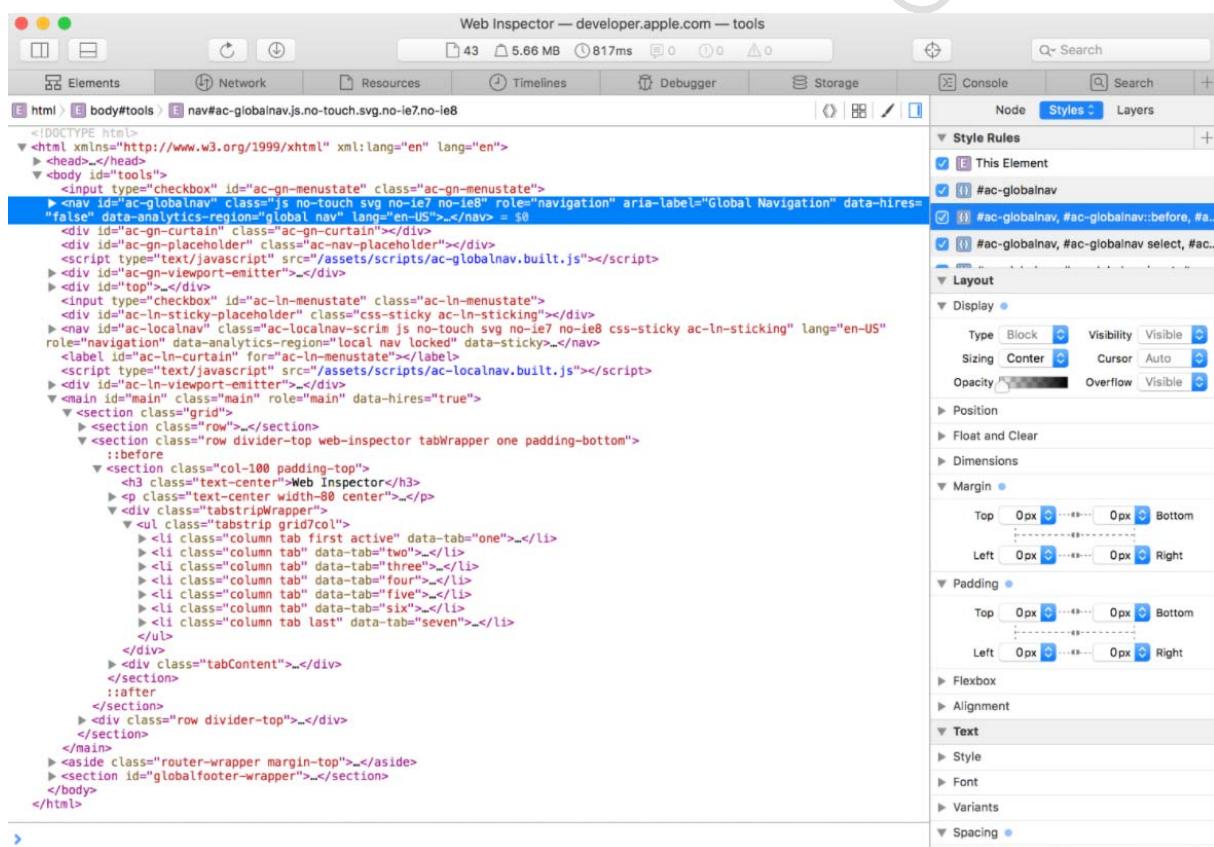


## Inspector Web

Esta herramienta la utiliza en caso de que utilice como navegador principal el navegador Safari.

Para activar esta herramienta, hay que dirigirse a la opción de Preferencias de Safari, Avanzado, y a continuación, activa la casilla de verificación Mostrar el menú de Desarrollo en la barra de menús. Cuando haya finalizado, podrá activar la herramienta Inspector Web. Para ello, utiliza el atajo de teclado utilizando las teclas Control, Mayús y l, al igual que hacía con la herramienta Dragonfly. No podrá utilizar este atajo de teclado para cerrar la aplicación.

Cuando quiera cerrar la aplicación podrá hacerlo con el propio ratón o utilizando el atajo de teclado Control y w. Esto solo funcionará si la herramienta está en una ventana aparte o esté acoplada a la ventana principal. Firebug e Inspector Web tienen el mismo diseño, en cambio su funcionamiento es muy diferente. Por ejemplo, los estilos generados aparecerán en el panel que se encuentra en el lateral derecho de la interfaz.



## 1.2. Hojas de estilo y accesibilidad

Cuando desarrolle una página web, siempre debe tener en cuenta que debe seguir las normas de accesibilidad. La razón de esto es que debe asegurarse de que los usuarios pueden acceder a su página web aun teniendo contextos diferentes, es decir, tengan alguna discapacidad o no. Por ello,

tiene que crear la estructura y el diseño de la página acorde a estas normas de accesibilidad que recomienda la W3C. En cuanto al diseño de la página, va a ver a continuación qué problemas son con los que se puede topar un usuario:

Los usuarios pueden estar utilizando un navegador con una versión anticuada, un navegador de texto o voz, o un sistema operativo alternativo a lo común.

Los usuarios no pueden escuchar, moverse, ver o procesar información de manera sencilla.

Puede haber usuarios que naveguen de manera que no puedan utilizar sus ojos, oídos o manos.

Hay usuarios que tienen dificultad para leer y comprender un texto.

También, hay usuarios que no puedan utilizar un teclado o un ratón.

Puede que el usuario no entienda bien el idioma en el que está redactada la página web o no pueda entenderla con fluidez.

Por último, puede haber usuarios que tengan conexiones lentas a Internet, una pantalla que solo pueda mostrar texto o una pantalla muy pequeña.

Si cuando desarrolle una web, tiene en cuenta estas pautas, esté diseñando una página web totalmente accesible para un diverso grupo de usuarios. En caso de que utilice código CSS para diseñar su web y controlar todos los parámetros de diseño como, por ejemplo, la tipografía de la misma, hará que elimine la etiqueta <Font> del HTML al mismo tiempo que le brinde a la página más accesibilidad para los usuarios y tendrá más control sobre la página pudiendo acortar el tiempo de carga de la misma.

Tiene dos motivos principales por los que se recomienda diseñar una página web accesible:

El primero, es para asegurar la transformación correcta, es decir, puede desarrollar página que se transformen de manera correcta al separar, por ejemplo, el contenido de la estructura del de la presentación, establecer los textos y establecer alternativas para los contenidos multimedia que no se puedan mostrar por múltiples razones, además, debe crear documentos que sean capaces de funcionar en diferentes dispositivos con diferente hardware.

El segundo de los motivos es que el contenido de la página sea comprensible y navegable por los usuarios, es decir, debe crear un contenido que sea muy fácil de comprender y muy simple de navegar. Para ello, puede proporcionar mapas del sitio web, barras de desplazamiento, entre otros.

Lo siguiente que va a ver es cuáles son las normas a seguir para crear una página web accesible:

Debe ofrecer alternativas para todo el contenido auditivo y visual de su página web.

Los textos y gráficos de una página web siempre deben ser comprensibles cuando los visualice en ausencia de color.

Debe utilizar hojas de estilo y marcadores de manera correcta. Debe diferenciar el contenido, de la estructura y de la presentación.

Se debe identificar el idioma utilizado en la página web. Además, añadir marcadores que aclaren las abreviaturas y las palabras extranjeras que pueda contener el texto de su página web.

Si utiliza tablas, debe asegurarse de que se pueden transformar de manera correcta cuando se utilicen navegadores accesibles y otras aplicaciones de accesibilidad.

En caso de que en su web utilice tecnologías muy novedosas, debe asegurarse de que se transformen de manera correcta.

Si incorpora elementos a su página web que se desplacen, parpadeen, o se actualicen automáticamente, debe asegurarse que el usuario es el que tiene el control total de los mismos pudiendo detenerlos cuando desee.

Proporcionar una interfaz totalmente accesible. Si en la interfaz incorpora un elemento que, a su vez, tenga su propia interfaz, debe asegurarse de que sea totalmente accesible o, en caso contrario, ofrecer una alternativa para este elemento.

Debe crear una página donde sus elementos tengan total independencia sobre el dispositivo en el que se muestran. Esto quiere decir que los elementos de la página se puedan activar sin importar el tipo de dispositivo de entrada que se esté utilizando para ello.

Debe utilizar soluciones de accesibilidad provisionales para asegurarse de que los navegadores antiguos puedan funcionar de manera correcta.

Debe seguir las recomendaciones y las tecnologías que ofrece la W3C como guía para crear una página web accesible. En cambio, también podría utilizar una manera alternativa para crear su web de manera accesible.

En caso de que en su página se muestren textos o elementos complejos, debe ofrecer una guía de contexto con la finalidad de orientar al usuario a entenderlos.

Debe proporcionar mecanismos de navegación coherentes, por ejemplo, un mapa del sitio, barras de desplazamiento, entre otros, con la finalidad de aumentar la probabilidad de que le usuario encuentre lo que busca de manera rápida en la página web.

Por último, debe asegurarse de que los textos de su página web sean claros y fáciles de entender, de manera que todos los usuarios puedan entender lo que están leyendo.

En definitiva, el desarrollador web deberá, una vez construida la web, realizar varias validaciones sobre la accesibilidad del sitio web. Para ello, utilizará diversas herramientas especializadas en accesibilidad que puede encontrar fácilmente en Internet. Gracias a estas herramientas podrá validar la sintaxis del código HTML, JavaScript y el CSS de las hojas de estilo, entre otras.

### 1.2.1. Adecuación de las hojas de estilos

Para crear hojas de estilo en cascada que sean accesibles, puede echar mano de diferentes técnicas. Con estas técnicas podrá crear una página web que siga las normas de accesibilidad del contenido web. En cambio, este tipo de técnicas no garantizan la conformidad absoluta sobre accesibilidad, es por ello que puede crear una web accesible utilizando otras técnicas que establezca que los contenidos sean conformes a las normas de accesibilidad establecidas.

Vea a continuación las técnicas que están recomendadas para crear un código CSS accesible:

Si utiliza varias hojas de estilo para un mismo sitio web, es recomendable utilizar un número mínimo de ellas.

Al utilizar hojas de estilo en su página web, se recomienda utilizarlas de manera vinculada en vez de incrustadas en el mismo documento HTML. Además, es mejor utilizar los estilos desde una hoja de estilos que incrustarlos dentro de etiquetas HTML.

Si utiliza varias hojas de estilo para el mismo documento web, establece los mismos nombres de clase para el mismo elemento en todas las hojas de estilo.

Para definir el tamaño de letra lo hará en unidad em.

Para establecer el tamaño de los objetos, utilizará porcentajes y medidas relativas. Solo podrá utilizar medidas absolutas al establecer la longitud, sin embargo, debe conocer las características físicas del medio de salida, por ejemplo, las imágenes de mapa de bits.

Para los contenidos multimedia y textos importantes generados en CSS, debe establecer un texto alternativo, por ejemplo, en las propiedades content o background-image.

Otra cosa a tener en cuenta es que el contenido más importante de la página siempre aparezca dentro del objeto del documento. La razón de esto es que el texto generado por las hojas de estilo no formará parte del código fuente del documento web y, por lo tanto, no podrá ser utilizado por las ayudas técnicas que acceden al DOM nivel 1.

Siempre que desarrolle un sitio web debe establecer un tipo de letra por defecto.

Para controlar el tipo de letra, utilizará las siguientes propiedades: Font, Font-weight, Font-family, Font-size, Font-stretch, Font-variant, Font-style y Font-size-adjust, en vez de los siguientes atributos obsoletos de HTML: Font, basefont, size y face.

Sin embargo, los elementos small y big de HTML sí puede utilizarlos, ya que, no están obsoletos. Estos elementos también controlan el tamaño de letra.

Para darle forma al estilo del texto, podrá utilizar las siguientes propiedades de CSS:

- Efectos de sombreado: text-shadow.

- Minúsculas y mayúsculas: text-transform.
- Subrayado: text-decoration.

No se recomienda utilizar los elementos Blink o Marquee, ya que, están obsoletos y, por ello, el W3C no los considera elementos estándar para HTML.

También, se recomienda utilizar hojas de estilo para dar estilo al texto, en lugar de representar el texto con imágenes. De este modo, el contenido de la web podrá ser visualizado por un número mayor de usuarios.

Por otra parte, puede utilizar las siguientes propiedades para controlar el formato y la posición del texto:

- Pseudoelementos: first-line y first-letter. Estas propiedades se utilizan para hacer referencia a la primera línea o a la primera letra de un párrafo.
- Sangría: text-indent. Esta es la propiedad que utilizará para establecer las sangrías. No se recomienda utilizar blockquote u otros elementos estructurales para este fin.
- Dirección del texto: direction y Unicode-bidi.
- Espaciado entre letras o entre palabras: Word-spacing y letter-spacing. En caso de que quiera crear un espaciado entre letras utilizará letter-spacing. Ponga un ejemplo en el que quiera escribir W E B. Un usuario normal, aunque las letras estén con un espacio entre medio, entendería que ahí pone web. Sin embargo, un usuario que utilice un lector de pantalla leería esa palabra como letras independientes. Si quiera evitar esto, utilizará en su lugar la propiedad Word-spacing.
- Espacio en blanco: White-space. Utilizará esta propiedad para controlar el espacio en blanco que hay entre los elementos de una web.

También, puede utilizar números en vez de nombres para indicar los diferentes colores. Vea cómo:

- Para establecer colores en los bordes, utilizará border-color y outline-color.
- Para establecer un color de fondo, utilizará background-color.
- Para establecer el color del texto, utilizará color
- Para controlar las propiedades de color de los vínculos utilizará :link, :active y :visited.

Al diseñar los colores en una página web, debe asegurarse de que el color de fondo y el del texto tengan un buen contraste para que se pueda leer bien.

La información de la web no solo se debe transmitir por los colores, sino por el contenido.

Se recomienda utilizar las etiquetas `<ul>` para listas desordenadas y `<ol>` para listas ordenadas, de esta manera puede establecer indicaciones contextuales en conjunto con el CSS.

Se recomienda indicar de manera contextual pistas en las listas anidadas no enumeradas. Debe hacer esto hasta que CSS esté completamente aceptado por las aplicaciones de usuario y que permitan controlar la interpretación de las listas de manera total.

Los contenidos de la web deben estar maquetados, posicionados, colocados en divs y alineados utilizando para ello las hojas de estilo y no mediante tablas. Para ello, utilizará las propiedades CSS float y colocación absoluta.

Para que los usuarios puedan controlar el espaciado sin que tenga que añadir espacios adicionales, utilizará las propiedades Font-stretch, Word-spacing, text-align y text-indent. Se recomienda utilizar en CSS text-align:center, en lugar del elemento center.

No es recomendable utilizar espacios de no separación con el elemento &nbsp; en vez de este elemento, utilizará las propiedades margin, margin-right, margin-left, margin-top y margin-bottom para crear los espacios pertinentes en los cuatro lados del elemento.

Para que el usuario sea capaz de controlar la posición visual de cualquiera de los elementos mostrados en su web, utilizará las propiedades position, top, float, right y left.

Al utilizar la propiedad empty-cells, estará permitiendo que los usuarios puedan dejar tablas vacías y, de este modo, ofrecerles un borde en la pantalla o en el papel. Debe recordar que una celda vacía no se deberá llenar nunca con un espacio en blanco o un espacio non-breaking si solo quiera crear un efecto visual.

Se debe, por recomendación, establecer textos similares para todas las imágenes, incluso para aquellas que no se muestren o sean transparentes. Si al diseñar la web no puede utilizar una hoja de estilo para ello y debe utilizar imágenes transparentes, debe utilizar la etiqueta alt para describirlas.



Para crear líneas y bordes, utilizará las hojas de estilo. La mayoría de las veces las líneas y los bordes dan una sensación de separación en una página web. Sin embargo, estos elementos no pueden interpretarse fuera de un contexto visual. Vea a continuación, las propiedades CSS que se recomiendan utilizar para especificar el estilo de los bordes:

- Propiedades para tablas: border-collapse y border-spacing.
- Para los propios bordes: border-style, border-color, border-width y border.
- Para aplicar contornos dinámicos: outline-color, outline2, outline-style y outline-width.

Se recomienda que al crear una presentación estilo, ésta sea comprensible sin aplicar las propias hojas de estilo.

Para poder ofrecerle información a invidentes y personas que utilicen un navegador por voz, debe utilizar las propiedades auditivas que ofrece CSS que verá a continuación:

- Speak-header es la propiedad que se encarga de describir cómo se dice la información que aparece en los encabezados anteriores a una celda de una tabla.
- Speak-numeral y speak-punctuation son las propiedades que se encargan de controlar la manera en la que se dicen los números y los signos de puntuación. Esto mejora la calidad de experiencia de la navegación por voz.
- Volumen esta propiedad se encarga de que pueda controlar el volumen del texto hablado.
- Speak esta propiedad establece qué contenido se pronunciará y si debe ser deletreado o leído como palabras enteras.
- Speech-rate, richness, voice-family, stress, pitch y pitch-range son propiedades que se encargan de controlar la calidad del contenido hablado. De este modo, los usuarios podrán ajustar con todo detalle el sonido y el audio con el que se interpretará el contenido.
- Pause, pause-after y pause-before son propiedades encargadas de controlar las pausas que se hacen antes y después de interpretar el contenido, esto permite separar el contenido de manera que sea más comprensible para el usuario.
- Azimuth y elevatio estas dos propiedades se encargan de dar diferentes dimensiones al sonido, por lo que, los usuarios podrán distinguir voces.
- Cue, cue-after y cue-before estas propiedades definen si un sonido se reproducirá antes y después del contenido. Esto se utiliza para orientar a los usuarios.

Para adaptar el documento web a diferentes dispositivos como sintetizadores de voz, braille, móviles, ordenadores, etc. Se recomienda utilizar diferentes hojas de estilo.

Estas recomendaciones son el resumen de las técnicas CSS para implementar las Pautas de Accesibilidad al contenido en la Web. Si quiera conocer más acerca de estas recomendaciones puede dirigirse a la siguiente [web](#).



### 1.3. Tipos de estilo: incrustados, enlazados, importados, en línea

Como ya vió anteriormente, las hojas de estilo contienen un conjunto de reglas que definen el estilo de los elementos que, anteriormente, se hayan estructurado dentro de un documento HTML y que, a su vez, forman la sintaxis del documento CSS.

Así es como un navegador leerá una hoja de estilo y, al ser interpretada, establecerá en el documento el formato que le haya dado al mismo, y ese mismo formato será el que se muestre al usuario.

Por esta razón, la aparición del CSS fue tan revolucionaria, puesto que, ofrece mucha flexibilidad y muchas opciones para poder realizar la misma tarea, es por ello que las hojas de estilo son fundamentales para el desarrollo de una web y para trabajar junto con el HTML. Vea a continuación algunas de las ventajas que ofrece CSS:

- Técnicas avanzadas y sofisticadas.
- Mejor control sobre la presentación de varios documentos desde una misma hoja de estilo.
- Poder establecer varias presentaciones dependiendo el tipo de medio donde se muestre.
- Presentación del estilo más precisa y controlada.

Ahora que ha visto las principales características de las hojas de estilo, puede darse cuenta de que una de las características más importante que ofrece CSS es la flexibilidad para hacer la misma cosa de múltiples maneras.

En el momento en que aprenda a cómo establecer un estilo en un documento HTML, puede confirmar que la flexibilidad es una de las características más importante de CSS, ya que, puede hacerlo de cuatro maneras diferentes.

Las cuatro posibilidades diferentes que brinda CSS para aplicar un estilo a su documento HTML son las siguientes:

- En línea.
- Incrustados.
- Importados.
- Enlazados.

### **1.3.1. Descripción de los tipos de estilo**

Como ha visto anteriormente, existen hasta cuatro maneras de establecer los estilos en un documento web:

- Mediante hojas de estilo incrustadas en el propio documento.
- Mediante hojas de estilos enlazadas al documento web.
- Mediante hojas de estilo importadas en el documento.
- Y, por último, mediante hojas de estilo en línea.

Para ser capaces de comprender cómo se comportan y cómo funcionan las hojas de estilo, antes debe entender cómo se comporta y cómo funcionan los documentos HTML.

Como ya sabe, los documentos HTML utilizan etiquetas o tags para crear su estructura, estas etiquetas ofrecen información relativa a un tipo determinado de contenido de la página que, a su

vez, es el que realizará la acción sobre el propio documento. Vea a continuación un ejemplo de código:

```
<h1>Esto es un título principal</h1>
```

Como puede ver, ha establecido las etiquetas `<h1>` para el texto del ejemplo anterior. Estas etiquetas van a realizar una acción sobre el texto de manera que van a darle el aspecto de un título principal. Por lo tanto, estas etiquetas serán las encargadas de dar formato al texto, sin estas etiquetas el texto quedaría sin formato.

En cambio, si utiliza una hoja de estilo para formatear el texto, verá más adelante que la forma de utilizar las etiquetas HTML cambia totalmente.

Al utilizar una hoja de estilo, se dará cuenta de que el estilo de los contenidos del texto quedará totalmente diferenciados de la estructura de la página web. Este hecho aporta múltiples ventajas, puesto que, ahorre tiempo a la hora de aplicar los estilos a los elementos y también, al modificarlos o añadir otros estilos.

De este modo, como desarrolladores web, tendrá un control total tanto del estilo de los elementos, como de la estructura HTML del documento. Esto se debe a que el documento HTML quedará totalmente limpio de código y será mucho más sencillo modificar los elementos en un futuro o buscar algún error que pueda aparecer.

### **1.3.2. Enlazar una hoja de estilo externa a un documento HTML**

La manera que más se recomienda para establecer los estilos en un documento HTML es enlazar la hoja de estilo de manera externa al propio documento HTML. Esta es la manera de establecer los estilos para sitios web muy grandes, ya que, puede contener todos los estilos en un mismo documento CSS y aplicarlos a todas las páginas que necesite. Este documento se enlaza en la cabecera del documento HTML que conforma el sitio web.

En definitiva, tendrá un documento HTML con la estructura de su sitio web y aparte, tendrá otro documento CSS con todas las reglas y estilos que darán formato al documento HTML y que se enlazará a este de manera externa. En resumen, tendrá dos archivos diferentes, un documento HTML y un documento CSS.

Para poder enlazar el documento CSS a su documento HTML debe seguir los siguientes pasos:

Primeramente, tendrá que tener un documento HTML con la estructura de la web que está creando. El archivo deberá estar guardado con la extensión .htm o .html.

Lo segundo que hará es crear otro archivo, el cual alojará los estilos CSS. Este archivo deberá estar guardado con la extensión .css.

Y, por último, lo que hará es volver a modificar su archivo HTML, que es en donde enlazará el archivo CSS, y utilizará la etiqueta `<link>` en la cabecera del propio documento.

Vea el ejemplo:

```
<head>  
  
<link rel="stylesheet" type="text/css" href="estilosdemiweb.css" media=screen>  
  
</head>
```

A continuación, cuando el navegador cargue el documento HTML, mostrará los contenidos que hay en él. Sin embargo, a su vez, también interpretará los archivos CSS que están enlazados en el archivo HTML con la etiqueta `<link>`, es por ello que aplicará los estilos que ha definido en el archivo CSS con anterioridad.

Lo que va a ver ahora son los atributos que ha utilizado para enlazar la hoja de estilo en su documento HTML. Pero, esta vez los va a ver en profundidad:

- Href: Este atributo es el encargado de indicar la url del archivo CSS, que es el que contiene los estilos que se aplicarán al documento. La url podrá ser relativa o absoluta dependiendo dónde se encuentre el documento y, además, esta url podrá apuntar a un recurso externo o interno.
- Type: Este atributo se encarga de indicar el tipo de recurso que se ha utilizado. Los valores que se le puede dar a este atributo están estandarizados y, para los archivos CSS, siempre será el mismo `text/css`.
- Rel: Este atributo se encarga de establecer la relación que hay entre el recurso enlazado, es decir, el archivo CSS y el documento HTML. En esta ocasión al ser un archivo CSS el que está enlazado siempre utilizará el valor `stylesheet`.
- Media: Este atributo es un atributo totalmente opcional, se encarga de especificar el medio en el cual se aplicarán los estilos. Vea a continuación qué valores puede utilizar este atributo:
  - Tv: Se utiliza para televisores.
  - Screen: Es el valor por defecto y se utiliza para monitores.
  - Tty: Se utiliza para pantalla en celda de caracteres.
  - Print: Se utiliza para la salida de una impresora.
  - Braille: Se utiliza para mostrarse en dispositivos braille.
  - Projection: Se utiliza para presentaciones en proyectores.
  - Embossed: Se utiliza para impresoras braille.
  - Aural: Se utiliza para sintetizadores de voz.
  - Handheld: Se utiliza para dispositivos móviles como PDA, móviles, etc.
  - Speech: Se utiliza para sintetizadores de navegadores de voz que utilizan las personas discapacitadas.
  - All: Se utiliza para aplicarlo a todos los dispositivos de salida.

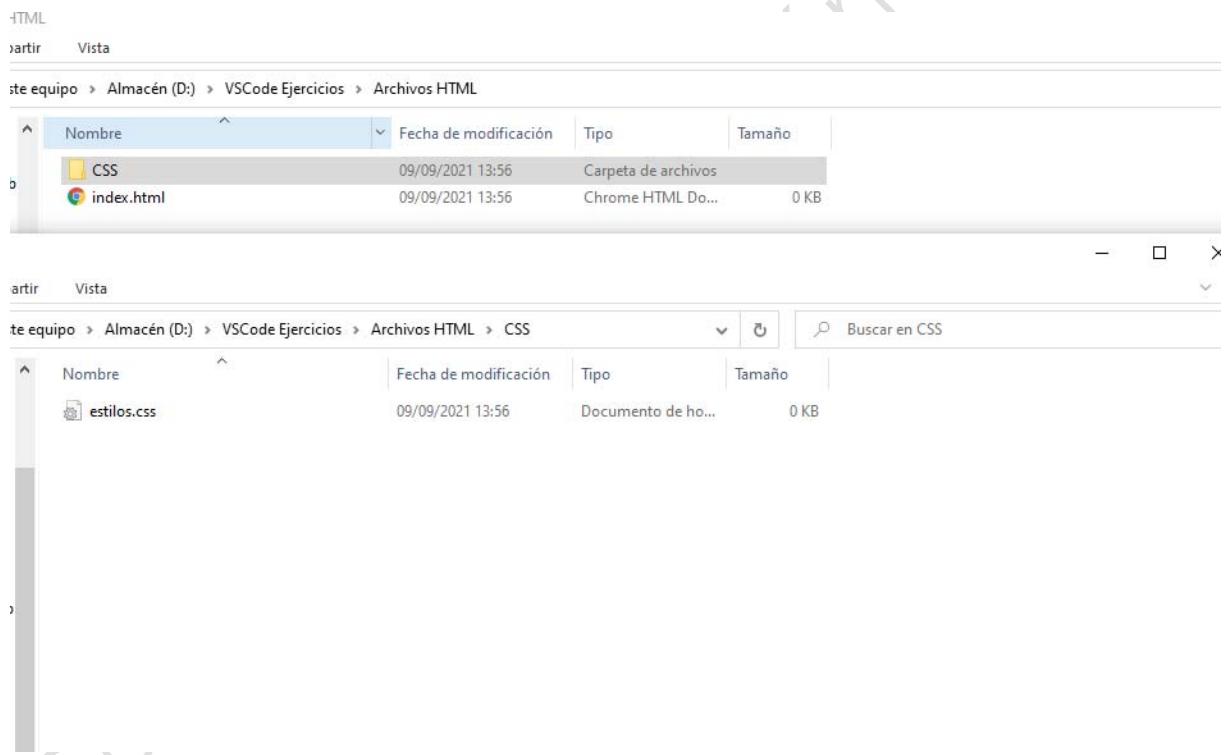
Si quiera aplicar varios medios a la vez, tan solo, debe separarlos por comas o utilizar el valor `all`.

La manera más eficiente y profesional de programar es utilizar un enlace a una hoja de estilo que esté externa al documento HTML, puesto que, como ha visto ya anteriormente, tendrá separados los estilos de la estructura del sitio web.

Realizar esta división hará que el mantenimiento de sitio web sea mucho más rápido y fácil, puesto que, si quiera realizar algún cambio en los estilos, solo debe abrir el archivo CSS y modificar el elemento que quiera, de ese modo, todos esos cambios se aplicarán a los elementos del archivo HTML de manera instantánea.

Además, al realizarlo de esta manera los estilos mantendrán una uniformidad en diferentes documentos HTML, puesto que, no tendrá que repetir las mismas reglas para cada uno de ellos y, por ello, tan solo necesita utilizar un solo documento CSS para establecer los estilos de múltiples documentos HTML.

Vea a continuación un ejemplo de cómo quedaría organizado los estilos y los archivos HTML si utiliza esta manera de enlazarlos:



Como puede observar en la imagen de ejemplo, la manera en la que mejor puede organizar los archivos es:

Primero, creando una carpeta para los archivos HTML

Y, segundo creando otra carpeta para los archivos CSS.

Los archivos de ambas carpetas estarán relacionados mediante el enlace que añadirá en el código HTML.

En definitiva, el código que establezca en el documento HTML será mediante el cual se llamará al archivo CSS que habrá guardado con anterioridad en la carpeta correspondiente, y será el que establecerá el formato de los elementos.

Mediante este enlace que establezca en el documento HTML, le dará las pautas al navegador de los estilos que tiene que mostrar el documento HTML. Lo realmente importante y útil de todo esto es que el archivo CSS se podrá vincular a tantos documentos HTML como sea necesario.

Como ejemplo puede poner un sitio web que esté conformado por 200 documentos HTML, es decir, 200 páginas web. Ahora, imagine que quiera cambiar el color de fondo del sitio web, es decir, el color de fondo de las 200 páginas web. Al utilizar CSS, no tendrá que ir cambiando el color de fondo archivo por archivo, sino que se dirige al archivo CSS y cambia el valor de la propiedad que establece el color de fondo. Por lo tanto, habrá cambiado el color de fondo de 200 páginas en menos de un minuto.

A continuación, va a ver un ejemplo real de cómo enlazar una hoja de estilo externa a un documento HTML:

Lo primero que hará es abrir el editor de texto que utilice normalmente para programar. Creará dos archivos, uno HTML y otro CSS. Vea el contenido de ambos:

Archivo main.html

```
<!DOCTYPE html>

<html>

<head>

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

<title>Mi página web</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css"/>

</head>

<body>

<h1>Aplicando mi primer estilo</h1>

</body>

</html>
```

Archivo estilo.css

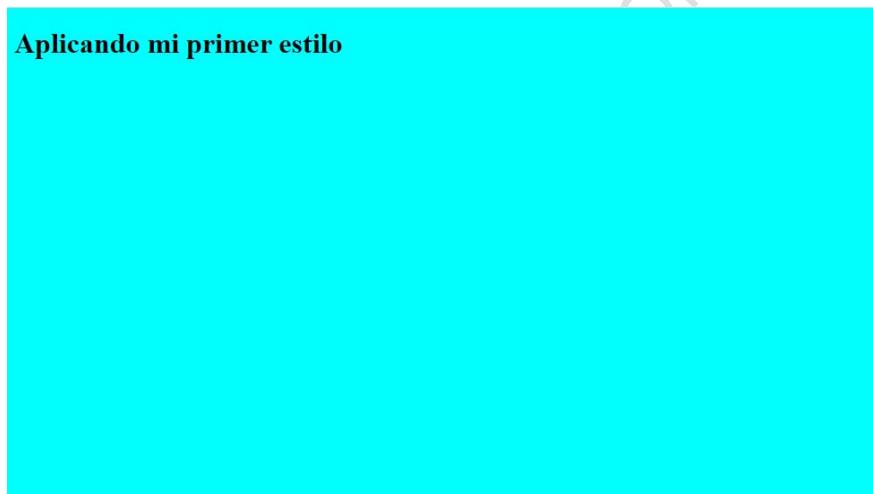
```
body {  
background-color: aqua;  
}
```

Como ha creado una carpeta para cada documento, debe tener en cuenta que la url que utilice para enlazar el archivo CSS de manera correcta al documento HTML será css/estilo.css, ya que, la carpeta CSS estará guardada en la raíz.

Una cosa muy importante que no puede olvidar es poner la extensión del documento en la url, ya que, si no, no funcionará.

Cuando abra el archivo main.html con un navegador verá que éste tiene un fondo celeste. Esto sucede gracias a su archivo CSS que aplica el fondo de pantalla.

La página debería verse como la que se muestra a continuación:



Como puede ver, en la imagen de ejemplo aparece el texto que ha puesto en el documento HTML con un formato de encabezado nivel 1. En cambio, el fondo es celeste, aunque en el documento HTML no haya especificado tal cosa en ninguna de las etiquetas.

El color de fondo se muestra gracias al documento CSS que ha vinculado al documento HTML. En el documento CSS sí que ha especificado que el color de fondo se mostrara en celeste. De esta manera, el navegador al interpretar el documento, interpreta que existe una hoja de estilo enlazada y mediante ésta, establece el color de fondo diferente al que se mostraría por defecto.

### **1.3.3. Incrustar un estilo dentro de un documento HTML**

Como ya vió anteriormente, otra de las maneras de establecer un estilo en un documento es incrustando el estilo dentro del propio documento HTML. Esto significa que tanto el código HTML, como el código CSS estarán en el mismo documento. Esta forma de definir los estilos se diferencia de

---

la anterior en que no tendrá que crear dos documentos diferentes, ya que, no necesita vincular ningún documento externo.

Con esta nueva forma de aplicar los estilos lo que hace es tener en un mismo documento separados los estilos del código HTML.

Al utilizar este método, podrá establecer una pequeña cantidad de estilos o poder incluir estilos muy específicos para ese documento en cuestión. Por ejemplo, como complemento de los estilos generales que pueda establecer desde un archivo externo.

En cambio, utilizar esta manera de establecer los estilos tiene un gran inconveniente, el inconveniente es que cuando quiera modificar alguno de los estilos, debe modificar todas las páginas que contengan este estilo una por una. Por lo tanto, será una tarea que ocupará bastante tiempo.

A continuación, va a ver cómo incrustar estilos dentro de un documento HTML:

Primero, va a partir de la base que tiene este documento HTML:

```
<!DOCTYPE html>

<html>

<head>

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

<title>Mi página web</title>

</head>

<body>

<h1>Aplicando mi primer estilo incrustado</h1>

</body>

</html>
```

Para incrustar su primer estilo, lo que hará es utilizar la etiqueta `<style>` que ofrece HTML. Esto lo hará siempre en la cabecera del documento, es decir, entre las etiquetas `<head>` y `</head>`.

Después, escribirá dentro de la etiqueta `<style>` todos los estilos que desee.

De este modo, los estilos se establecerán en una parte del documento HTML, específicamente, en su cabecera. Y, así, en el cuerpo del documento solo se hallará la estructura de la web. Así es como quedará el documento finalmente:

```
<!DOCTYPE html>

<html>

<head>

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

<title>Mi página web</title>

<style type="text/css">

body {

background-color: chartreuse;

}

</style>

</head>

<body>

<h1>Aplicando mi primer estilo incrustado</h1>

</body>

</html>
```

Como puede comprobar en el ejemplo, además de utilizar la etiqueta `<style>`, también se ha utilizado el atributo `type` que es el encargado de indicar el recurso que va enlazado. Los valores de este atributo, como ya se mencionó anteriormente, están estandarizados y, para el CSS, siempre se utilizará el valor `text/css`. Dentro de la etiqueta `<style>` es donde ha establecido el estilo de la página y, en este caso, ha puesto el color de fondo verde.

Además, también puede utilizar la etiqueta `<style>` dentro del `body` del documento HTML. Sin embargo, a diferencia del método que acaba de ver, el estilo se podría aplicar de manera individual a cualquier elemento del `body`. A este método le denomina estilo en línea.

El estilo en línea es el estilo menos recomendado para utilizar, esto sucede debido a su inflexibilidad y su imposibilidad de separar el contenido de los estilos.

A continuación, va a ver un ejemplo de cómo utilizar un estilo en línea:

Primero, debe tener un documento HTML con la estructura básica.

Después, para que pueda utilizar un estilo en línea, debe declarar un único lenguaje de hojas de estilo para el documento. Para ello, empleará la etiqueta meta en la cabecera del documento. Vea el ejemplo:

```
<meta http-equiv="content-style-type" content="text/css">
```

Lo siguiente que hará es utilizar la etiqueta `<style>`, que aplicar a cualquier elemento del cuerpo del documento HTML, menos en los elementos `<param>`, `<script>` y `<basefont>`. La manera de utilizar esta etiqueta puede verla a continuación:

```
<p style="color:green; Font-family='Verdana', serif">Este párrafo utiliza estilos en verde con fuente Verdana.</p>
```

Como puede ver en el ejemplo, ha añadido el atributo `style` dentro de la propia etiqueta `<p>` y, posteriormente, ha añadido los estilos con cada uno de sus valores, para finalizar, ha separado cada atributo con un punto y coma.

Si coge como ejemplo el ejemplo anterior, y quiera hacer lo mismo con el estilo, quedará un código como el siguiente:

```
<!DOCTYPE html>

<html>

<head>

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

<title>Mi página web</title>

</head>

<body>

<p style="background-color: aqua;">Aplicando mi primer estilo incrustado</p>

</body>

</html>
```

#### **1.3.4. Importar una hoja de estilo desde un documento HTML**

La siguiente manera de añadir los estilos CSS a su documento que va a ver es importando una hoja de estilo desde un documento HTML. Antes ha visto cómo enlazar un documento CSS externo con la

---

etiqueta <link>. Sin embargo, para utilizar este método hará lo mismo que hizo anteriormente, pero sin utilizar la etiqueta <link>.

Para enlazar un archivo CSS externo, esta vez, utilizará la etiqueta <style> y con ella, podrá importar los estilos que se hayan establecido en la hoja de estilos externa para que se apliquen a su documento HTML.

Además, para importar la hoja de estilos a su documento HTML, también debe utilizar la regla @import. Esta regla es la que va a permitir importar una hoja de estilos externa dentro de una etiqueta <style>.

En definitiva, al utilizar la etiqueta <style> y la regla @import, podrá importar la hoja de estilos que haya creado para aplicar los estilos a su documento HTML.

A continuación, va a ver los pasos que tiene que seguir para importar una hoja de estilos en un documento HTML utilizando las etiquetas que ha aprendido:

Lo primero que debe hacer es tener un documento HTML con su estructura más simple, además, tendrá que tener aparte un documento CSS externo, como hizo en el primer ejemplo. Ambos archivos se ubicarán en sus carpetas pertinentes.

La sintaxis que utilizará es la siguiente:

```
<style>  
@import url("/css/estilos.css");  
</style>
```

Ubicará esta sintaxis, siempre, en la cabecera del documento HTML en el que quiera importar los estilos.

Debe saber que la regla tipo @import precede a cualquier otra regla que pudiera tener el CSS, menos a la regla @charset.

Además, puede utilizar otra forma de sintaxis para importar la hoja de estilos. Vea a continuación:

```
<style>  
@import  
"estilos.css";  
</style>
```

Utilizando ambas sintaxis, el navegador importará de igual manera las hojas de estilo y los aplicará en el HTML.

Por otra parte, también puede importar un archivo teniendo en cuenta el tipo de medio para el que se destinará, por ejemplo, si quiera aplicar unos estilos para una hoja que va a ser impresa y otros estilos para la página web, utilizará el siguiente código:

```
<style>  
@import url ("imprimir.css") print;  
@import url ("web.css") screen;  
</style>
```

De este modo, habrá importado dos hojas de estilos diferentes, una para la versión impresa de la página y otra para la versión por pantalla de la página, es decir, la página web.

### **1.3.5. Importar una hoja de estilo a través de un archivo con estilos**

También, puede utilizar la regla @import para importar los estilos de una hoja de estilos en otra hoja de estilos.

Como la regla @import permite importar hojas de estilo desde otras hojas de estilo, tendrá dos documentos. Tendrá el documento HTML con la estructura de la web y, por otro lado, tendrá una hoja de estilo con los estilos que haya establecido y, además, donde haya añadido la regla @import para poder establecer más estilos al documento CSS.

Vea las dos maneras en las que puede utilizar esta regla:

Primera manera:

```
@import "/css/estilos.css";
```

Segunda manera:

```
@import url ("/css/estilos.css");
```

Siempre debe especificar esta regla antes que cualquier otra regla que contenga la hoja de estilo, si no, las reglas que estén especificadas en la propia hoja de estilo prevalecerán sobre las reglas de las demás hojas de estilo, de manera que puede contradecir algunas reglas y no funcionar adecuadamente.

### 1.3.6. Utilización y optimización de los tipos de estilo

Ahora que ha visto todas las maneras en las que implementar una hoja de estilo en su documento HTML, debe aprender cuándo utilizar cada una de estas formas dependiendo de la finalidad que quiera que tengan los estilos.

Como método más recomendable es el método de definir el código CSS de manera externa y enlazar los estilos mediante la etiqueta <link>. Este método es el más recomendado debido a la flexibilidad que ofrece. Además, es el método que utiliza la mayoría de desarrolladores y diseñadores web, puesto que, permite modificar o aplicar estilos de una manera rápida y homogénea a todas las páginas que requieran de este estilo.

Además, el mantenimiento de la página web con este tipo de método es algo muy sencillo de realizar, haciendo que las modificaciones en los estilos sean instantáneos y rápidos sin importar a cuántas páginas afecte esta modificación, ya que, modificando el archivo CSS, se modificarán todas las páginas a la vez.

Otro método que tiene para enlazar los documentos CSS a los documentos HTML es utilizar la regla @import, aunque cabe mencionar que este no es un método tan utilizado como el anterior. Este método se suele utilizar cuando quiera importar un estilo a un documento HTML, pero teniendo varias opciones de estilo en un mismo documento. Es decir, si quiera, por ejemplo, tener un estilo para la web por pantalla y otro para imprimir, el usuario podría elegir cualquiera de los dos que le venga mejor para mostrar la web a su conveniencia.

Otro de los métodos que ha aprendido es establecer los estilos sin tener que enlazarlos de manera externa, es decir, que los estilos se establecerán en el propio documento HTML utilizando la etiqueta <style> para ello. Este método únicamente debe utilizarlo cuando quiera aplicar pocos estilos y muy específicos. Sin embargo, no se recomienda su uso, ya que, lo correcto es separar la estructura de los estilos de una web.

Podrá utilizar la etiqueta <style> en el encabezado del documento HTML donde establezca los estilos correspondientes. O, también puede utilizar la etiqueta <style> en línea, es decir, en los propios elementos como ya vió anteriormente.

Si utiliza la etiqueta <style> en el encabezado del documento HTML, cambiará algunos de los estilos que estaban predefinidos. El gran problema de utilizar de esta manera los estilos, es que, si está desarrollando un sitio web con un gran número de páginas, tendrá que ir modificando los estilos documento por documento, de esa manera, perdería muchísimo tiempo y desaparecería la principal finalidad de las hojas de estilos que es facilitar la modificación de los mismos.

Este método de estilos en línea es recomendado, únicamente, cuando tenga que modificar algo muy específico en un solo documento HTML de todo el sitio web.

## 1.4. Selectores y reglas de estilo

Para saber cómo funciona realmente CSS, antes debe entender el concepto de selector. Como anteriormente ya vió, puede separar la estructura de una web del estilo de la misma utilizando para ello hojas de estilo. Sin embargo, aún no ha visto cómo asignar los estilos a las diferentes partes del contenido como, por ejemplo, párrafos, títulos, tablas, formularios, entre otros.

Mediante los selectores podrá asignar los estilos a los diferentes elementos de una web. Los selectores son los encargados de unir los estilos que defina para los elementos del documento HTML.

Los selectores son los encargados de especificar qué estilos van asignados a qué elementos del documento HTML. En caso de que haya creado un estilo para aplicárselo a todos los párrafos de una página o si quiera que un párrafo específico utilice un estilo concreto, utilizará los selectores para ello.

Existen dos tipos de selectores:

Selectores simples.

Selectores contextuales.

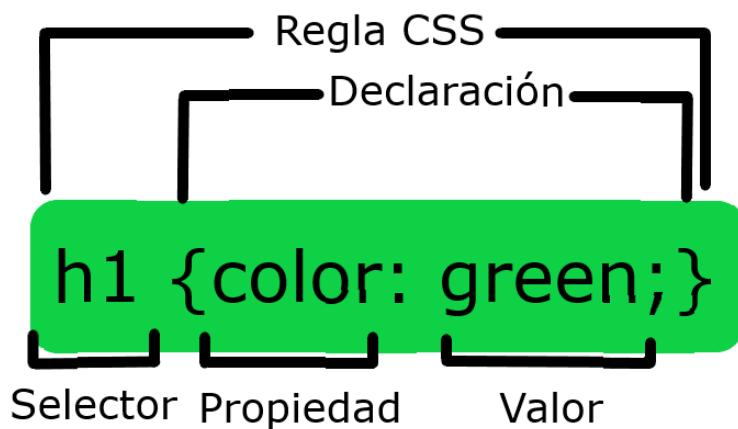
Los selectores simples indican el tipo de elemento, mientras que los selectores contextuales indican su posición dentro de la página web.

Por lo tanto, puede decir que los selectores son el eje principal que hace que el CSS funcione, ya que, sin los selectores no podría establecer ningún estilo a ningún elemento específico, solo podría incorporar los estilos a sus atributos y, en ese caso, todo sería más complicado y trabajoso.

En resumen, gracias a la existencia de los selectores puede seleccionar elementos específicos y conjuntos de elementos enormes a los que puede aplicarles un estilo y realizar diversas modificaciones en el documento mediante unas pocas líneas de CSS.

### 1.4.1. Estructura de los estilos

A la hora de crear un documento CSS, va a utilizar un conjunto de términos que van a ayudar a nombrar cada una de las partes que usará cuando cree los estilos. Va a ver un ejemplo muy básico de un estilo sencillo y las partes que lo forman:



Como puede observar en la imagen, hasta los estilos más básicos en CSS se componen de diferentes términos para denominar sus partes como, por ejemplo, Selector, Regla, Valor o Propiedad. La Regla de estilos se refiere a la estructura completa que conforma en sí un estilo CSS.

Cada regla de un estilo CSS está formada por las partes que verá a continuación:

- Selector.
- Símbolo de llave de apertura.
- Declaración terminada en punto y coma.
- Símbolo de llave de cierre.

Normalmente, una hoja de estilos tiene un conjunto de reglas donde cada una de ellas se pueden aplicar a varios selectores diferentes y en cada declaración podrá incluir tantas propiedades y valores como necesite.

En el antiguo estándar de CSS 2.1 se llegaron a definir 115 propiedades en total con cada uno de sus valores permitidos. Sin embargo, en el nuevo estándar CSS 3 existen unas 239 propiedades.

Los selectores que ha visto anteriormente son muy importantes para comprender la estructura de los estilos. Sin embargo, también debe tener en cuenta para entender la estructura, otras dos características muy importantes:

### Herencia

Cuando se trata de CSS es muy importante hablar de la herencia entre los elementos y la herencia de los estilos y las propiedades. En el caso del CSS los elementos hijos, normalmente, son los que heredan el estilo de sus padres.

Por ejemplo, un elemento que contenga a otro elemento le proporcionará sus características siempre y cuando el elemento que está contenido no tenga unas características que contradigan las de su padre. Uno de los ejemplos más utilizados es el del tamaño del texto. El tamaño del texto se

define con el selector body, por ello, es heredado al resto de elementos, por lo tanto, si cambia una regla de ese estilo, estará cambiando el estilo de todos los elementos que la heredan.

Debe saber que no todas las propiedades se heredan de manera automática, sin embargo, puede utilizar la palabra inherit para asegurarse de que sea una propiedad heredable. En definitiva, si el valor de una propiedad es inherit, sabrá que esta propiedad será heredada de padres a hijos.

### Cascada

Que las siglas CSS signifiquen hojas de estilo en cascada no es una mera coincidencia. Esto se debe a que se asemeja a la cascada de un río la cual arrastra todo lo que encuentra a su paso, por ello, en CSS las propiedades de los elementos que se sitúan más alto dentro de la jerarquía de la página, traspasan esas propiedades a sus hijos. En definitiva, un elemento recibe estilos de la propia página, mientras hay otros estilos que están definidos en el navegador y que influyen con los anteriores.

Será el navegador el que decidirá qué estilos son los que se deben aplicar de todos esos estilos definidos. Lo que intenta la cascada es poner en orden toda esa información que proporciona la hoja de estilos. Serán las reglas definidas quienes decidirán los estilos que se muestran y cuáles no se muestran.

Aunque, como siempre, hay excepciones y en CSS puede saltarse la cascada si así lo necesita. Para ello, debe utilizar !important, que será una expresión que definió en la declaración de un estilo. Lo que consigue es que la propiedad donde haya utilizado esta expresión, tenga preferencia sobre cualquier otra, aunque esta otra sea heredada. Vea un ejemplo a continuación:

```
Body p {color: #01DF74}  
  
p.black {color: #000 !important}
```

En este ejemplo, puede ver como el color de los párrafos heredado es el verde, sin embargo, al establecer la expresión !important, ha hecho que el color de los párrafos sea negro.

#### 1.4.2. Sintaxis básica de los estilos

Como ha visto previamente, los selectores se pueden dividir en selectores simples y selectores contextuales. A continuación, va a ver en profundidad cada uno de ellos y los comprende mejor mediante algunos ejemplos:

### Selectores simples

Los selectores simples son los encargados de identificar los elementos HTML a los cuales se les asignará un estilo dependiendo del tipo de elemento que sea o el atributo que contenga. Vea un ejemplo donde podrá entender mejor este concepto:

```
Td {color: green}
```

Como puede comprobar, este selector hace posible que en todas las celdas de las tablas de una página el texto aparezca de color verde, por ello, el criterio para este selector es que éste sea aplicado a todos los elementos td del documento HTML. Por ello, este selector hará efecto a todos los elementos td de la página en la que se aplique este estilo CSS.

Vea los tipos de selectores simples que existen:

Selectores de tipos.

Selectores de clases.

Selectores de ID.

Selectores universales.

### **Selectores de tipos**

Los criterios que se aplican a los selectores de tipo es un elemento HTML. Por ejemplo, en el ejemplo que vió anteriormente, el criterio era que todos los elementos td tuvieran el texto en color verde. El elemento que le aplique como criterio a un selector puede ser cualquiera, incluso el elemento body en el que los estilos se aplicarían a todos los elementos contenidos en el mismo. Vea un ejemplo:

```
Body {Font-size: 10px;}
```

En este ejemplo, todos los elementos que se ubiquen dentro del body tendrán un tamaño de letra de 10 píxeles, a no ser que se indique lo contrario.

Los selectores de tipos lo que hacen es seleccionar todos los elementos de la página que coincidan con una determinada etiqueta HTML. Vea un ejemplo:

```
P {}
```

Este selector seleccionaría todos los elementos tipo párrafo de una página. Si quiera utilizar este tipo de selector, tan solo, escribe el nombre de la etiqueta HTML del elemento que quiera seleccionar, pero sin los signos < y >.

Vea un ejemplo donde se seleccionan los elementos h1, h2 y p y se les aplica un determinado estilo:

```
H1 {color: green;}
```

```
H2 {color: yellow;}
```

```
P {color: Orange;}
```

Otra de las cosas que puede hacer es aplicar un mismo estilo a dos elementos diferentes, esto se debe a que puede encadenar los selectores que necesite. Vea un ejemplo:

```
h1 {  
    color: #8de923;  
    font-weight: normal;  
    font-family: Verdana, Geneva, Tahoma, sans-serif;  
}  
  
h2 {  
    color: #8de923;  
    font-weight: normal;  
    font-family: Verdana, Geneva, Tahoma, sans-serif;  
}  
  
h3{  
    color: #8de923;  
    font-weight: normal;  
    font-family: Verdana, Geneva, Tahoma, sans-serif;  
}
```

Como puede ver en este ejemplo todos los encabezados tienen el mismo estilo. Es por ello que CSS permite encadenar todas las reglas individuales en una sola regla con un tipo de selector múltiple. Para ello, incluirá todos los selectores necesarios y los separará con una coma. Vea cómo quedaría el resultado de esto:

```
h1, h2, h3 {  
    color: #8de923;  
    font-weight: normal;  
    font-family: Verdana, Geneva, Tahoma, sans-serif;  
}
```

Cuando haga hojas de estilo complejas, debe saber que lo normal es encadenar estilos que tengan propiedades idénticas. Sin embargo, posteriormente, puede definir otras propiedades más específicas para los mismos elementos en caso en que así lo precise. Vea un ejemplo donde lo entenderá mejor:

```
h1, h2, h3 {  
    color: #8de923;  
    font-weight: normal;  
    font-family: Verdana, Geneva, Tahoma, sans-serif;  
}  
  
h1 {font-size: 1.5em;}  
  
h2 {font-size: 2em;}  
  
h3 {font-size: 1.3em;}
```

Como puede observar, primero ha encadenado los elementos y les ha aplicado las propiedades que tenían en común y, posteriormente, ha seleccionado cada uno de los elementos uno por uno para aplicarles el tamaño de letra específico que cada uno debe tener. Por lo tanto, lo que finalmente tiene es un bloque con todos los estilos en común de esos tres elementos y otro bloque con sus propiedades específicas de cada uno de ellos.

## Selectores de clases

Este tipo de selectores son los selectores que más se utilizan. Esto se debe a que permiten asignar un estilo a cualquier elemento que se encuentre en el HTML, tan solo, utilizando el atributo class. A su vez, este atributo permite asignar cualquier estilo que definirá en CSS a cualquier elemento de HTML. Para utilizar este tipo de selector, tan solo, pondrá un punto delante del nombre que le haya dado al elemento en HTML. Vea un ejemplo:

```
.colorVerde {color: green;}
```

Mientras tanto, en el código HTML habrá utilizado el atributo class en los elementos en los que sea necesario, vea el ejemplo:

```
<font class="colorVerde">Texto en color verde</font>  
  
<td class="colorVerde">Texto de una celda en color verde</td>  
  
<input type="text" class="colorVerde" value="Texto">
```

A continuación, va a ver un ejemplo más sobre el selector de clase. Vea el siguiente código HTML:

```
<body>  
  
<p>Párrafo primero</p>  
  
<p>Párrafo segundo</p>  
  
<p>Párrafo tercero</p>  
  
</body>
```

Ahora, imagine que solo quiera aplicarle un estilo específico a un solo párrafo. Para ello, utilizará el atributo class en ese elemento determinado para posteriormente indicar a qué elemento aplicar la regla CSS. Vea el ejemplo:

```
<body>  
  
<p class="conEstilo">Párrafo primero</p>  
  
<p>Párrafo segundo</p>  
  
<p>Párrafo tercero</p>  
  
</body>
```

Posteriormente, en el archivo CSS es donde creará la nueva regla a la que le llame conEstilo, y le aplicará los estilos que desee para ese elemento en especial. Para que el navegador pueda interpretar este tipo de selector, aplicar un punto delante del nombre que le ha dado quedando de la siguiente manera:

```
.conEstilo {color: green;}
```

Al establecer esta regla, el navegador interpretará el selector .conEstilo como si fuera cualquier elemento del documento que tenga como atributo class igual a conEstilo. En este ejemplo es el primer párrafo el que contiene tal atributo, por lo tanto, será al que se le aplique el estilo que ha definido en CSS.

Además, otra de las características que tienen los selectores de clase es que varios elementos pueden utilizar el mismo valor del atributo class. Vea el siguiente ejemplo para comprender mejor esto:

```
<body>  
  
<p class="conEstilo">Párrafo primero</p>
```

```
<p>Párrafo <a href="#" class="conEstilo">segundo</a></p>  
<p>Párrafo <em class="conEstilo">tercero</em></p>  
</body>
```

A la hora de diseñar páginas web complejas, los selectores de clase serán imprescindibles, puesto que, permiten controlar la selección de elementos con mucha precisión.

Además, los selectores de clase también se pueden reutilizar para elementos diferentes, tal y como ha visto en el ejemplo anterior.

### Selectores de ID

Este tipo de selectores funcionan mediante el atributo id de los diferentes elementos y, por ello, asignan los estilos a los elementos que tengan ese id específico. Es por ello que tan solo servirían para un solo elemento. Vea el ejemplo:

```
#parrafo2 {Font-size: 10px;}
```

Por otro lado, establecerá el elemento de la siguiente manera en el código HTML:

```
<p id="parrafo2">Este es el texto del párrafo dos.</p>
```

Aunque estos selectores parezcan muy limitados por solo poder darles un solo uso, son muy importantes para que pueda trabajar con capas.

Puede utilizar los selectores de id cuando necesite aplicar estilos CSS a un solo elemento de la página web. Podría utilizar el selector de clase para el mismo cometido, sin embargo, los selectores de id son más eficiente para esta labor.

Los selectores de id permitirán seleccionar un elemento de la página mediante su atributo id. De este modo, podrá seleccionar un único elemento de la página, puesto que, el valor del atributo id de un elemento debe de ser único para el documento.

La sintaxis que utilizará en su hoja de estilos para los selectores de id es muy similar a la que utilizaba con los selectores de clase, lo único es que, en vez de un punto, utilizará una almohadilla en su lugar. Vea un ejemplo:

```
#conEstilo {color: green;}
```

```
<p>Párrafo primero</p>  
<p id="conEstilo">Párrafo segundo</p>  
<p>Párrafo tercero</p>
```

En este ejemplo puede ver que el selector `#conEstilo` tan solo selecciona el segundo párrafo donde el atributo `id` es igual a `conEstilo`.

### **Selector universal**

Los selectores universales se utilizan para asignar un estilo a todos los elementos que pueda haber en la página. Vea un ejemplo:

```
* {color: green;}
```

Cualquier elemento que haya en la página tendrá este estilo definido.

Vea a continuación otro ejemplo en el que podrá ver cómo asignar los espacios y márgenes de manera universal en su página web:

```
* {  
    Margin: 0;  
    Padding: 0;  
}
```

#### **1.4.3. Utilización de elementos y seudoelementos**

Como ya sabe utilizar la sintaxis y la estructura básica de los estilos, en este epígrafe va a aprender cómo utilizar los elementos y seudoelementos del código CSS.

#### **Elementos**

A veces, va a ocurrir que encuentre situaciones en las que debe restringir aún más el alcance de los selectores de `id` para un elemento específico.

Del mismo modo que hacía con los selectores de clase, puede también limitar el alcance del selector gracias a la combinación de ambos selectores. Vea el ejemplo de cómo realizar esta labor:

```
P#advertencia {color: yellow;}
```

Como puede observar en este ejemplo, esta regla CSS solo se aplicará al elemento que sea de tipo párrafo y que su `id` sea igual a `advertencia`.

Puede pensar que no tiene ningún sentido limitar el alcance de un selector de `id`, sin embargo, el selector `p#advertencia` solo cobra sentido cuando el mismo documento CSS se aplica en diversos documentos HTML diferentes.

En el caso en que se utilice el mismo documento CSS para diversos documentos HTML puede ocurrir que en algunos documentos HTML existan elementos con un atributo igual a la advertencia pero que éstos no sean párrafos, en ese caso, esta regla no se aplicará sobre esos elementos.

## Seudoelementos

Los seudoelementos permiten acceder a elementos específicos a los que no podría acceder de otra manera, ya que, no podría identificarlos mediante HTML. Vea a continuación cuáles son estos elementos:

- **:before y :after:**

Estos elementos permiten acceder al elemento posterior y anterior de otro elemento. En caso de que una página contenga diversos títulos, podrá ahorrar tiempo y errores gracias al uso de estos seudoelementos. Vea en el siguiente ejemplo cómo el elemento anterior a h1, que es p, se verá a un tamaño un 80% más pequeño que el del título:

```
P {Font-size: 12pt;}
```

```
H1:before {Font-size: 80%;}
```

- **:first-letter**

Podrá acceder a la primera letra de un elemento aplicando este seudoelemento. Por ejemplo, cuando quiera hacer un párrafo donde la primera letra sea completamente diferente al resto, utilizará este seudoelemento. Vea un ejemplo donde modificar el tamaño de la primera letra del elemento párrafo:

```
P:first-letter {Font-size: 16px;}
```

- **:first-line**

Este seudoelemento se utiliza para seleccionar la primera línea de un elemento. Este seudoelemento es muy interesante cuando quiera destacar de alguna manera la primera línea de cada párrafo. Vea un ejemplo a continuación:

```
Td:first-line {color:green;}
```

Por último, debe saber que los elementos :before y :after solo funcionarán en los navegadores que tengan la versión 6 en adelante. Esto se debe a que estos elementos son de CSS2. En cambio, los seudoelementos :first-letter y :first-line son compatibles con todos los navegadores modernos.



TOME NOTA

**Debe tener presente siempre la compatibilidad del código que desarrolle con los diferentes navegadores. Para ello, puede utilizar múltiples herramientas que ayudan con esta tarea.**

#### 1.4.3. Utilización de clases y seudoclases

Ahora, que ya sabe utilizar las clases, su sintaxis y su estructura básica, va a aprender cómo utilizar tanto las clases, como las seudoclases.

##### Clases

Al igual que pasaba con los selectores de id, con los selectores de clases también ocurrirá a veces que tendrá que limitar su alcance para un elemento específico. Para ello, tome el ejemplo anterior:

```
<body>  
  
<p class="conEstilo">Párrafo primero</p>  
  
<p>Párrafo <a href="#" class="conEstilo">segundo</a></p>  
  
<p>Párrafo <em class="conEstilo">tercero</em></p>  
  
</body>
```

Ahora, piense que quiera establecer un estilo determinado a un párrafo en el que su atributo class sea igual conEstilo. Para ello, hará una combinación de un selectore de tipo con un selector de clase, de este modo conseguirá un selector mucho más específico. Vea el ejemplo:

```
p.conEstilo {color:green;}
```

Si analiza esta combinación, el selector p.conEstilo será interpretado por el navegador de igual manera que los elementos de tipo párrafo que tengan como valor del atributo id .conEstilo. De igual manera sucede con el selector a.conEstilo que seleccionará solo los elementos de enlace que contengan como valor conEstilo en el atributo class.

Resumiendo, llega a la conclusión de que el atributo .conEstilo es lo mismo que \*.conEstilo, es por ello que la mayoría de desarrolladores web no utilizan el símbolo \* al establecer selectores de clase normal.

Habrá ocasiones en las que aplicar estilos de varias clases CSS sobre un mismo elemento. Para ello, utilizará la misma sintaxis que ha aprendido separando los valores del atributo class con espacios en blanco. Vea un ejemplo:

```
<p class="especial conEstilo advertencia">Párrafo</p>
```

En este elemento se aplicarán los estilos que haya definido en las reglas .especial, .conEstilo y .advertencia. Por lo tanto, el párrafo se verá con la letra en color verde, negrita y con un tamaño de 10 píxeles. Vea el ejemplo:

```
.advertencia {color: green;}
```

```
.conEstilo {font-size: 10px;}
```

```
.especial {Font-weight: bold;}
```

```
<p class="especial conEstilo advertencia">Párrafo</p>
```

Del mismo modo, si necesita que un elemento específico disponga de un atributo class con más de un valor, podrá especificar más aún su estilo utilizando un selector más avanzado:

```
.advertencia {color: green;}
```

```
.advertencia.conEstilo {color: yellow;}
```

```
.conEstilo {font-size: 10px;}
```

```
.especial {Font-weight: bold;}
```

```
<p class="especial conEstilo advertencia">Párrafo</p>
```

En este caso, lo que sucedería es que el color de letra en vez de ser verde, pasaría a ser amarillo. Esto se debe a que el navegador interpreta que el selector de clase múltiple .advertencia.conEstilo se aplicará a los elementos que tengan en su atributo class al menos los valores advertencia y conEstilo.

## Seudoclases

Las seudoclases referencian a elementos que existen más allá de la propia estructura del documento y que el desarrollador no puede alcanzar. Por ejemplo, los enlaces visitados, o el primer hijo de cada elemento. Vea a continuación las seudoclases que existen:

- :link y :visited

Con esta seudoclase puede definir un estilo para los enlaces no visitados y para los enlaces visitados. Para ello, combinará la seudoclase con el selector de tipo para los elementos de tipo a, que son los enlaces. Vea el ejemplo:

A:link {color: green;}

A:visited {color: yellow;}

- **:first-child**

Esta seudoclase selecciona el primer hijo de un elemento que defina como su contenedor. Esta seudoclase es muy útil cuando quiera destacar títulos que están anidados en otros tipos de elementos y no quiera que pasen desapercibidos en la página web. Vea un ejemplo:

H1 :first-child {Font-weight: bold;}

- **:hover**

Esta seudoclase permite acceder al enlace por el que pasa el puntero del ratón. Mediante esta seudoclase puede lograr el efecto de cambiar el color a un enlace por el que pasa el ratón, sin embargo, para ello, debe asignar esta regla a un elemento a en vez de al contenedor del mismo. Vea el siguiente ejemplo:

A:hover {text-decoration: none;}

Puede utilizar seudoclases en todos los elementos HTML, sin embargo, debe tener en cuenta la incompatibilidad de algunos navegadores. Es por ello que solo se recomienda su uso para formatear enlaces.

A continuación, va a ver un ejemplo sencillo donde se cambia la letra de color cuando pasa el ratón por un enlace:

```
<!DOCTYPE html>

<html>

<head>

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

    <title>Ejemplo de seudoclase :hover</title>

<style>

    a {color: green;}

    a:hover {color: greenyellow;}

</style>
```

```
</head>

<body>

<a href="">Enlace</a>

</body>

</html>
```

## 1.5. Atributos de estilo para fuentes, color y fondo, texto y bloques ( párrafos )

La mejor manera que hay de definir los atributos estéticos en HTML es, indudablemente, mediante una hoja de estilos. Gracias al CSS los diseñadores web tienen todo lo necesario a su disposición para crear una estética atractiva de una página web y, además, poder separar la estructura del documento del estilo del mismo.

Como ha visto anteriormente, gracias a las propiedades de estilo podrá establecer la sintaxis de código CSS sobre un elemento del documento HTML y, de ese modo, poder cambiar el aspecto del mismo. Utilizando esto, podrá establecer los pares de valores que necesita para establecer un estilo sobre un elemento específico o sobre un bloque de elementos.

En definitiva, a todos los elementos con los que se compone una página web se les puede aplicar atributos de estilo para modificar su apariencia. Es por ello que debe entender completamente la sintaxis que se utiliza para asignar un valor a los atributos id y class.

De esta manera, creará páginas más profesionales y evitará perder tiempo cuando quiera encontrar un elemento específico, sobre todo si trabaja en un equipo de trabajo donde haya personas que tengan que modificar algo del documento que ha creado. Por ello, se recomienda que a los atributos id y class se les asigan valores que indiquen la función específica del elemento, en vez de, indicar algo sobre su aspecto o posición.

Vea a continuación una tabla donde puede ver los valores que se recomienda utilizar y cuáles no son recomendados:

Valores recomendados	Valores no recomendados
Importante	negrita
Titular	Arial11
Normal	verdanaGrande
menuSecundario	menulzquierdo
error	letraRosa

Aunque pueda parecer que elegir un valor correcto para los atributos id y class puede ser complicado, no es así. En realidad, es algo bastante sencillo, por ejemplo: si la apariencia del elemento cambia el valor de id o class deberá ser correspondiente al elemento, por lo tanto, debe evitar valores que reflejen su posición en el documento como pueden ser izquierdo, derecho, segundo, primero, inferior, superior, entre otros. También debe evitar valores que reflejen su color como textoVerde, subrayadoAmarillo, etc. O de tipo de letra Verdana10, Arial11, etc.

Para entender mejor la nomenclatura que se recomienda para los atributos, debe saber que tanto id como class tienen que cumplir algunas restricciones. Vea a continuación:

- Los valores solo podrán comenzar por letras, guion bajo o guion medio.
- Los caracteres siguientes podrán ser números, guiones medios, letras y guiones bajos.
- Debe saber que los navegadores son capaces de distinguir entre minúsculas y mayúsculas.
- Se pueden utilizar tanto acentos como la letra ñ, aunque esta práctica no se recomienda, puesto que, no puede estar seguros de que funcionen de manera correcta en todos los navegadores.

Ahora, que conoce las reglas sobre la nomenclatura que deben seguir los atributos de estilo, debe conocer cuáles atributos son los que más se utilizan. A continuación, va a ver las propiedades establecidas en las recomendaciones de las hojas de estilo. Las propiedades que encuentre serán parte de CSS2, las propiedades y valores añadidos en CSS 2.1 y las propiedades CSS3 que son compatibles con los navegadores de IE, Firefox y Google Chrome.

Las propiedades en las que va a profundizar forman parte de los siguientes elementos que puede encontrar en cualquier página web:

- Tipo de letra.
- Texto.
- Bordes.
- Margen exterior.
- Margen interior.
- Fondos.
- Listas.
- Tablas.
- Tamaño.
- Posicionamiento.
- Impresora.
- Interfaz de usuario.
- Seudoelementos.
- Seudoclases.
- Tipos.
- CSS3.

### 1.5.1. Descripción de los atributos de estilo

En este epígrafe va a profundizar en cada uno de los atributos de estilo mencionados anteriormente:

#### Tipo de letra

Propiedad	Descripción
<b>Font</b>	Propiedad compuesta
<b>Font-family</b>	Tipo de letra
<b>Font-size</b>	Tamaño de letra
<b>Font-style</b>	Cursiva
<b>Font-variant</b>	Versalitas
<b>Font-weight</b>	negrilla

- **Font-family.** Esta propiedad permite elegir el tipo de tipografía que quiera utilizar en su documento, es decir, la fuente del texto. Puede especificar todas las familias que desee, para ello, debe separarlas por espacios en blanco. Para que se muestre alguna de estas tipografías, el usuario deberá tenerla instalada en su equipo.
- **Font-size.** En HTML para poder darle tamaño a un texto solo tiene 5 valores disponibles. La razón de esto es que HTML intenta ser compatible en el mayor número de plataformas y dispositivos posibles. Sin embargo, gracias a CSS puede mantener esa compatibilidad dando más posibilidades para modificar los elementos de HTML. Por ello en CSS puede especificar el tamaño de letra de diversas maneras que verá a continuación:

Puede utilizar los tamaños predefinidos por CSS como son los siguientes: xx-small, x-small, small, medium, large, x-large y xx-large. Aunque estos tamaños predefinidos no especifican el tamaño de la fuente en sí, podrá saber que una fuente será mayor que otra sin entrar en los tamaños por píxeles y puntos que podrían llegar a confundir.

Además, otra de las cosas que puede hacer es indicar el tamaño de la letra relativo a su herencia. Los valores larger-smaller van a depender de la herencia para que el texto tenga un tamaño mayor o menor, de este modo los caracteres serán mayor o menor que el tamaño heredado. Si utiliza larger, el tamaño heredado de la letra siempre será mayor que el tamaño padre, en cambio, si utiliza smaller sucederá, al contrario, la letra será más pequeña que su tamaño heredado. Utilizar esta propiedad es muy útil a la hora de escribir textos muy extensos y querer resaltar ciertas partes del mismo.

Puede utilizar el seudoelemento :after, que vió anteriormente, para indicar que todos los elementos posteriores a un título se muestren con una tipografía más pequeña.

Lo más común es indicar de manera exacta el tamaño de la fuente. Esta es la manera más específica de indicar el tamaño de una fuente, además esta es la manera de hacer que los textos se muestren de la misma manera en todos los navegadores. Las medidas que se suelen utilizar para indicar el tamaño de una fuente son píxeles y puntos.

Además, otra opción que tiene es indicar el tamaño de una fuente con un porcentaje. Al utilizar los porcentajes para indicar el tamaño de una fuente, este porcentaje se aplica en base al valor original de la misma y, en caso de que no tenga un tamaño especificado, se tomará el tamaño por defecto de la misma.

- **Font-style.** Esta propiedad se utiliza para especificar el estilo de la fuente. Puede utilizar tres valores: normal, italic y oblique.
- **Font-weight.** Esta propiedad permite establecer el ancho de la tipografía. De esta manera, puede utilizar esta propiedad para convertir una letra a negrita o, incluso, darle un grosor más fino al original, así conseguir una tipografía más fina. Esta propiedad puede utilizar dos valores:

Valores predefinidos como: lighter, normal, bold y bolder.

Valores como: 100, 200, 300, 400, 500, 600, 700, 800, y 900. Estos números indican el grosor de la tipografía donde 400 es tamaño normal, mientras 700 equivale a bold.

## Texto

Propiedad	Descripción
Color	Color de texto
Direction	Dirección del texto
Letter-spacing	Espacio entre caracteres
Word-spacing	Espacio entre palabras
White-space	Saltos de línea
Vertical-align	Alineación vertical
Unicode-bidi	Dirección del texto
Text-indent	Sangría de primera línea
Text-transform	Mayúsculas y minúsculas
Text-decoration	Subrayado
Text-align	Alineación del texto
Line-height	Espaciado entre líneas

## Bordes

---

Propiedad	Descripción
<b>Border</b>	Establece los cuatro bordes a la vez
<b>Border-right</b>	Establece el borde derecho
<b>Border-left</b>	Establece el borde izquierdo
<b>Border-top</b>	Establece el borde superior
<b>Border-bottom</b>	Establece el borde inferior
<b>Border-color</b>	Establece el color de los bordes
<b>Border-style</b>	Establece el estilo de los bordes
<b>Border-width</b>	Establece el grosor de los bordes
<b>Border-right-color</b>	Establece el color del borde derecho
<b>Border-right-style</b>	Establece el estilo del borde derecho
<b>Border-right-width</b>	Establece el grosor del borde derecho
<b>Border-left-color</b>	Establece el color del borde izquierdo
<b>Border-left-style</b>	Establece el estilo del borde izquierdo
<b>Border-left-width</b>	Establece el grosor del borde izquierdo
<b>Border-top-color</b>	Establece el color del borde superior
<b>Border-top-style</b>	Establece el estilo del borde superior
<b>Border-top-width</b>	Establece el grosor del borde superior
<b>Border-bottom-color</b>	Establece el color del borde inferior
<b>Border-bottom-style</b>	Establece el estilo del borde inferior
<b>Border-bottom-width</b>	Establece el grosor del borde inferior

Las propiedades principales que utilizará para dar formato a los bordes de los diferentes elementos de una página son:

- **Border-style:** Con esta propiedad definirá el estilo que tendrá el borde. Por ejemplo, podrá ser sólido, rayado, punteado, etc.
- **Border-color:** Utilizará esta propiedad para modificar el color por defecto del borde. Podrá indicar un color distinto para cada borde, como ya ha visto en la tabla anterior. Para especificar los colores, puede utilizar el valor hexadecimal del propio color o el nombre por defecto.
- **Border-width:** Esta propiedad permite especificar el grosor de los bordes. Para ello, lo podrá hacer de dos formas:

Utilizando los valores predeterminados thin, medium y thick.

Utilizando un valor específico que le dé al valor del ancho, pudiendo especificar de manera individual cada uno de los bordes.

### Margen externo

Propiedad	Descripción
<b>Margin</b>	Establece los cuatro márgenes al mismo tiempo
<b>Margin-bottom</b>	Establece el margen inferior
<b>Margin-top</b>	Establece el margen superior

<b>Margin-left</b>	Establece el margen izquierdo
<b>Margin-right</b>	Establece el margen derecho

La propiedad margin se utiliza para especificar los márgenes de un elemento. Se entiende un margen como la distancia entre el borde externo de un elemento con el siguiente elemento. Al establecer la propiedad margin solo se le dará un valor para los cuatro lados del elemento. En cambio, también puede establecer un valor para cada uno de los márgenes utilizando las demás propiedades derivadas de margin.

### Margen interno

Propiedades	Descripción
<b>Padding</b>	Establece los cuatro márgenes internos al mismo tiempo
<b>Padding-bottom</b>	Establece el margen interno inferior
<b>Padding-top</b>	Establece el margen interno superior
<b>Padding-left</b>	Establece el margen interno izquierdo
<b>Padding-right</b>	Establece el margen interno derecho

La propiedad padding establece la distancia que hay desde el contenido del elemento hasta su borde, lo que entiende como la distancia interna del elemento. Además, también se puede utilizar como relleno, ya que, una utilidad muy común para esta propiedad es que el texto no se pegue a los bordes de una capa o tabla. Puede especificar esta propiedad para todos sus márgenes internos a la vez o para cada uno de ellos.

### Fondos o Modelo de Caja

Propiedades	Descripción
<b>Background</b>	Propiedad compuesta
<b>Background-repeat</b>	Repetición de la imagen de fondo
<b>Background-position</b>	Posición de la imagen de fondo
<b>Background-image</b>	Establece una imagen de fondo
<b>Background-color</b>	Establece un color de fondo
<b>Background-attachment</b>	Imagen adjunta

Estas propiedades pueden aplicarse a todos los elementos de la página y podrán funcionar en todos los navegadores que soporten CSS1. Estas propiedades son la base del Modelo de Caja y con ellas puede comprender cómo se construyen los elementos en el navegador. Vea a continuación:

- **Background-position:** Esta propiedad se utiliza para controlar la posición de la imagen de fondo en relación al margen izquierdo y al margen superior. De esta manera podrá ajustar las imágenes de fondo sin tener que utilizar para ello una tabla. Para realizar esta tarea, podrá utilizar los siguientes valores: right, left, bottom, top, center, top. Podrá indicar las distancias en porcentajes o en valores específicos.

- **Background-attachment:** Gracias a esta propiedad puede indicar si una imagen de fondo es estática o se puede desplazar mediante una barra de desplazamiento. Los valores que puede adquirir son: scroll y fixed.
- **Background-color:** Esta propiedad puede tener dos valores: color y transparent. Este último es el que tiene por defecto. Esta propiedad puede aplicársela a cualquier elemento que tenga un fondo.
- **Background-image:** Esta propiedad permite establecer una imagen de fondo para cualquier elemento de la página. El valor que le da a esta propiedad será la url donde se almacene la imagen que quiera establecer.
- **Background-repeat:** Esta propiedad permite que cuando se aplica una imagen más pequeña que el fondo en el que va a establecerla, ésta se repita mediante todo el espacio del fondo. Para ello, puede seleccionar si la imagen se comportará de esta manera o no. En caso de que quiera que se repita, debe especificar el eje. Los valores que puede utilizar para ello son: repeat-x, repeat-y o no-repeat.
- **Background:** Esta propiedad es la encargada de definir el fondo de un elemento de la página. Esta propiedad se suele utilizar para etiquetas que contienen otros elementos como pueden ser <table> o <body>. Sin embargo, también se puede utilizar para el resto de elementos de HTML. Lo bueno de esta propiedad es que permite establecer de fondo una imagen y poder posicionarla como deseé.

## Listas

Propiedades	Descripción
<b>List-style</b>	Propiedad compuesta
<b>List-style-type</b>	Establece el tipo de marcador
<b>List-style-position</b>	Establece la posición del marcador
<b>List-style-image</b>	Establece la imagen del marcador

## Tablas

Propiedades	Descripción
<b>Table-layout</b>	Ancho de tabla
<b>Empty-cells</b>	Establece bordes en las celdas vacías
<b>Border-collapse</b>	Modalidad de bordes
<b>Border-spacing</b>	Establece la separación entre celdas
<b>Caption-side</b>	Establece la posición de la leyenda

## Tamaño

Propiedades	Descripción
<b>Overflow</b>	Se usa cuando el contenido desborda al

	elemento
<b>Height</b>	Establece la altura
<b>Max-height</b>	Establece la altura máxima
<b>Min-height</b>	Establece la altura mínima
<b>Width</b>	Establece la anchura
<b>Max-width</b>	Establece la anchura máxima
<b>Min-width</b>	Establece la anchura mínima

## Posicionamiento

Propiedades	Descripción
<b>Position</b>	Modo de posicionamiento
<b>Float</b>	Posicionamiento flotante
<b>Clear</b>	No se permiten elementos flotantes
<b>Bottom</b>	Posición del borde inferior del elemento
<b>Top</b>	Posición del borde superior del elemento
<b>Right</b>	Posición del borde derecho del elemento
<b>Left</b>	Posición del borde izquierdo del elemento
<b>Display</b>	Tipo de caja
<b>Clip</b>	Recorte del elemento
<b>Visibility</b>	Visibilidad del elemento
<b>z-index</b>	Apilamiento del elemento

## Impresora

Propiedades	Descripción
<b>Windows</b>	Cantidad de líneas al principio de la página
<b>Orphans</b>	Cantidad de líneas al final de la página
<b>Page-break-before</b>	Salto de página anterior a
<b>Page-break-after</b>	Salto de página posterior a
<b>Page-break-inside</b>	Salto de página dentro de

## Interfaz de usuario

Propiedades	Descripción
<b>Outline</b>	Establece los cuatro bordes a la vez
<b>Outline-style</b>	Establece el estilo de los bordes
<b>Outline-color</b>	Establece el color de los bordes
<b>Outline-width</b>	Establece el grosor de los bordes
<b>Cursor</b>	Establece el tipo de cursor

## Seudo-elementos

Propiedades	Descripción
<b>Quotes</b>	Comillas

<b>Content</b>	Contenido generado
<b>Counter-increment</b>	Incremento del contador
<b>Counter-reset</b>	Puesta a cero del contador
<b>:first-letter</b>	Primera letra
<b>First-line</b>	Primera línea
<b>:before</b>	Insertar contenido antes de un elemento
<b>:after</b>	Insertar contenido después de un elemento

## Seudo-clases

Propiedades	Descripción
<b>:visited</b>	Enlaces visitados
<b>:link</b>	Enlaces no visitados
<b>:Lang</b>	Idioma
<b>:active</b>	Al pulsar un elemento
<b>:first-child</b>	Primer elemento hijo
<b>:hover</b>	Al pasar el ratón por encima de un elemento
<b>:focus</b>	Cuando el elemento tiene foco

## CSS3

Propiedades	Descripción
<b>Border-radius</b>	Bordes redondeados
<b>Border-top-left-radius</b>	Borde superior izquierdo redondeado
<b>Border-bottom-left-radius</b>	Borde inferior izquierdo redondeado
<b>Border-top-right-radius</b>	Borde superior derecho redondeado
<b>Border-bottom-right-radius</b>	Borde inferior derecho redondeado

### 1.5.2. Utilización de los atributos de estilo

En este epígrafe, va a ver cómo puede utilizar cada uno de los atributos de estilo. Los valores de las propiedades que se han añadido en la recomendación CSS 2.1 estarán indicados con un símbolo más, y los que se hayan eliminado estarán indicados con un símbolo menos.

#### Tipografía

Propiedades	Valores
<b>Font</b>	Font-size, Font-style, Font-variant, Font-weight, Font-family, line-height, caption, menú, icon, message-box, small-caption, status-bar
<b>Font-family</b>	Nombre de fuente y familia genérica
<b>Font-size</b>	Tamaño absoluto, tamaño relativo, distancia, porcentaje
<b>Font-style</b>	Italic, normal, oblique
<b>Font-variant</b>	Normal y small-caps
<b>Font-weight</b>	Lighter, normal, bold, bolder, 100, 200, 300, 400, 500, 600, 700, 800, 900

## Texto

Propiedades	Valores
<b>Word-spacing</b>	Normal, distancia
<b>White-space</b>	Normal, pre, pre-line, wrap, pre-wrap
<b>Color</b>	Color
<b>Vertical-align</b>	Bottom, baseline, sub, middle, super, top, text-bottom, text-top, porcentaje, distancia
<b>Unicode-bidi</b>	Embed, normal, bidi-override
<b>Text-transform</b>	Uppercase, lowercase, none, capitalize
<b>Text-indent</b>	Distancia, porcentaje
<b>Direction</b>	Ltr, rtl
<b>Text-decoration</b>	Underline, overline, line-through, blink
<b>Line-height</b>	Número, normal, distancia, porcentaje
<b>Text-align</b>	Right, left, justify, center
<b>Letter-spacing</b>	Distancia, normal

A continuación, va a ver algunos ejemplos de los atributos de estilo que ha visto:

- **Primer ejemplo:**

### HTML

```
<!DOCTYPE html>

<html>

<head>

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

    <title>Ejemplo 1</title>

    <link rel="stylesheet" type="text/css" href="css/estilo.css"/>

</head>

<body>

    <div class="general">

        <div class="titulo">Ejemplo 1</div>

        <div class="menu">
```

```
<a href="main.html">Inicio</a>

<a href="imagen.html">Imagen de perfil</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<div class="contenido">

    <p><span id="primeraLetra">S</span>oy un párrafo</p>

</div>

    <div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

</div>

</body>

</html>
```

## CSS

```
.contenido{

    font-family: Verdana, Geneva, Tahoma, sans-serif;

    position: relative;

    padding-top: 25px;

    padding-right: 20px;

    padding-bottom: 20px;

    padding-left: 20px;

    border-style: dotted;

    border-color: blueviolet;

    border-width: 3px;
```

```
text-align: justify;  
min-height: 150px;  
}
```

En este caso, el texto de la web tendría la tipografía Verdana.

- **Segundo ejemplo:**

#### HTML

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  
    <title>Ejemplo 1</title>  
  
    <link rel="stylesheet" type="text/css" href="css/estilo.css"/>  
  
</head>  
  
<body>  
  
    <div class="general">  
  
        <div class="titulo">Ejemplo 1</div>  
  
        <div class="menu">  
  
            <a href="main.html">Inicio</a>  
  
            <a href="imagen.html">Imagen de perfil</a>  
  
            <a href="cv.html">CV</a>  
  
            <a href="contacto.html">Contacto</a>  
  
        </div>  
  
        <div class="contenido">
```

```
<p><span id="primeraLetra">S</span>oy un párrafo</p>

</div>

<div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

</div>

</body>

</html>
```

## CSS

```
.contenido{

    font-family: Verdana, Geneva, Tahoma, sans-serif;

    font-size: 15px;

    position: relative;

    padding-top: 25px;

    padding-right: 20px;

    padding-bottom: 20px;

    padding-left: 20px;

    border-style: dotted;

    border-color: blueviolet;

    border-width: 3px;

    text-align: justify;

    min-height: 150px;

}
```

En esto otro ejemplo, el tamaño de letra será de 15 píxeles.

- **Tercer ejemplo:**

## HTML

```
<!DOCTYPE html>

<html>

<head>

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

<title>Ejemplo 1</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css"/>

</head>

<body>

<div class="general">

<div class="titulo">Ejemplo 1</div>

<div class="menu">

<a href="main.html">Inicio</a>

<a href="imagen.html">Imagen de perfil</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<div class="contenido">

<p><span id="primeraLetra">S</span>oy un párrafo</p>

</div>

<div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

</div>

</body>
```

```
</html>
```

## CSS

```
.contenido{  
    font-family: Verdana, Geneva, Tahoma, sans-serif;  
    font-size: 15px;  
    font-style: oblique;  
    position: relative;  
    padding-top: 25px;  
    padding-right: 20px;  
    padding-bottom: 20px;  
    padding-left: 20px;  
    border-style: dotted;  
    border-color: blueviolet;  
    border-width: 3px;  
    text-align: justify;  
    min-height: 150px;  
}
```

En este ejemplo, el texto que hay en la capa contenido será oblicuo.

- **Cuarto ejemplo:**

## HTML

```
<!DOCTYPE html>  
  
<html>  
    <head>
```

---

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

<title>Ejemplo 1</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css"/>

</head>

<body>

<div class="general">

    <div class="titulo">Ejemplo 1</div>

    <div class="menu">

        <a href="main.html">Inicio</a>

        <a href="imagen.html">Imagen de perfil</a>

        <a href="cv.html">CV</a>

        <a href="contacto.html">Contacto</a>

    </div>

    <div class="contenido">

        <p><span id="primeraLetra">S</span>oy un párrafo</p>

    </div>

    <div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

    </div>

</body>

</html>
```

## CSS

```
.contenido{

    font-family: Verdana, Geneva, Tahoma, sans-serif;
```

```
font-variant: small-caps;  
font-size: 15px;  
font-style: oblique;  
position: relative;  
padding-top: 25px;  
padding-right: 20px;  
padding-bottom: 20px;  
padding-left: 20px;  
border-style: dotted;  
border-color: blueviolet;  
border-width: 3px;  
text-align: justify;  
min-height: 150px;  
}
```

En este ejemplo, las letras de la capa contenido estarán todas en mayúsculas.

- **Quinto ejemplo:**

#### HTML

```
<!DOCTYPE html>  
<html>  
<head>  
  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  
<title>Ejemplo 1</title>  
  
<link rel="stylesheet" type="text/css" href="css/estilo.css"/>
```

---

```
</head>

<body>

<div class="general">

    <div class="titulo">Ejemplo 1</div>

    <div class="menu">

        <a href="main.html">Inicio</a>

        <a href="imagen.html">Imagen de perfil</a>

        <a href="cv.html">CV</a>

        <a href="contacto.html">Contacto</a>

    </div>

    <div class="contenido">

        <p><span id="primeraLetra">S</span>oy un párrafo</p>

    </div>

    <div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

</div>

</body>

</html>
```

## CSS

```
.contenido{

    font-family: Verdana, Geneva, Tahoma, sans-serif;

    font-weight: bolder;

    position: relative;
```

```
padding-top: 25px;  
padding-right: 20px;  
padding-bottom: 20px;  
padding-left: 20px;  
border-style: dotted;  
border-color: blueviolet;  
border-width: 3px;  
text-align: justify;  
min-height: 150px;  
}
```

En este ejemplo, podrá ver que las letras son más gruesas aún que la negrita.

- **Sexto ejemplo:**

#### HTML

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  
  <title>Ejemplo 1</title>  
  
  <link rel="stylesheet" type="text/css" href="css/estilo.css"/>  
  
</head>  
  
<body>  
  
  <div class="general">  
  
    <div class="titulo">Ejemplo 1</div>
```

```
<div class="menu">  
    <a href="main.html">Inicio</a>  
    <a href="imagen.html">Imagen de perfil</a>  
    <a href="cv.html">CV</a>  
    <a href="contacto.html">Contacto</a>  
</div>  
  
<div class="contenido">  
    <p><span id="primeraLetra">S</span>oy un párrafo</p>  
</div>  
  
<div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>  
</div>  
</body>  
</html>
```

## CSS

```
.contenido{  
    font-family: Verdana, Geneva, Tahoma, sans-serif;  
    font-weight: normal;  
    position: relative;  
    padding-top: 25px;  
    padding-right: 20px;  
    padding-bottom: 20px;  
    padding-left: 20px;  
    border-style: dotted;
```

```
border-color: blueviolet;  
  
border-width: 3px;  
  
text-align: justify;  
  
min-height: 150px;  
  
}
```

En este ejemplo, puede ver que el grosor de la fuente es normal.

- **Séptimo ejemplo:**

#### HTML

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  
  <title>Ejemplo 1</title>  
  
  <link rel="stylesheet" type="text/css" href="css/estilo.css"/>  
  
</head>  
  
<body>  
  
  <div class="general">  
  
    <div class="titulo">Ejemplo 1</div>  
  
    <div class="menu">  
  
      <a href="main.html">Inicio</a>  
  
      <a href="imagen.html">Imagen de perfil</a>  
  
      <a href="cv.html">CV</a>
```

```
<a href="contacto.html">Contacto</a>

</div>

<div class="contenido">

<p><span id="primeraLetra">S</span>oy un párrafo</p>

</div>

<div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

</div>

</body>

</html>
```

### CSS

```
.contenido{

font-family: Verdana, Geneva, Tahoma, sans-serif;

font-weight: inherit;

position: relative;

padding-top: 25px;

padding-right: 20px;

padding-bottom: 20px;

padding-left: 20px;

border-style: dotted;

border-color: blueviolet;

border-width: 3px;

text-align: justify;

min-height: 150px;
```

}

En este ejemplo, el grosor de la letra será heredado.

- **Octavo ejemplo:**

#### HTML

```
<!DOCTYPE html>

<html>

<head>

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

<title>Ejemplo 1</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css"/>

</head>

<body>

<div class="general">

<div class="titulo">Ejemplo 1</div>

<div class="menu">

<a href="main.html">Inicio</a>

<a href="imagen.html">Imagen de perfil</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<div class="contenido">

<p><span id="primeraLetra">S</span>oy un párrafo</p>
```

```
</div>

<div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

</div>

</body>

</html>
```

## CSS

```
.contenido{

    font-family: Verdana, Geneva, Tahoma, sans-serif;

    font-weight: lighter;

    position: relative;

    padding-top: 25px;

    padding-right: 20px;

    padding-bottom: 20px;

    padding-left: 20px;

    border-style: dotted;

    border-color: blueviolet;

    border-width: 3px;

    text-align: justify;

    min-height: 150px;

}
```

En este último ejemplo, obtendría un grosor de letra más fino de lo normal.

## Texto

---

Ahora, va a ver algunos de los atributos que se aplican al **texto de un documento web**.

- **Primer ejemplo:**

#### HTML

```
<!DOCTYPE html>

<html>

<head>

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

    <title>Ejemplo 1</title>

    <link rel="stylesheet" type="text/css" href="css/estilo.css"/>

</head>

<body>

    <div class="general">

        <div class="titulo">Ejemplo 1</div>

        <div class="menu">

            <a href="main.html">Inicio</a>

            <a href="imagen.html">Imagen de perfil</a>

            <a href="cv.html">CV</a>

            <a href="contacto.html">Contacto</a>

        </div>

        <div class="contenido">

            <p><span id="primeraLetra">S</span>oy un párrafo</p>

        </div>

        <div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>
```

```
</div>
```

```
</body>
```

```
</html>
```

## CSS

```
.contenido{  
    color:deeppink;  
    position: relative;  
    padding-top: 25px;  
    padding-right: 20px;  
    padding-bottom: 20px;  
    padding-left: 20px;  
    border-style: dotted;  
    border-color: blueviolet;  
    border-width: 3px;  
    text-align: justify;  
    min-height: 150px;  
}
```

En este ejemplo, puede ver cómo ha establecido un estilo a la letra para que sea de color rosa.

- **Segundo ejemplo:**

## HTML

```
<!DOCTYPE html>  
  
<html>  
    <head>
```

---

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

<title>Ejemplo 1</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css"/>

</head>

<body>

<div class="general">

    <div class="titulo">Ejemplo 1</div>

    <div class="menu">

        <a href="main.html">Inicio</a>

        <a href="imagen.html">Imagen de perfil</a>

        <a href="cv.html">CV</a>

        <a href="contacto.html">Contacto</a>

    </div>

    <div class="contenido">

        <p><span id="primeraLetra">S</span>oy un párrafo</p>

    </div>

    <div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

    </div>

</body>

</html>
```

## CSS

```
.contenido{  
    line-height: 5%;  
}
```

```
position: relative;  
  
padding-top: 25px;  
  
padding-right: 20px;  
  
padding-bottom: 20px;  
  
padding-left: 20px;  
  
border-style: dotted;  
  
border-color: blueviolet;  
  
border-width: 3px;  
  
text-align: justify;  
  
min-height: 150px;  
  
}
```

En este ejemplo, ha aplicado una altura al texto del 5%.

- **Tercer ejemplo:**

#### HTML

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  
  <title>Ejemplo 1</title>  
  
  <link rel="stylesheet" type="text/css" href="css/estilo.css"/>  
  
</head>  
  
<body>  
  
  <div class="general">
```

```
<div class="titulo">Ejemplo 1</div>

<div class="menu">

    <a href="main.html">Inicio</a>

    <a href="imagen.html">Imagen de perfil</a>

    <a href="cv.html">CV</a>

    <a href="contacto.html">Contacto</a>

</div>

<div class="contenido">

    <p><span id="primeraLetra">S</span>oy un párrafo</p>

</div>

<div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

</div>

</body>

</html>
```

## CSS

```
.contenido{

    text-align: justify;

    position: relative;

    padding-top: 25px;

    padding-right: 20px;

    padding-bottom: 20px;

    padding-left: 20px;

    border-style: dotted;
```

```
border-color: blueviolet;  
  
border-width: 3px;  
  
text-align: justify;  
  
min-height: 150px;  
  
}
```

En este ejemplo, ha hecho que el texto quede con una alineación justificada.

- **Cuarto ejemplo:**

#### HTML

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  
  <title>Ejemplo 1</title>  
  
  <link rel="stylesheet" type="text/css" href="css/estilo.css"/>  
  
</head>  
  
<body>  
  
  <div class="general">  
  
    <div class="titulo">Ejemplo 1</div>  
  
    <div class="menu">  
  
      <a href="main.html">Inicio</a>  
  
      <a href="imagen.html">Imagen de perfil</a>  
  
      <a href="cv.html">CV</a>  
  
      <a href="contacto.html">Contacto</a>  
    </div>  
  </div>  
</body>
```

```
</div>

<div class="contenido">

<p><span id="primeraLetra">S</span>oy un párrafo</p>

</div>

<div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

</div>

</body>

</html>
```

## CSS

```
.contenido{

text-decoration: wavy;

position: relative;

padding-top: 25px;

padding-right: 20px;

padding-bottom: 20px;

padding-left: 20px;

border-style: dotted;

border-color: blueviolet;

border-width: 3px;

text-align: justify;

min-height: 150px;

}
```

En este ejemplo, ha aplicado una decoración del texto ondulada.

- **Quinto ejemplo:**

#### HTML

```
<!DOCTYPE html>

<html>

<head>

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

<title>Ejemplo 1</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css"/>

</head>

<body>

<div class="general">

<div class="titulo">Ejemplo 1</div>

<div class="menu">

<a href="main.html">Inicio</a>

<a href="imagen.html">Imagen de perfil</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<div class="contenido">

<p><span id="primeraLetra">S</span>oy un párrafo</p>

</div>

<div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

</div>
```

```
</body>
```

```
</html>
```

## CSS

```
.contenido{  
    text-indent: 1%;  
    position: relative;  
    padding-top: 25px;  
    padding-right: 20px;  
    padding-bottom: 20px;  
    padding-left: 20px;  
    border-style: dotted;  
    border-color: blueviolet;  
    border-width: 3px;  
    text-align: justify;  
    min-height: 150px;  
}
```

En este ejemplo, ha establecido para el texto una tabulación del 1%.

- **Sexto ejemplo:**

## HTML

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>Ejemplo 1</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css"/>

</head>

<body>

<div class="general">

    <div class="titulo">Ejemplo 1</div>

    <div class="menu">

        <a href="main.html">Inicio</a>

        <a href="imagen.html">Imagen de perfil</a>

        <a href="cv.html">CV</a>

        <a href="contacto.html">Contacto</a>

    </div>

    <div class="contenido">

        <p><span id="primeraLetra">S</span>oy un párrafo</p>

    </div>

    <div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

</div>

</body>

</html>
```

## CSS

```
.contenido{

    text-transform: lowercase;

    position: relative;
```

```
padding-top: 25px;  
padding-right: 20px;  
padding-bottom: 20px;  
padding-left: 20px;  
border-style: dotted;  
border-color: blueviolet;  
border-width: 3px;  
text-align: justify;  
min-height: 150px;  
}
```

En este ejemplo, ha puesto todas las letras en minúsculas.

- **Séptimo ejemplo:**

#### HTML

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  
  <title>Ejemplo 1</title>  
  
  <link rel="stylesheet" type="text/css" href="css/estilo.css"/>  
  
</head>  
  
<body>  
  
  <div class="general">  
  
    <div class="titulo">Ejemplo 1</div>
```

```
<div class="menu">  
    <a href="main.html">Inicio</a>  
    <a href="imagen.html">Imagen de perfil</a>  
    <a href="cv.html">CV</a>  
    <a href="contacto.html">Contacto</a>  
</div>  
  
<div class="contenido">  
    <p><span id="primeraLetra">S</span>oy un párrafo</p>  
</div>  
  
<div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>  
</div>  
</body>  
</html>
```

## CSS

```
.contenido{  
    unicode-bidi: embed;  
    position: relative;  
    padding-top: 25px;  
    padding-right: 20px;  
    padding-bottom: 20px;  
    padding-left: 20px;  
    border-style: dotted;  
    border-color: blueviolet;
```

```
border-width: 3px;  
  
text-align: justify;  
  
min-height: 150px;  
  
}
```

En este ejemplo, ha establecido que el texto tenga un nivel adicional, por lo tanto, puede quedar embebido por otro texto. Utiliza esta propiedad cuando quiera añadir varios idiomas en una misma página web.

- **Octavo ejemplo:**

#### HTML

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  
    <title>Ejemplo 1</title>  
  
    <link rel="stylesheet" type="text/css" href="css/estilo.css"/>  
  
</head>  
  
<body>  
  
    <div class="general">  
  
        <div class="titulo">Ejemplo 1</div>  
  
        <div class="menu">  
  
            <a href="main.html">Inicio</a>  
  
            <a href="imagen.html">Imagen de perfil</a>  
  
            <a href="cv.html">CV</a>  
  
            <a href="contacto.html">Contacto</a>  
    
```

```
</div>

<div class="contenido">

    <p><span id="primeraLetra">S</span>oy un párrafo</p>

</div>

    <div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

</div>

</body>

</html>
```

## CSS

```
.contenido{

    vertical-align: 50%;

    position: relative;

    padding-top: 25px;

    padding-right: 20px;

    padding-bottom: 20px;

    padding-left: 20px;

    border-style: dotted;

    border-color: blueviolet;

    border-width: 3px;

    text-align: justify;

    min-height: 150px;

}
```

En este ejemplo, puede ver que ha alineado verticalmente el texto al 50% tomando como referencia el espacio de la esquina superior izquierda.

- **Noveno ejemplo:**

#### HTML

```
<!DOCTYPE html>

<html>

<head>

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

    <title>Ejemplo 1</title>

    <link rel="stylesheet" type="text/css" href="css/estilo.css"/>

</head>

<body>

    <div class="general">

        <div class="titulo">Ejemplo 1</div>

        <div class="menu">

            <a href="main.html">Inicio</a>

            <a href="imagen.html">Imagen de perfil</a>

            <a href="cv.html">CV</a>

            <a href="contacto.html">Contacto</a>

        </div>

        <div class="contenido">

            <p><span id="primeraLetra">S</span>oy un párrafo</p>

        </div>

    </div>
```

```
<div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

</div>

</body>

</html>
```

### CSS

```
.contenido{

white-space: nowrap;

position: relative;

padding-top: 25px;

padding-right: 20px;

padding-bottom: 20px;

padding-left: 20px;

border-style: dotted;

border-color: blueviolet;

border-width: 3px;

text-align: justify;

min-height: 150px;

}
```

En este ejemplo, el texto va a quedar en una sola línea donde no tendrá espacios en los saltos de línea.

- **Décimo ejemplo:**

### HTML

```
<!DOCTYPE html>
```

---

```
<html>

<head>

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

<title>Ejemplo 1</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css"/>

</head>

<body>

<div class="general">

<div class="titulo">Ejemplo 1</div>

<div class="menu">

<a href="main.html">Inicio</a>

<a href="imagen.html">Imagen de perfil</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<div class="contenido">

<p><span id="primeraLetra">S</span>oy un párrafo</p>

</div>

<div class="piePagina"><p><a href="mapasitio.html" target="_blank">Mapa del sitio</a></p></div>

</div>

</body>

</html>
```

## CSS

---

```
.contenido{  
    word-spacing: 30px;  
  
    position: relative;  
  
    padding-top: 25px;  
  
    padding-right: 20px;  
  
    padding-bottom: 20px;  
  
    padding-left: 20px;  
  
    border-style: dotted;  
  
    border-color: blueviolet;  
  
    border-width: 3px;  
  
    text-align: justify;  
  
    min-height: 150px;  
}
```

En este último ejemplo, puede ver cómo le ha dado un espacio de 30 píxeles entre letras al texto.

## 1.6. Creación de ficheros de estilo

Ahora, que ha llegado al último punto del tema, ya conoce bastante bien CSS como para comenzar a realizar sus primeros documentos y utilizarlos para añadir estilos en su página web.

Aunque es cierto que crear un documento CSS es muy sencillo, antes, debe tener una idea de qué estilos va a aplicar en su web. Para ello, lo más aconsejable es tomar un lápiz y un papel y empezar a diseñar el estilo que quiera que tenga su web.

Lo mejor es ir apuntando las ideas principales sobre qué colores va a utilizar como base, qué tipos de letra tendrá la web, el tamaño de la letra, el color de la misma, reunir las imágenes necesarias para la web y pensar el tamaño de los elementos y sus dimensiones que ocuparán en la página. Cuando haya terminado de reunir estos conceptos y este de acuerdo con ellos, será hora de comenzar a aplicarlos en un entorno real.

A continuación, va a aprender cómo crear un documento CSS paso por paso. Debe saber que un documento CSS es tan solo la hoja de estilo que será la encargada de aportarle estilo al documento HTML. Siempre que cree una hoja de estilos, lo haga de manera externa, puesto que, es lo más recomendado y profesional.

Los epígrafes que verá a continuación serán los pasos que da para crear su hoja de estilos, los cuales serán los siguientes:

- Definir lo que es un fichero de estilo.
- Creación del fichero.
- Adaptación del fichero a diferentes páginas web.

### **1.6.1. Definición de los ficheros de estilo**

En el año 1970, fue cuando aparecieron por primera vez las hojas de estilo, debido a que se necesitaba una manera de aplicar estilos a los documentos web. Sin embargo, no fue hasta que empezó el uso de Internet de manera mundial cuando, definitivamente, se asentó el uso de las hojas de estilo. En cambio, aún era complicado que los estilos tuvieran la misma apariencia en todos los navegadores, ya que, no había ningún estándar para ello.

Definitivamente, fue el W3C quien hizo una propuesta seria para comenzar a crear los estándares web que conoce hoy día y, en esos estándares estaba también la creación de un lenguaje para las hojas de estilo y para los documentos HTML. Después de varios años, se creó en 1996 el CSS como se conoce en la actualidad, con su versión CSS1.

Por otro lado, en 1998 se publicó la versión CSS2, sin embargo, los navegadores actuales funcionan con CSS2.1 y CSS3.

Cuando diseñe una página web siempre debe seguir las reglas del W3C para que su página pueda visualizarse en todos los navegadores. Por ello, es importante saber el soporte que tienen de CSS los navegadores más conocidos.

Lo más importante para un diseñador es conocer el motor que tiene cada navegador. El motor consiste en muchos componentes, entre ellos, la parte que está encargada de interpretar el código HTML y CSS.

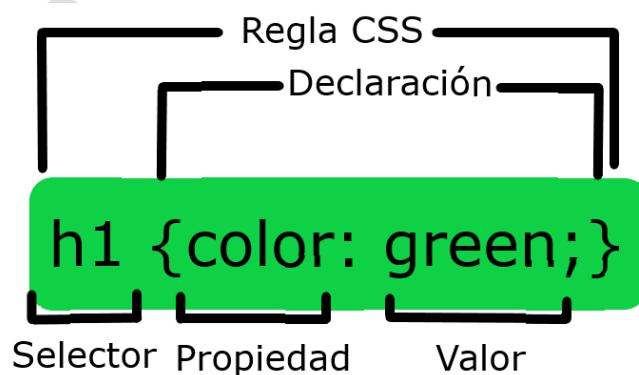
Vea a continuación una tabla donde se muestran los navegadores más utilizados por los usuarios y el motor que tienen junto con las reglas CSS que soportan:

Navegador	Motor	CSS1	CSS2.1	CSS3
Opera	Presto	Completo desde la versión 1.0	Completo	Todas las seudoclases, selectores y diversas propiedades.
Safari	WebKit	Completo desde la versión 85.	Completo	Todas las seudoclases, selectores y diversas propiedades.
Firefox	Gecko	Completo desde la versión 1.0	Completo	Todas las seudoclases, selectores y diversas propiedades.
Google Chrome	WebKit	Completo desde la versión 85.	Completo	Todas las seudoclases, selectores y diversas propiedades.

Como puede comprobar es muy importante conocer las especificaciones del W3C, puesto que, el W3C trabaja constantemente en la creación de nuevos estándares para sacar nuevas versiones mejoradas de los lenguajes CSS y HTML. Por ejemplo, en las especificaciones de la versión CSS3 puede encontrar muchos cambios nuevos. Si compara CSS3 con sus versiones anteriores, esta es mucho más compleja y amplia. Si quiera estar al tanto de las actualizaciones que realiza el W3C, puede visitar su [blog](#) o [web](#) regularmente.

Al conocer las especificaciones actualizadas del W3C, podrá crear documentos que cumplan las normas actuales y, por lo tanto, sean estables, de manera que puedan ser visualizado en la mayoría de navegadores que utilizan los usuarios actualmente.

Como CSS utiliza algunos términos interesantes que ha dado anteriormente, va a recordarlos para poder reconocer mejor los aspectos del código. Como ya vió, CSS utiliza determinados términos para nombrar las diferentes partes del código. Va a ver la siguiente imagen para recordarlo:



Como puede ver en la imagen, estos son los términos que se utilizan para las diferentes partes que forman una regla básica en CSS.

A continuación, va a ver una definición sobre cada uno de los términos que ha visto:

- Regla: Las reglas son los estilos que componen una hoja de estilo. Cada una de las reglas se cierran entre llaves.
- Selector: Los selectores se utilizan para seleccionar un elemento en HTML al que aplicarle un estilo específico.
- Declaración: La declaración es la encargada de indicar cuáles de los estilos son los que se van a aplicar a los elementos. Las declaraciones pueden estar compuestas por una única propiedad o por múltiples propiedades.
- Propiedad: Esta es la parte que se modifica en los elementos seleccionados, por ejemplo, una propiedad es el tamaño de letra, el color, la fuente, entre otras.
- Valor: Este término se encarga de establecer un nuevo valor en la característica que modificará el aspecto de un elemento.



TOME NOTA

Debe saber que en un documento de estilo podrá haber tantas reglas CSS como sea necesario y, a su vez, las reglas pueden contener todos los selectores que sean necesarios con tantas propiedades y valores como sean necesarias.

### 1.6.2. Creación de ficheros de estilo genéricos

Llegados a este punto, sabe que puede aplicar estilos a un documento HTML de diversas maneras. Sin embargo, la más recomendada es creando un archivo CSS externo y enlazándolo con el archivo HTML deseado.

Al crear un documento CSS externo, también podrá utilizarlo de manera simultánea en todas las páginas web que desee y, además, contara con la ventaja de que, si necesita modificar alguna propiedad, se modificará para todas las páginas donde se haya usado el documento CSS, lo que ahorrará mucho tiempo. Debe seguir los siguientes pasos para crear su archivo CSS:

- Primero, abrir un nuevo documento en un editor de texto.
- Segundo, le pondrá el nombre que desee y lo guarda en la ubicación que desee.
- Por último, lo guarda con la extensión .css. Debe asegurarse de utilizar esa extensión, ya que, si no estará creando un documento de texto que no funcionará con HTML.

Para comenzar a escribir en su archivo CSS, tan solo debe escribir las reglas CSS que necesita. En los archivos CSS no pasa como en HTML que debe establecer una declaración de apertura del archivo y establecer unas etiquetas.

La razón por la que en los archivos CSS solo tenga que escribir los estilos es que, al ser un archivo con extensión .css, los navegadores lo interpretan como un archivo de formato, por ello, todas las reglas CSS tendrán el mismo significado que las etiquetas HTML.

Debe asegurarse de añadir la línea de código correspondiente en el archivo HTML para que los estilos que haya establecido en la hoja de estilo se apliquen a los elementos. Además, debe recordar que esta línea de código siempre se añadirá en el encabezado del documento HTML, entre las etiquetas <head> y </head>, la sintaxis será la siguiente:

```
<link rel="stylesheet" type="text/css" href="estilos.css"></link>
```

Donde estilos.css se modificará con el nombre que le haya puesto a su archivo CSS.

En resumen, debe seguir dos pasos fundamentales para crear un documento de estilo:

- Primero, crear la hoja de estilos mediante un documento de texto con la extensión .css.
- Segundo, crear el enlace en el documento HTML hacia el documento CSS.

Cuando haya terminado estos dos pasos, antes de empezar a escribir en el documento, lo primero que haga es tener en cuenta las recomendaciones generales que haya de CSS.

La primera recomendación será sobre los atributos class e id y cómo utilizarlos de manera correcta dentro de su documento CSS.

Utilizará el atributo id para identificar cada elemento que haya en el documento HTML, es por ello que los identificadores deben ser únicos en todo el documento. Por lo que, dos elementos diferentes en una misma página no podrán tener un id idéntico.

Por otro lado, utilizará el atributo class para indicar la clase o clases a las que pertenece un elemento. Una clase puede tener diversos elementos y un elemento puede pertenecer a diferentes clases. Todo ello lo indicar mediante espacios en blanco.

Es cierto que el atributo id tiene otros propósitos, no solo emplearlos en CSS, sin embargo, CSS los emplea básicamente para seleccionar los elementos donde aplicar los estilos.

Vea un ejemplo donde hay dos listas creadas por un mismo elemento HTML y donde sus atributos id diferencian ambos elementos de manera correcta:

```
<ul id="menú">  
  </ul>  
  
<ul id="vínculos">  
  </ul>
```

El uso de los atributos id y class son una de las recomendaciones para el diseño de documentos CSS y XHTML y es que, el uso de estos dos atributos debe estar relacionado con la mejora de la semántica del documento, o lo que es lo mismo, con darle un significado a cada elemento que compone la página.

Es por ello que la siguiente recomendación que verá está relacionada con darle los valores correctos a los atributos class e id, lo cuales deberán indicar la función de cada uno de los elementos y no deberá relacionarse con su aspecto. Estas recomendaciones ya las vió anteriormente, pero va a recordarlas:

Valores recomendados	Valores no recomendados
Importante	negrita
Titular	Arial11
Normal	verdanaGrande
menuSecundario	menulzquierdo
error	letraRosa

Aunque pueda parecer que elegir un valor correcto para los atributos id y class puede ser complicado, no es así. En realidad, es algo bastante sencillo, por ejemplo: si la apariencia del elemento cambia el valor de id o class deberá ser correspondiente al elemento, por lo tanto, debe evitar valores que reflejen su posición en el documento como pueden ser izquierdo, derecho, segundo, primero, inferior, superior, entre otros. También debe evitar valores que reflejen su color como textoVerde, subrayadoAmarillo, etc. O de tipo de letra Verdana10, Arial11, etc.

Para entender mejor la nomenclatura que se recomienda para los atributos, debe saber que tanto id como class tienen que cumplir algunas restricciones. Vea a continuación:

- Los valores solo podrán comenzar por letras, guion bajo o guion medio.
- Los caracteres siguientes podrán ser números, guiones medios, letras y guiones bajos.
- Debe saber que los navegadores son capaces de distinguir entre minúsculas y mayúsculas.
- Se pueden utilizar tanto acentos como la letra ñ, aunque esta práctica no se recomienda, puesto que, no puede estar seguros de que funcionen de manera correcta en todos los navegadores.

Vea otras recomendaciones que también debe tener en cuenta cuando use los atributos id y class:

- Utilizar de manera excesiva la etiqueta <div> y el atributo class es un error que cometen muchos principiantes al diseñar una página web.

Vea un ejemplo bastante común donde puede verse esta práctica:

```
<div id="menu">

<ul class="menu">

    <li class="elemento"><span class="textoElemento"></span></li>

    <li class="elemento"><span class="textoElemento"></span></li>
```

```
<li class="elemento"><span class="textoElemento"></span></li>  
  
<li class="elemento"><span class="textoElemento"></span></li>  
  
</ul>  
  
</div>
```

Gracias a los selectores CSS, puede prescindir de la mayoría de las etiquetas `<div>` y de los atributos `id` y `class`. Crear páginas con un uso excesivo de esos elementos no va a mejorar la semántica del documento y lo único que conseguirá es dificultar el código HTML.

El hecho de que pueda establecer los estilos en un archivo externo al HTML, da la posibilidad de mantener los estilos en un archivo exclusivamente para los mismos y, además, mantener el código HTML estructurado y ordenado de manera que facilite su lectura y modificación.

Vea a continuación las recomendaciones que existen para que pueda estructurar de una mejor manera un documento CSS. Cuando tiene archivos CSS complejos, sus reglas se agrupan dependiendo de la funcionalidad de las mismas y se suelen establecer en el siguiente orden:

Primero se agrupan los estilos básicos como, por ejemplo, la fuente por defecto, los márgenes de los elementos, etc.

Después se agrupan los estilos de la estructura como, por ejemplo, la altura, anchura y posición de los elementos.

A continuación, van el estilo de los enlaces.

Por último, los estilos de cada una de las zonas como son títulos, elementos de cabecera, etc.

Otra cosa a destacar es el uso de los comentarios CSS para mejorar la estructura de las hojas de estilo que sean excesivamente complejas.

Las hojas de estilo CSS permiten añadir comentarios entre las reglas y los estilos de la misma. Los comentarios son bloques de texto que añaden los diseñadores en un archivo para ofrecer información relativa al trozo de código. Estos comentarios también se utilizan en archivos HTML y .js.

Los comentarios son totalmente ignorados por los navegadores, puesto que, la única utilidad que tienen es proporcionar información al desarrollador al programar archivos muy complejos y largos.

En CSS, puede escribir comentarios utilizando los caracteres `/*` para abrir el comentario y, `*/` para cerrarlo. Vea un ejemplo:

```
/*Esto es un comentario de una hoja de estilo CSS*/
```

Además, puede añadir comentarios que ocupen varias líneas del documento, en cambio, no podrá añadir un comentario dentro de otro comentario. Vea el ejemplo:

```
/*Esto es un comentario
```

De una hoja de estilo CSS

```
Que ocupa varias líneas*/
```

Aunque es cierto que los navegadores no son capaces de interpretar los comentarios que añade a un documento, éstos se envían junto con los demás datos al navegador, es por ello que, no debe incluir ninguna información que pueda ser confidencial como, por ejemplo, contraseñas de su sitio o urls privadas.

Por otro lado, para añadir un comentario en un archivo HTML, lo haga de manera diferente al de los archivos CSS. Para añadir un comentario en un archivo HTML lo haga de la siguiente manera:

```
<!--Esto es un comentario en un archivo HTML-->
```

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

En un archivo HTML

```
De varias líneas-->
```

Ahora, que conoce todas las recomendaciones y los atributos que puede utilizar para crear y cambiar los estilos de un documento web, toca comenzar a hacerlo.

Anteriormente, ya ha creado un archivo HTML con su estructura básica y, su respectivo documento CSS enlazado al mismo. Ahora, toca saber cómo se relacionan los selectores de los elementos con la propia página web.

Una de las grandes dudas es si utilizar el atributo id o class para establecer los estilos. La respuesta es que utilizará el atributo class para las etiquetas que aparecerán más de una vez en la página, en cambio, utilizará el atributo id para las que solo aparecen una única vez. Se entiende etiqueta como la palabra que añadirá como valor a los atributos id y class.

Por lo tanto, en el archivo CSS que ha creado, va a escribir los atributos id y class. Como ya vió anteriormente, estos dos atributos son los selectores que escribirá en su archivo CSS donde aplicar el estilo determinado.

Resumiendo, en su archivo CSS se añade un selector que coincida con el nombre especificado en el atributo class o id que ha establecido en el documento HTML.



TOME NOTA

**Debe recordar que los selectores de clase están precedidos por un punto y los selectores de id por una almohadilla.**

Para finalizar, debe recordar que cuando tenga listos sus archivos HTML y CSS, debe pasar al proceso de validación de la página.

Este proceso es uno de los más importantes y la mayoría de gente que realiza una web se lo saltan. Sin embargo, es algo que debe realizar de manera obligatoria si quiera tener una página web profesional y funcional.

Además, debe realizar este proceso cada cierto tiempo para ir actualizando su web. Siempre es mejor detectar los posibles errores en la página a tiempo, y poder resolverlos cuanto antes, ya que, de no ser así, tomará mucho más trabajo realizarlo en un futuro.

Para ello, puede echar mano de estos dos validadores, uno para HTML y el otro para CSS:

[Validador CSS](#)

[Validado HTML](#)

En caso de que estuviera desarrollando la web utilizando un firewall o utilizando un servidor local, podrá utilizar las herramientas de validación que ofrece Firebug.

#### **1.6.3. Adaptación de los ficheros de estilo para distintas páginas web**

Como ya se mencionó anteriormente, una de las características más destacables de CSS es que permite definir diversos estilos para diferentes medios como móviles, impresoras, pantallas, proyectores, etc.

Por otro lado, CSS también establece algunas propiedades específicas para algunos medios, por ejemplo, la página y los saltos de página para las impresoras o el volumen y tipo de voz para medios de audio. Vea a continuación una tabla con los nombres que utiliza CSS para poder identificar cada uno de los medios:

Medio	Descripción
<b>Tv</b>	Dispositivos con baja resolución y televisores
<b>Tty</b>	Dispositivos textuales que solo pueden interpretar texto
<b>Speech</b>	Sintetizadores de voz
<b>Screen</b>	Monitores
<b>Projection</b>	Proyectores

<b>Print</b>	Impresoras y navegadores en vista de impresión
<b>Handheld</b>	Móviles, PDA, etc.
<b>Embossed</b>	Impresoras braille
<b>Braille</b>	Dispositivos táctiles que utilizan Braille
<b>all</b>	Todos los medios

Los medios que más se utilizan actualmente son screen, para los monitores del ordenador, print, para poder imprimir la página web, y handheld, que se utiliza para los dispositivos móviles.

CSS hace una clasificación de los medios en diferentes grupos dependiendo de sus características. Vea a continuación la tabla:

Medio	Continuo/ Paginado	Visual/Auditivo/táctil/vocal	Mapa	de	Interactivo/
			bits/Caracteres		Estático
<b>Screen</b>	Continuo	Auditivo y visual	Mapa de bits	Ambos	
<b>Projection</b>	Paginado	Visual	Mapa de bits	Interactivo	
<b>Print</b>	Paginado	Visual	Mapa de bits	Estático	
<b>Handheld</b>	Ambos	Visual, vocal, auditivo	Ambos	Ambos	
<b>Embossed</b>	Paginado	Táctil	Caracteres	Estático	
<b>Braille</b>	Continuo	Táctil	Caracteres	Ambos	
<b>Tv</b>	ambos	Auditivo y visual	Mapa de bits	Ambos	
<b>Tty</b>	Continuo	Visual	Caracteres	Ambos	
<b>speech</b>	Continuo	Vocal	-	ambos	

Que CSS permita modificar los estilos de una página dependiendo del medio en el que se muestren es una gran ventaja. Para ello, puede hacerlo de cuatro maneras diferentes.

El uso de las reglas @media permite indicar en CSS de manera directa el medio o los medios en los que se van a aplicar los estilos incluidos en la misma. Para indicar qué medio será en el que se incluirá los estilos, debe escribir el nombre del medio a continuación de la regla. En caso en que vaya a aplicar los estilos a más de un medio, tan solo debe escribir los medios separados por comas. Vea un ejemplo:

```
@media print{
    Body {Font-size: 11pt;}
}

@media screen{
    Body {Font-size: 14px;}
}
```

```
@media screen, print{  
    Body {line-height: 2}  
}
```

En este ejemplo puede ver que se establece una fuente con un tamaño de 14 píxeles cuando se muestre a través de un monitor. En cambio, cuando se muestre en una impresión, la letra tendrá un tamaño de 11 puntos. Para finalizar, tanto en monitor como en impresión, el texto tendrá un interlineado de 2 veces el tamaño de la letra del texto.

Al utilizar reglas de tipo @import para enlazar archivos CSS externos, puede indicar el medio en el que se establecen los estilos, tan solo, indicando el nombre del medio posterior a la url del archivo CSS. Vea un ejemplo:

```
@import url ("estilosGenerales") screen;
```

```
@import url ("estilosImpresoras") print;
```

En este ejemplo puede ver que cuando el archivo CSS se muestra por pantalla, se cargará el archivo correspondiente a los estilos de la pantalla. Sin embargo, cuando se muestra en una impresión, el archivo CSS que se cargará será el que contiene los estilos para imprimir.

En caso de que necesite que un archivo CSS se aplique para varios medios, solo debe indicar los nombres de esos medios separados por comas, tal y como hacía anteriormente. Por otra parte, en caso de que no indique ningún medio en la regla @import, el navegador interpretará que el medio es all, es decir, que se aplicará el mismo estilo a todos los medios.

Por otra parte, si utiliza la etiqueta <link> para enlazar los archivos CSS, puede utilizar el atributo media para indicar el tipo de medio en el que se aplicarán los estilos de ese archivo. Vea un ejemplo:

```
<link rel="stylesheet" type="text/css" media="screen" href="estilosGenerales.css"/>
```

```
<link rel="stylesheet" type="text/css" media="print, handheld" href="estilosImpresoras.css"/>
```

Como puede ver en el ejemplo, los estilos del primer archivo se van a mostrar en los monitores y los del segundo archivo se mostrarán en las impresiones y en los móviles.

En caso de que la etiqueta <link> no indique el medio, se interpretará que los estilos se aplicarán a todos los medios, al igual que pasaba con el anterior ejemplo.

Por último, en CSS también puede mezclar los tres métodos para indicar los estilos que se aplicarán a los medios correspondientes. Vea un ejemplo:

```
<link rel="stylesheet" type="text/css" media="screen" href="estilosGenerales.css"/>
```

```
@import url ("estilosGenerales") screen;  
  
@media print {  
  
/*Estilos impresión*/  
  
}
```

Para finalizar, debe saber que si incluye en una hoja de estilo varios estilos dependiendo del dispositivo donde se vaya a mostrar la página, define los estilos por defecto al principio de la hoja de estilo, y posteriormente, define las reglas y los estilos específicos para cada medio. La razón de esto es que los estilos por defectos se aplicarán en cascada sobre cada estilo que coincida con la regla.

## 2. DISEÑO, UBICACIÓN Y OPTIMIZACIÓN DE LOS CONTENIDOS DE UNA PÁGINA WEB

En este tema va a aprender a crear un documento funcional donde va a añadir elementos que harán que su página sea dinámica y atractiva para el usuario. Cree el documento HTML con la estructura de la página desde cero y, después, aplicar los estilos a cada uno de los elementos para dar un formato visual aún más atractivo.

Cuando esté creando la hoja de estilo, utilizará las clases, previamente codificadas en HTML, para establecer estilos específicos a determinado grupo de elementos.

Además, aprenderá sobre algunos de los programas que se utilizan tanto para editar código, como para subir los documentos a un servidor.

Aprenderá a insertar imágenes en su web, enlazándolas con el tamaño de imagen real. Para modificar el tamaño de las imágenes lo haga mediante CSS.

Analizará cada uno de los elementos paso a paso de la web que va a crear, para de esta manera comprender mejor cómo funcionan en relación a los demás elementos. Por otra parte, aprenderá a establecer la herencia de los objetos unos con otros.

Además de añadir enlaces en la web, va a enlazar un documento PDF descargable desde la propia web.

Por último, va a aprender cómo hacer una guía de usuario paso a paso. Debe tener en cuenta varios criterios como la legibilidad y ponerse en la piel del usuario para poder escribir una guía útil y eficaz.

### 2.1. Creación de un documento funcional

Antes de comenzar a crear un documento funcional donde añadirá los conocimientos de CSS que tiene, necesita los siguientes elementos:

#### **Un equipo informático.**

Debe tener un ordenador cualquiera, sin importar si tiene Mac, Windows o Linux. Podría servir incluso una PDA. Lo que sí es importante es que tenga un equipo potente para crear páginas web. Un dato interesante es que los ordenadores de hace más de diez años eran menos potentes que una PDA.

Por otro lado, debe tener en cuenta que cuanto mayor tamaño tenga su pantalla mejor, ya que, tendrá que trabajar con varias ventanas abiertas,

### **Conexión a Internet.**

Aunque tener Internet puede ser algo opcional, ya que, existe la posibilidad de crear páginas web en entorno local, es muy recomendable para poder subir la página a un servidor y publicarla en Internet.

### **Editor de texto.**

Como ya sabe, un documento HTML es un documento de texto en el que la extensión es .htm o .html, en vez de .txt. Lo mismo ocurre con los archivos CSS donde la extensión es .css.

Realmente, para crear archivos de este tipo podrá utilizar el bloc de notas que ofrece Windows. Sin embargo, se recomienda utilizar una herramienta especializada para escribir código como, por ejemplo, Visual Code. Este editor es gratuito y puede descargarlo desde su [web oficial](#) e instalarlo en su equipo de manera sencilla.

Utilizar este tipo de herramientas ofrece como ventaja principal el reconocimiento de los diferentes lenguajes de programación. De esta manera, los destacará para que sea más sencillo utilizar el lenguaje, ya que, facilitará la tarea de programar. Por otra parte, también enumera las líneas de código, de esta manera va a ser más sencillo localizar los errores que puedan surgir durante el código.

Por otra parte, el bloc de notas que ofrece Mac es el Textedit y el que ofrece Linux se llama Gedit.

Además, para crear páginas web también tiene los editores WYSIWYG, que permiten crear páginas sin tocar el código HTML. Estas herramientas funcionan de la misma manera que lo hacen los procesadores de texto al aplicar la cursiva o negrita a un texto, o al insertar una imagen. Es decir, lo hace todo de manera automática sin tener que conocer ningún tipo de código de programación.

La herramienta de este tipo más conocida es Dreamweaver. Sin embargo, como en este curso quiera aprender a utilizar CSS y HTML, evitará el uso de estas herramientas. Además, el uso de este tipo de programas tiene algunos inconvenientes como, por ejemplo, que no es gratuito y que no crea un código optimizado o sencillo de comprender para un programador novel.

### **Programa FTP.**

Precisa este tipo de programas para, posteriormente a haber creado la página web, pueda subirla al servidor mediante este tipo de programa.

### **Imágenes.**

Para las imágenes que utilizará en su web, debería utilizar un programa de retoque fotográfico para editar las fotos de manera que pueda reducir su tamaño y guardarlas en PNG o JPG.

Hasta los programas más básicos de edición fotográfica permitirán realizar estas modificaciones en una imagen. Puede utilizar diversos editores de imágenes gratuitos como Gimp. Este editor es muy completo y parecido a Photoshop.

### **Navegador.**

Los epígrafes, que va a ver más adelante, van a funcionar en cualquier navegador como Google Chrome o Mozilla Firefox. Lo más aconsejable es que utilice la última versión del navegador que más guste. Sin embargo, el navegador más propicio para probar las páginas web es el navegador Mozilla Firefox, puesto que, ofrece muchas herramientas de depuración de errores.

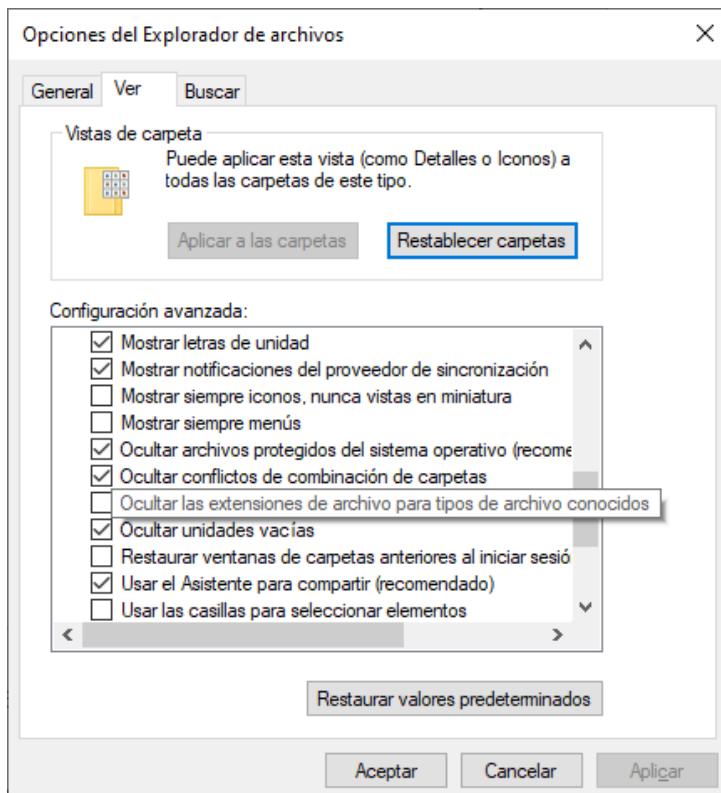
### **Bloc de notas.**

Tener un bloc de notas siempre será útil para poder apuntar cosas como trozos de código, reglas, o como recordatorio de ideas a implementar. Tener esta información apuntada en un archivo de texto es muy útil para poder consultarla a medida que avanza en la creación de la web.

### **Mostrar extensiones.**

Windows por defecto oculta las extensiones de los archivos. Sin embargo, a veces es necesario que se muestren, ya que, de esa forma en el caso de tener varios archivos de imagen, con distintos formatos de imagen, dependerá el uso que vaya a darle en la web. Debe saber qué archivo de imagen tiene la extensión correcta para el uso que va a darle y, por ello, va a ser muy útil que se muestren las extensiones de los archivos.

Para que se muestren los archivos, debe ir a Herramientas, Opciones de Carpeta y haga clic en Ver. A continuación, se mostrará la configuración avanzada. Aquí es donde debe desactivar la casilla de Ocultar extensiones del archivo para tipos de archivos conocidos. Vea la siguiente imagen de ejemplo:



De este modo, podrá ver la extensión de todos los archivos.

### Creación de archivos CSS y HTML.

Lo recomendado es que cree una carpeta donde se guardarán los dos archivos. Estos dos archivos puede crearlos con Visual Code o con un simple blog de notas. Tan solo debe crear dos documentos, uno con la extensión .html y el otro con la extensión .css.

Cuando haya creado el archivo, éste se mostrará en la carpeta con el icono del navegador que tenga por defecto. Para abrir el archivo con su herramienta de edición de texto, tan solo, debe pulsar el botón derecho del ratón sobre el archivo y seleccionar la opción de Abrir con, después pulsar el editor de texto deseado. De esta manera es como modificar el documento web.

Para abrir el documento CSS, debe realizar los mismos pasos para poder comenzar a editar el documento.

#### 2.1.1. Descripción de los objetivos de la página web

En este epígrafe, va a comenzar a crear la página web desde cero. La página que va a crear va a ser una página básica en la que va a aplicar diferentes estilos. Además, va a estar compuesta por elementos funcionales que van a ser útiles para ir practicando con todos los elementos que ha visto hasta este punto.

Los elementos funcionales que va a ver son los siguientes:

- Cuerpo del documento
- Menú
- Título
- Tabla

Con estos elementos construirá el documento principal de HTML. Todos los elementos mencionados anteriormente tendrán un estilo aplicado mediante CSS, el cual irá editando.

Va a crear la página de un fotógrafo donde se mostrarán sus fotos, su CV y dónde podrán con él.

Vea a continuación los elementos básicos que va a tener la página:

## Página web

[Inicio](#) [Fotografías](#) [CV](#) [Contacto](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem, vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit, neque metus accumsan ligula, non tempor ante est ac ante. Curabitur bibendum, quam vitae dapibus maximus, eros lacus congue massa, nec eleifend urna augue sit amet dui. Cras bibendum sit amet orci sit amet consectetur. Nam nec tortor et risus vehicula lobortis. Nam tristique ipsum non metus lobortis pharetra. Sed porta massa ut justo feugiat, sit amet tincidunt lectus varius. Nunc volutpat ullamcorper quam, vel molestie nisl suscipit sed. Suspendisse nisi purus, bibendum nec bibendum cursus, placerat luctus justo. Nullam mattis sit amet eros ac iaculis.

Como puede observar, la página va a tener elementos muy básicos. Un menú, un título y un pequeño texto. A partir de aquí, va a comenzar a aplicarle los estilos para, de ese modo, obtener una páginas más profesional y atractiva.

A continuación, va a ver los cuatro documentos que debe crear para componer la web. Cada uno de ellos va a pertenecer a un elemento del menú:

- Inicio
- Fotos
- CV
- Contacto

Ahora, va a ver cada uno de los elementos que va a añadir en cada documento web:

### Inicio.

Esta página será la página principal donde tendrá varios elementos funcionales que irá añadiendo poco a poco dándole los estilos que desee.

Los elementos que va a crear para esta página son:

Encabezado: donde pondrá un texto descriptivo de la página web que describa la misma.

Menú: El menú va a servir para navegar por las diferentes páginas de la web y poder retornar a la página principal.

Texto descriptivo: En su página de Inicio, debe introducir un texto descriptivo que describirá el sitio web. Esto sirve para darle una noción a los usuarios sobre qué contenidos van a encontrar en su web.

Vea una imagen de lo que quiera conseguir en un principio en la página de inicio:

## Página web

[Inicio](#) [Fotografías](#) [CV](#) [Contacto](#)

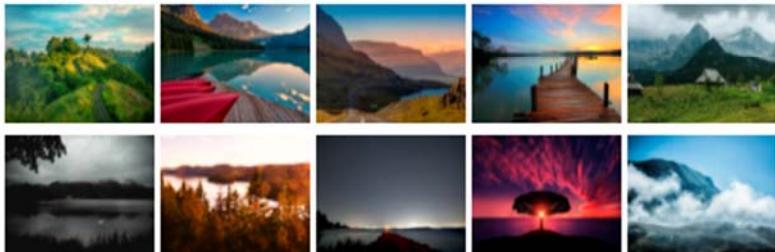
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem, vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit, neque metus accumsan ligula, non tempor ante est ac ante. Curabitur bibendum, quam vitae dapibus maximus, eros lacus congue massa, nec eleifend urna augue sit amet dui. Cras bibendum sit amet orci sit amet consectetur. Nam nec tortor et risus vehicula lobortis. Nam tristique ipsum non metus lobortis pharetra. Sed porta massa ut justo feugiat, sit amet tincidunt lectus varius. Nunc volutpat ullamcorper quam, vel molestie nisl suscipit sed. Suspendisse nisi purus, bibendum nec bibendum cursus, placerat luctus justo. Nullam mattis sit amet eros ac iaculis.

### Fotografías.

Esta es la página donde redirigirá cuando pulse Fotografías en el menú. En esta página verá algunos de los trabajos fotográficos de la persona en sí. Las fotos estarán dentro de una tabla. Las imágenes se mostrarán como miniaturas de 100px por 80px. Al pulsar sobre cada imagen, se abrirá en una nueva ventana con su tamaño real. Vea cómo se verá la base de esta página:

## Página web

[Inicio](#) [Fotografías](#) [CV](#) [Contacto](#)



## CV.

En esta página construirá un apartado donde va a haber un enlace que dejará descargar el CV de la persona. Vea la base:

# Página web

[Inicio](#) [Fotografías](#) [CV](#) [Contacto](#)

Pulse aquí para descargar CV

## Contacto.

En esta página, va a tener la foto del fotógrafo y un recuadro donde podrá enviarle un mensaje para contactar con él. Vea el ejemplo:

Página web

[Inicio](#) [Fotografías](#) [CV](#) [Contacto](#)



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem, vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit, neque metus accumsan ligula, non tempor ante est ac ante.

Comentarios

### 2.1.2. Definición de los elementos funcionales de la página web

En este epígrafe va a ver el código de los elementos funcionales de la página que está creando. Además, verá los cuatro documentos que forman la página web y los crear paso a paso para formar la estructura base de la página.

## Inicio

Va a ver el código HTML que se ha utilizado para crear el Inicio de la web. Este documento lo llamará main.html o index.html:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>

</head>

<body>

<div><h1>Página web</h1></div>

<div>

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<div>

<span>L</span>orem ipsum dolor sit amet, consectetur adipiscing elit.

Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem,
vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit,
neque metus accumsan ligula, non tempor ante est ac ante.
```

Curabitur bibendum, quam vitae dapibus maximus, eros lacus congue massa,  
nec eleifend urna augue sit amet dui. Cras bibendum sit amet orci sit amet consectetur.  
Nam nec tortor et risus vehicula lobortis. Nam tristique ipsum non metus lobortis pharetra.  
Sed porta massa ut justo feugiat, sit amet tincidunt lectus varius.  
Nunc volutpat ullamcorper quam, vel molestie nisl suscipit sed.  
Suspendisse nisi purus, bibendum nec bibendum cursus, placerat luctus justo.  
Nullam mattis sit amet eros ac iaculis.

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

Como puede observar, en el código hay una primera división que funciona como contenedor de todo. Vea el ejemplo:

```
<body>  
<div>  
<div><h1>Página web</h1></div>  
  
<div>  
<a href="main.html">Inicio</a>  
<a href="fotos.html">Fotografías</a>  
<a href="cv.html">CV</a>  
<a href="contacto.html">Contacto</a>  
  
</div>
```

<div>

<span>L</span>orem ipsum dolor sit amet, consectetur adipiscing elit.

Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem,

vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit,

neque metus accumsan ligula, non tempor ante est ac ante.

Curabitur bibendum, quam vitae dapibus maximus, eros lacus congue massa,

nec eleifend urna augue sit amet dui. Cras bibendum sit amet orci sit amet consectetur.

Nam nec tortor et risus vehicula lobortis. Nam tristique ipsum non metus lobortis pharetra.

Sed porta massa ut justo feugiat, sit amet tincidunt lectus varius.

Nunc volutpat ullamcorper quam, vel molestie nisl suscipit sed.

Suspendisse nisi purus, bibendum nec bibendum cursus, placerat luctus justo.

Nullam mattis sit amet eros ac iaculis.

</div>

</div>

</body>

Posteriormente, observa que hay una división para cada uno de los elementos:

La primera división es la del título principal:

<div><h1>Página web</h1></div>

Después, encontrará la división del menú:

<div>

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

```
<a href="cv.html">CV</a>  
  
<a href="contacto.html">Contacto</a>  
  
</div>
```

Y, por último, encontrará la división que va a contener los elementos que quiera mostrar en cada uno de las páginas web:

```
<div>  
  
<span>L</span>orem ipsum dolor sit amet, consectetur adipiscing elit.  
  
Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem,  
  
vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit,  
  
neque metus accumsan ligula, non tempor ante est ac ante.  
  
Curabitur bibendum, quam vitae dapibus maximus, eros lacus congue massa,  
  
nec eleifend urna augue sit amet dui. Cras bibendum sit amet orci sit amet consectetur. Nam nec to  
rto et risus vehicula lobortis. Nam tristique ipsum non metus lobortis pharetra.  
  
Sed porta massa ut justo feugiat, sit amet tincidunt lectus varius.  
  
Nunc volutpat ullamcorper quam, vel molestie nisl suscipit sed.  
  
Suspendisse nisi purus, bibendum nec bibendum cursus, placerat luctus justo.  
  
Nullam mattis sit amet eros ac iaculis.  
  
</div>
```

## Fotografías

Vea el código que se ha utilizado para establecer los elementos de la página fotografías:

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

```
<title>Página fotógrafo</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>

<div>

<div><h1>Página web</h1></div>

<div>

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<div>

<table>

<tr>

<td><a href="imagenes/foto1.jpg" title="imagen 1"></a></td>

<td><a href="imagenes/foto2.jpg" title="imagen 2"></a></td>

<td><a href="imagenes/foto3.jpg" title="imagen 3"></a></td>

<td><a href="imagenes/foto4.jpg" title="imagen 4"></a></td>
```

```
<td><a href="imagenes/foto5.jpg" title="imagen 5"></a></td>

</tr>

<tr>

<td><a href="imagenes/foto6.jpg" title="imagen 6"></a></td>

<td><a href="imagenes/foto7.jpg" title="imagen 7"></a></td>

<td><a href="imagenes/foto8.jpg" title="imagen 8"></a></td>

<td><a href="imagenes/foto9.jpg" title="imagen 9"></a></td>

<td><a href="imagenes/foto10.jpg" title="imagen 10"></a></td>

</tr>

</table>

</div>

</div>

</body>

</html>
```

Como se puede observar, en este documento puede encontrar en la primera parte el mismo código que había en la página de Inicio con la primera división donde está el título de la página y la segunda división para el menú.

Lo que diferencia este documento del documento Inicio es la tercera división que es donde contiene una tabla con las diferentes imágenes que quiera mostrar en este álbum. La tabla que ha hecho tiene cinco columnas y dos filas.

A cada una de las celdas de la tabla, le ha añadido un archivo de imagen mediante la etiqueta `<a href>`, además, ha añadido un título a las imágenes mediante la etiqueta `<title>` y una url hacia el archivo de imagen mediante la etiqueta `<src>`.

Por último, ha establecido el alto y el ancho de cada imagen con las etiquetas `<height>` y `<width>`.

## CV

EL código que ha utilizado para crear la página CV es el siguiente:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>

</head>

<body>

<div>

<div><h1>Página web</h1></div>

<div>

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<div>

<div><p>Pulse aquí para descargar CV</p></div>

</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

Lo mismo que ha hecho con los demás documentos pertenecientes a esta web, este documento también tiene su título principal y el menú mediante el cual puede navegar por las diferentes páginas.

Además, también tiene un `<div>` en el que está añadido el contenido de este documento web específico, que es el que muestra el enlace de descarga del CV.

Por último, dentro del texto que ha añadido es donde insertará un vínculo hacia el CV, que será un archivo PDF. Para llevar esta tarea a cabo, utilizará el elemento `<a>` donde añadir la ruta específica hacia el documento.

## Contacto

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>

</head>

<body>

<div>

<div><h1>Página web</h1> </div>

<div>

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

<a href="cv.html">CV</a>
```

```
<a href="contacto.html">Contacto</a>

</div>

<div>

<div>



</div>

<span>L</span>orem ipsum dolor sit amet, consectetur adipiscing elit.

Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem,
vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit,
neque metus accumsan ligula, non tempor ante est ac ante.

<form method="post" action="mailto:correo@dominio.com" enctype="text/plain">

<p>Comentarios</p>

<textarea rows="6" cols="30" name="comentarios"></textarea>

<p><button type="submit">Enviar</button>&nbsp; &nbsp;
<button type="reset">Cancelar</button></p>

</form>

</div>

</div>

</body>

</html>
```

Como se puede observar, este documento está formado en la primera sección de igual manera que el resto de documentos, primero el título y después el menú.

El contenido específico para este documento es el formulario de contacto y la imagen personal del fotógrafo. Además, el formulario que ha añadido es un formulario muy básico, ya que, simplemente debe enviar mensajes al correo electrónico del fotógrafo para poder contactarle.

Por último, se ha añadido una foto personal utilizando la etiqueta <img>.

### 2.1.3. Descripción de cada elemento

En este apartado, va a describir cada uno de los elementos que se encuentran en las páginas web de manera que permita conocer cada una de las páginas web de manera específica para, a continuación, poder aplicar los estilos específicos a cada una de las partes.

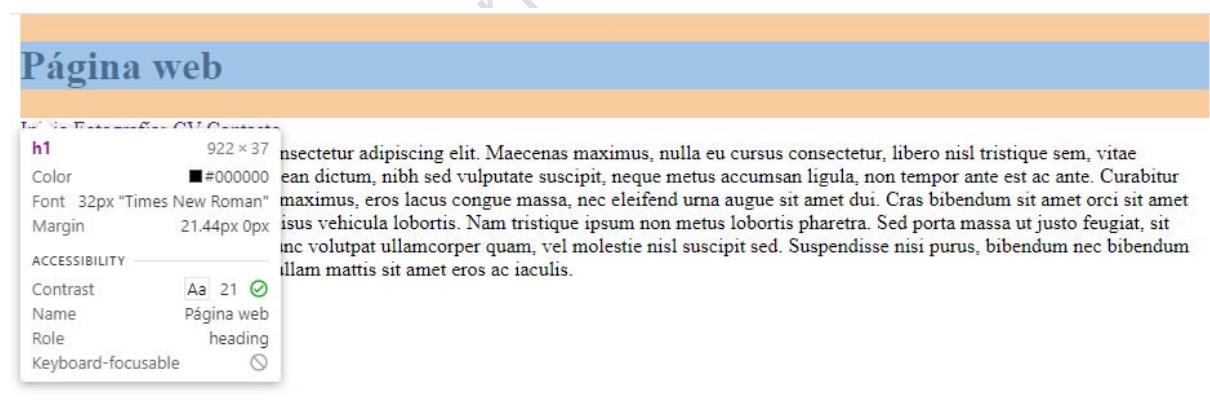
Va a comenzar por la página de Inicio.

#### Inicio

Esta página puede ver que se divide en tres partes diferenciadas de manera clara. Las divisiones que son comunes para todas las páginas son el menú y el título principal, por lo tanto, se explicarán una sola vez.

Como aún no ha añadido ningún estilo a la página, puede ver esas divisiones con inspeccionando la página desde un navegador y verá las divisiones del título y del menú.

Si se fija en la siguiente imagen, puede ver el espacio que tiene el título principal en la página. En este espacio puede insertar una imagen o un color de fondo.



Esta primera división es donde colocará el título de la página o un banner para presentar la página. Ahora mismo, este espacio solo contiene un texto, pero puede añadir lo que desee.

Posteriormente, en la siguiente imagen puede ver la división donde se encuentra el menú mediante el cual podrá navegar por la página:

## Página web

The screenshot shows a browser's developer tools with the 'Elements' tab selected. At the top, there are navigation links: Inicio, Fotografías, CV, and Contacto. Below this, a sidebar displays various element properties:

Element	Value
div	922 × 18
Color	#000000
Font	16px "Times New Roman"
ACCESSIBILITY	
Name	
Role	generic
Keyboard-focusable	<input checked="" type="checkbox"/>

The main content area contains a large amount of placeholder text (Lorem ipsum) in Spanish, describing a maximus element with various styling details.

Estas dos divisiones serán comunes para todos los documentos que formen el sitio web, por lo tanto, en esas páginas debe repetir el mismo código.

Por último, llega a la tercera y última división que es donde añadir los elementos específicos de cada página. En la página Inicio, tan solo contiene un texto de bienvenida y explicativo sobre la página para que los usuarios puedan saber sobre qué va la página en sí.

Vea la división que muestra la herramienta de inspección del navegador:

## Página web

The screenshot shows a browser's developer tools with the 'Elements' tab selected. At the top, there are navigation links: Inicio, Fotografías, CV, and Contacto. Below this, a sidebar displays various element properties:

Element	Value
div	922 × 108
Color	#000000

The main content area contains a large amount of placeholder text (Lorem ipsum) in Spanish, describing a maximus element with various styling details.

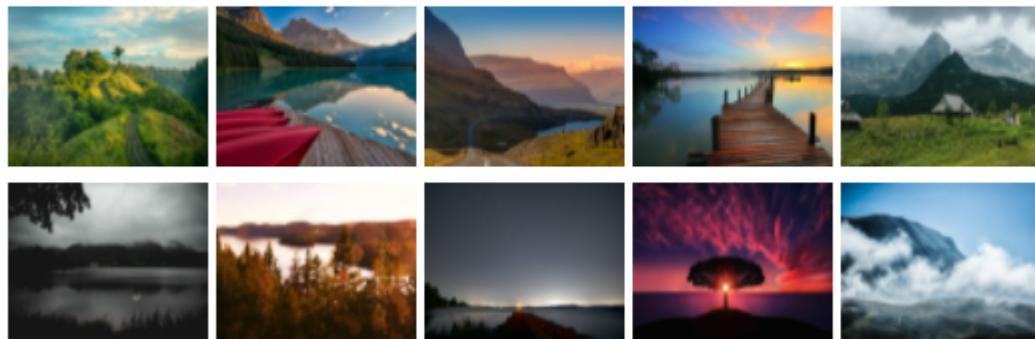
## Fotografías

En esta página, encuentre los trabajos fotográficos del fotógrafo al que pertenece la página. Para ello, se ha creado una tabla donde se han ido introduciendo todas las fotos en miniaturas. Al seleccionar la foto, ésta se abrirá en otra ventana del navegador y podrá verlas en tamaño real.

Como aún no se le ha aplicado ningún estilo a ninguno de los elementos, la página se muestra de la siguiente manera:

# Página web

[Inicio](#) [Fotografías](#) [CV](#) [Contacto](#)



## CV

En este documento, aún no ha añadido ningún estilo, tan solo hay una frase que indica que si pulsa ahí podrá descargar el documento del CV del fotógrafo. Sin embargo, aún no tiene ninguna funcionalidad. Esto lo haga más adelante aplicándole CSS.

El documento CV.html luce actualmente de la siguiente manera:

# Página web

[Inicio](#) [Fotografías](#) [CV](#) [Contacto](#)

Pulse aquí para descargar CV

## Contacto

En el documento de contacto, puede ver que tiene una caja de texto donde los usuarios pueden escribir un mensaje y enviarlo al propietario de la página.

Además, también aparece una imagen del fotógrafo, en este caso una fotógrafa, para que los usuarios la puedan conocer.

Por último, el formulario cuenta con dos botones donde los usuarios podrán enviar el mensaje si pulsan en Enviar o borrar el texto en caso de que pulse el botón Cancelar.

Como actualmente no cuenta con ningún estilo aplicado, esta página se ve así:

#### Página web



## 2.2. Diseño de los contenidos

A continuación, en este epígrafe, va a añadir los estilos a sus páginas para que queden más atractivas para los usuarios. Los estilos que añada van a ser sencillos y claro para que no obtenga una página muy recargada. Va a ver el diseño de cada una de las páginas paso a paso.

Al comenzar a diseñar una web, lo primero que ha hecho es pensar el aspecto que tendría la misma. Para ello, utiliza un lápiz y un papel en el que apunte los elementos y los estilos que tendría el sitio web.

Lo primero que hizo fue un pequeño croquis y cuando obtuvo el resultado esperado y lo dió por válido, es cuando empezó a desarrollar la web mediante código.

Lo que obtuvo fue un sitio en el que habría un bloque principal conteniendo todos los elementos de la web, otro bloque para el logo, banner o título y otro para un menú de navegación, el último bloque sería para el contenido específico de cada una de las páginas.

### Diseño de página de Inicio

En la página de Inicio el cuerpo del documento tendrá un fondo color blanco, no tendrá márgenes y tendrá un ancho de 1000 píxeles.

Por otro lado, hay una división que es donde se contendrán todos los elementos del documento. Esta división tiene una anchura de 800px y no tiene márgenes. El texto se encuentra alineado al centro con un margen superior de 100 píxeles. El color del borde es blanco y las esquinas son redondeadas.

El diseño donde va el título será con un fondo blanco. Las letras del título tienen un tamaño extra grande y la tipografía es Verdana, negrita y de color negro. Además, tendrá un espaciado superior e inferior de 20px. Por último, contendrá un borde de 2px de ancho color negro y con esquinas redondeadas.

El menú cuenta con un borde de 2px de ancho con color darkslateblue con un margen superior e inferior de 10px, las esquinas del borde estarán redondeadas su fondo será rgb(116, 234, 238).

El texto del menú está en versalitas, con un espacio de 10px en la parte superior e inferior. Además, el texto estará en negrita y tendrá un espaciado entre letras de 10px. El color de los enlaces del menú será darkred sin el típico subrayado de enlace. Cuando pase el puntero del ratón por encima de los enlaces, el color del texto del menú pasará a ser fuchsia. Si el enlace ya ha sido visitado, el color del texto del menú cambiará a blueviolet.

Por último, la tercera división de la página tendrá una posición relativa, un color de borde gray con 2 píxeles de ancho y visible. La altura máxima de la división serán 300 píxeles, el texto que se encuentra en la división estará justificado, con un espacio de 0 píxeles en la parte superior y 15 píxeles en el resto de las partes. Por último, la primera letra del texto será capitular, por lo tanto, le aplicar el diseño de fuente extra grande.

Vea a continuación el código CSS resultante de todos estos estilos:

```
.general{  
    width: 800px;  
    margin: 0 auto;  
    text-align: center;  
    margin-top: 100px;  
    border-color: white;  
    border-radius: 10px;  
    background-color: white;  
}
```

```
.titulo{  
    font-size: xx-large;  
    font-family: Verdana, Geneva, Tahoma, sans-serif;  
    font-weight: bold;  
    color: black;  
    background-color: white;  
    padding-top: 20px;  
    padding-bottom: 20px;  
    border-width: 2px;  
    border-style: solid;  
    border-color: black;  
    border: 2px solid black;  
    border-radius: 10px;  
}
```

```
.menu{  
    background-color: rgb(116, 234, 238);  
    margin-top: 10px;  
    margin-bottom: 10px;  
    border-style: solid;  
    border-color: darkslateblue;  
    border-width: 2px;  
    padding-top: 10px;
```

```
padding-bottom: 10px;  
  
font-weight: bold;  
  
font-variant: small-caps;  
  
word-spacing: 10px;  
  
border-radius: 10px;  
  
}
```

```
.menu a{  
  
color: darkred;  
  
text-decoration: none;  
  
}
```

```
.menu a:hover{  
  
color: fuchsia;  
  
}
```

```
/* .menu a:visited {  
  
color: blueviolet;  
  
} */
```

```
.contenido {  
  
position: relative;  
  
padding-top: 20px;
```

---

```
padding-right: 15px;  
  
padding-bottom: 15px;  
  
padding-left: 15px;  
  
border-style: solid;  
  
border-color: gray;  
  
border-width: 2px;  
  
text-align: justify;  
  
min-height: 300px;  
  
border-radius: 10px;  
  
}
```

```
#capitular{
```

```
font-size: xx-large;  
  
}
```

Vea el resultado de la página de inicio:

**Página Fotografía**

INICIO FOTOGRAFIAS CV CONTACTO

**L**orem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem, vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit, neque metus accumsan ligula, non tempor ante est ac ante. Curabitur bibendum, quam vitae dapibus maximus, eros lacus congue massa, nec eleifend urna augue sit amet dui. Cras bibendum sit amet orci sit amet consectetur. Nam nec tortor et risus vehicula lobortis. Nam tristique ipsum non metus lobortis pharetra. Sed porta massa ut justo feugiat, sit amet tincidunt lectus varius. Nunc volutpat ullamcorper quam, vel molestie nisl suscipit sed. Suspendisse nisi purus, bibendum nec bibendum cursus, placerat luctus justo. Nullam mattis sit amet eros ac iaculis.

Ahora, se explicará el diseño del resto de páginas mencionando solo su diseño específico en la tercera división.

### Página Fotografías

Para diseñar esta página, ha añadido una tabla que recoge todas las imágenes en su interior.

El diseño de la tabla tendrá un 70% de ancho, no tendrá bordes y tendrá márgenes automáticos tanto a la izquierda como a la derecha, con una altura de 300px.

Por otro lado, las celdas tendrán un ancho de 30%, con margen derecho e izquierdo automático.

Por último, las fotografías tendrán un borde de 1px de color limegreen y visible.

Vea a continuación el código CSS para esta página:

```
table{  
    width: 70%;  
  
    margin-left: auto;  
  
    margin-right: auto;  
  
    height: 300px;  
  
    border-style: none;  
}  
  
  
td{  
    width: 30%;  
  
    margin-left: auto;  
  
    margin-right: auto;  
  
}  
  
  
.foto{
```

```
border-width: 1px;  
border-color: limegreen;  
border-style: solid;  
}
```

Vea cómo queda esta página con sus estilos aplicados:



## Página CV

El diseño de esta página es uno de los más sencillos. Solo tiene una división de 300 píxeles de alto y un texto alineado al centro de manera horizontal y vertical.

Además, el enlace del texto no está subrayado, tendrá el color darkred y cuando pase el ratón por encima, cambiará a fuchsia.

El código CSS para esta página es el siguiente:

```
#cv {  
text-align: center;  
min-height: 300px;
```

```
background-color: white;  
}  
  
#cv div{
```

```
display: table;  
min-height: 300px;  
width: 100%;  
}
```

```
#cv div p{  
display: table-cell;  
vertical-align: middle;  
}
```

```
#cv a{  
font-weight: bold;  
color: darkred;  
text-decoration: none;  
}
```

```
#cv a:hover{  
color: fuchsia;  
}
```

## Página Contacto

La página de contacto es la que menos estilos aplicados tiene, tan solo, aplicar los estilos a la foto que ha añadido de la fotógrafa para que se vea a la derecha del texto. Además, le ha añadido a la foto un ancho de 200px. Un punto muy importante en esta página es que la foto deberá tener un margen izquierdo para que no se solape con el texto. Para ello, se le ha añadido un margen izquierdo de 20 píxeles. V

Vea a continuación el código CSS resultante para esta página:

```
#fotocv{  
    float: right;  
    width: 200px;  
    margin-left: 20px;  
}
```

Por último, va a ver cómo ha quedado esta última página:

# Página Fotografía

[INICIO](#) [FOTOGRAFIAS](#) [CV](#) [CONTACTO](#)

**L**orem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem, vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit, neque metus accumsan ligula, non tempor ante est ac ante.

Comentarios



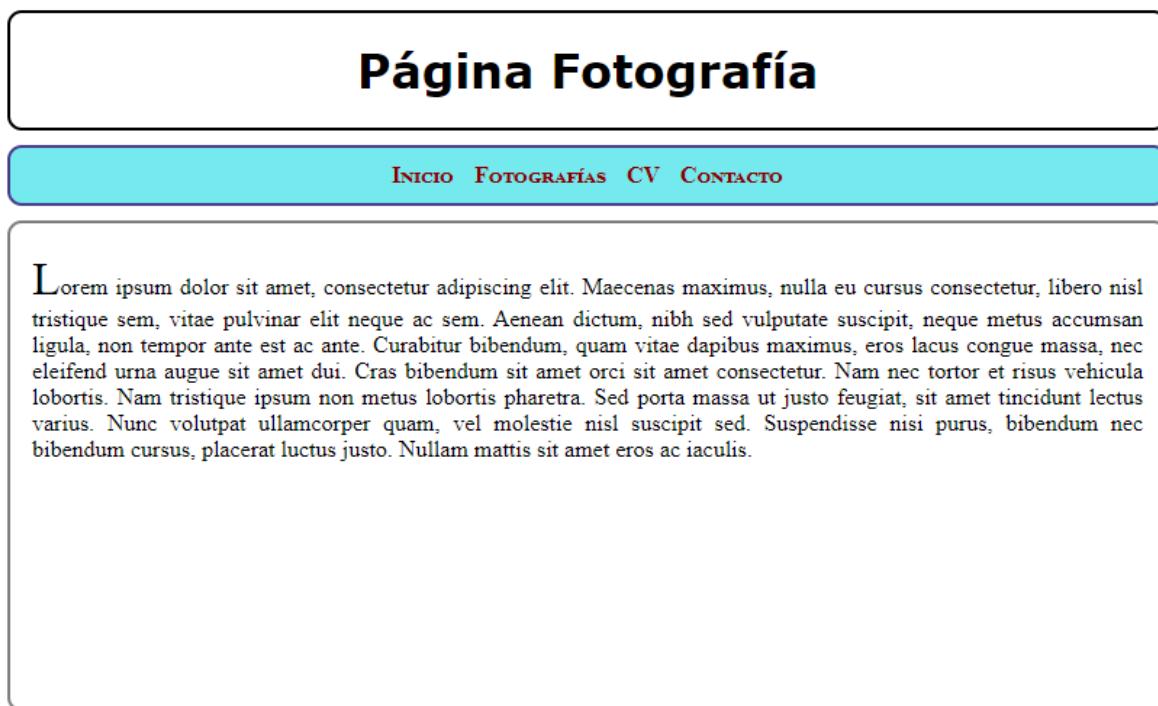
## 2.3. Identificación de la información a ubicar en la página web

En esta página web que está creando paso a paso, va a encontrar aspectos que van a ayudar al usuario a obtener información de la propia web.

Va a identificar esos elementos de manera específica para poder, posteriormente, ubicar e integrar la estos aspectos en su documento HTML.

### Página Inicio

A continuación, va a ver una captura de la página de inicio para poder ver mejor los elementos de la misma mientras los identifica:



Lo primero que puede observar son las divisiones de la página en tres capas. Las capas puede reconocerlas fácilmente, ya que, cada división tiene los bordes establecidos.

Dentro de la primera capa puede encontrar el título de la página. Dentro de esta capa, además, podría añadir un banner con alguna imagen en movimiento, o estática. Esta primera información será muy útil para el usuario, puesto que, podrá saber de qué trata la página.

Lo que encuentra seguidamente es el menú de la página donde puede ver cuatro enlaces que permitirán navegar a través del sitio web.

Por último, encuentra la tercera capa que contendrá el contenido principal de cada una de las páginas que forman el sitio web. Esta capa tiene 300 píxeles de largo, en esta capa puede incluir un

texto de bienvenida para que el usuario conozca de manera más específica los aspectos del autor de la web, en qué consiste su trabajo y qué es lo que va a encontrar en la misma.

Como puede observar, ese texto no está redactado aún, en su lugar encuentra un texto por defecto. Este tipo de texto se suele añadir al final del desarrollo web.

## Página Fotografías

Vea la siguiente imagen de la página Fotografías donde puede ver los diferentes elementos que la componen:



En esta página, puede observar que lo único que cambia es la tercera capa de la misma. En esta capa puede ver una tabla que contiene las fotografías del autor de la web.

Para crear este documento, ha tenido que echar mano de un banco de imágenes gratuitas en Internet. Hay multitud de páginas que ofrecen este tipo de imágenes sin copyright.

Para hacer las miniaturas de las imágenes, ha tenido que redimensionar las imágenes antes de insertarlas en la tabla, todo ello desde un editor de imágenes.

Por otro lado, las miniaturas tendrán las dimensiones 100X80 píxeles, con una resolución de 96ppp. Cuando haya editado las imágenes, ya estarán listas para ser utilizadas en la web.

CV

Vea antes la captura de la página CV para poder observar detenidamente todos sus elementos:

# Página Fotografía

INICIO FOTOGRAFIAS CV CONTACTO

Pulse aquí para descargar CV

En esta página, vea un fondo blanco y un enlace donde si pulsa puede descargar el CV de la persona.

El enlace cambiará de color si se posiciona el ratón encima de él. La altura mínima de esta sección está establecida en 300 píxeles.

## Página Contacto

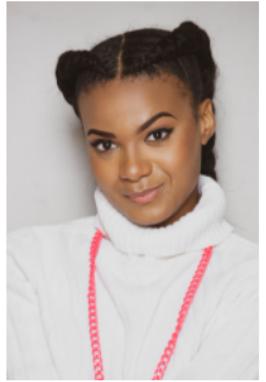
A continuación, vea la página de contacto con todos sus elementos:

# Página Fotografía

INICIO FOTOGRAFIAS CV CONTACTO

**L**orem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem, vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit, neque metus accumsan ligula, non tempor ante est ac ante.

Comentarios



En esta página, encuentra elementos que invita al usuario a contactar con el autor. Este formulario enviará el mensaje que escriba el usuario al correo electrónico del autor. Además, tiene una imagen del autor para que pueda conocerlo.

El texto que invitará al usuario a enviar un mensaje al autor todavía es un texto por defecto y se rellenará al finalizar el diseño de la web.

El textarea del formulario de contacto consta de 6 filas y 30 columnas predefinidas en HTML.

Por otro lado, la foto del autor se ubica a la derecha de la página con un margen izquierdo para que no se solape con el texto.

A continuación, vea los elementos de la página web que están codificados en código HTML. Empieza con el documento de inicio main.html:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>

<div class="general">

<div class="titulo">Página Fotografía</div>

<div class="menu">

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>
```

</div>

<div class="contenido">

<span id="capitular">L</span>orem ipsum dolor sit amet, consectetur adipiscing elit.

Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem,

vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit,

neque metus accumsan ligula, non tempor ante est ac ante.

Curabitur bibendum, quam vitae dapibus maximus, eros lacus congue massa,

nec eleifend urna augue sit amet dui. Cras bibendum sit amet orci sit amet consectetur.

Nam nec tortor et risus vehicula lobortis. Nam tristique ipsum non metus lobortis pharetra.

Sed porta massa ut justo feugiat, sit amet tincidunt lectus varius.

Nunc volutpat ullamcorper quam, vel molestie nisl suscipit sed.

Suspendisse nisi purus, bibendum nec bibendum cursus, placerat luctus justo.

Nullam mattis sit amet eros ac iaculis.

</div>

</div>

</body>

</html>

Este es el documento main.html que compone la página de inicio de su web. Como puede observar, ya le ha aplicado algunos estilos en las partes del documento deseadas. Va a analizar el documento por partes para que pueda diferenciar mejor la información que ha insertado en este documento main.html.

Primeramente, puede observar la cabecera de este documento:

<head>

```
<meta charset="utf-8" />  
  
<title>Página fotógrafo</title>  
  
<link rel="stylesheet" type="text/css" href="css/estilo.css">  
  
</head>
```

Como puede observar, en esta parte del documento tiene el tipo de documento, el título de la página y el enlace a la hoja de estilos CSS.

Lo segundo que encuentra es el cuerpo del documento. En el cuerpo del documento se encuentra la primera capa que es la que contiene todos los elementos visibles de la web. Lo siguiente que verá es la capa que contiene el título de la web. Vea:

```
<body>  
  
<div class="general">  
  
<div class="titulo">Página Fotografía</div>
```

Lo tercero que se encuentra es la capa que contiene el menú de la web que es mediante el cual el usuario podrá navegar por la página:

```
<div class="menu">  
  
<a href="main.html">Inicio</a>  
  
<a href="fotos.html">Fotografías</a>  
  
<a href="cv.html">CV</a>  
  
<a href="contacto.html">Contacto</a>  
  
</div>
```

Por último, verá la capa donde se encuentra ubicado el contenido específico de esta página. Además, verá el id de la primera letra del texto el cual le dará un estilo específico a la misma. Lo último que verá es que la capa se cierra y la capa general también, aquí se terminará el documento HTML:

```
<div class="contenido">  
  
<span id="capitular">L</span>orem ipsum dolor sit amet, consectetur adipiscing elit.  
  
Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem,
```

vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit,  
neque metus accumsan ligula, non tempor ante est ac ante.  
  
Curabitur bibendum, quam vitae dapibus maximus, eros lacus congue massa,  
nec eleifend urna augue sit amet dui. Cras bibendum sit amet orci sit amet consectetur.  
  
Nam nec tortor et risus vehicula lobortis. Nam tristique ipsum non metus lobortis pharetra.  
  
Sed porta massa ut justo feugiat, sit amet tincidunt lectus varius.  
  
Nunc volutpat ullamcorper quam, vel molestie nisl suscipit sed.  
  
Suspendisse nisi purus, bibendum nec bibendum cursus, placerat luctus justo.  
  
Nullam mattis sit amet eros ac iaculis.

</div>

</div>

</body>

</html>

Vea a continuación el código HTML que se encuentra en la página fotos.html:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>
```

```
<div class="general">

<div class="titulo">Página fotografía</div>

<div class="menu">

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<div class="contenido">

<table>

<tr>

<td><a href="imagenes/foto1.jpg" title="Imagen 1"></a></td>

<td><a href="imagenes/foto2.jpg" title="Imagen 2"></a></td>

<td><a href="imagenes/foto3.jpg" title="Imagen 3"></a></td>

<td><a href="imagenes/foto4.jpg" title="Imagen 4"></a></td>

<td><a href="imagenes/foto5.jpg" title="Imagen 5"></a></td>

</tr>

<tr>
```

```
<td><a href="imagenes/foto6.jpg" title="imagen 6"></a></td>

<td><a href="imagenes/foto7.jpg" title="imagen 7"></a></td>

<td><a href="imagenes/foto8.jpg" title="imagen 8"></a></td>

<td><a href="imagenes/foto9.jpg" title="imagen 9"></a></td>

<td><a href="imagenes/foto10.jpg" title="imagen 10"></a></td>

</tr>

</table>

</div>

</div>

</body>

</html>
```

Ahora, va a analizarlo por partes al igual que ha hecho con el documento anterior.

La primera parte de este documento es la misma que la del documento anterior. Es el comienzo del documento y contiene el título, el menú y toda la parte visible del mismo:

```
<!DOCTYPE html>

<html>
  <head>
    <meta charset="utf-8" />
    <title>Página fotógrafo</title>
    <link rel="stylesheet" type="text/css" href="css/estilo.css">
  </head>
```

```
<body>

<div class="general">

<div class="titulo">Página fotografía</div>

<div class="menu">

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>
```

En la tercera capa, que es donde está el contenido específico de cada página, puede observar que se compone de una tabla de dos filas y cinco columnas donde en cada una de las celdas hay una pequeña imagen de 100X80 píxeles, a las que se les ha añadido un título con el número de imagen y un borde con un grosor de 1 pixel.

Por último, finalizan todas las capas y termina el documento HTML:

```
<div class="contenido">

<table>

<tr>

<td><a href="imagenes/foto1.jpg" title="Imagen 1"></a></td>

<td><a href="imagenes/foto2.jpg" title="Imagen 2"></a></td>

<td><a href="imagenes/foto3.jpg" title="Imagen 3"></a></td>

<td><a href="imagenes/foto4.jpg" title="Imagen 4"></a></td>
```

```
<td><a href="imagenes/foto5.jpg" title="imagen 5"></a></td>

</tr>

<tr>

<td><a href="imagenes/foto6.jpg" title="imagen 6"></a></td>

<td><a href="imagenes/foto7.jpg" title="imagen 7"></a></td>

<td><a href="imagenes/foto8.jpg" title="imagen 8"></a></td>

<td><a href="imagenes/foto9.jpg" title="imagen 9"></a></td>

<td><a href="imagenes/foto10.jpg" title="imagen 10"></a></td>

</tr>

</table>

</div>

</div>

</body>

</html>
```

A continuación, va a ver el código de la estructura del documento cv.html:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>
```

```
<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>

    <div class="general">

        <div class="titulo">Página Fotografía</div>

        <div class="menu">

            <a href="main.html">Inicio</a>

            <a href="fotos.html">Fotografías</a>

            <a href="cv.html">CV</a>

            <a href="contacto.html">Contacto</a>

        </div>

        <div class="contenido">

            <div id="cv"><p><a href="documentos/cv.pdf" target="_blank">Pulse aquí para descargar CV</a></p></div>

        </div>

    </div>

</body>

</html>
```

Al igual que los documentos que ha visto hasta ahora, el documento cv.html también tiene un encabezado de página, la primera capa es la que contiene todos los elementos visibles del documento y las dos primeras capas se ubican dentro de la primera, las cuales contienen el título y el menú:

```
<!DOCTYPE html>
```

```
<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>

<div class="general">

    <div class="titulo">Página Fotografía</div>

    <div class="menu">

        <a href="main.html">Inicio</a>

        <a href="fotos.html">Fotografías</a>

        <a href="cv.html">CV</a>

        <a href="contacto.html">Contacto</a>

    </div>

</div>
```

En la tercera capa es donde se añade un enlace para descargar el documento PDF. Este contenido tiene sus estilos establecidos en la hoja de estilos y, como puede observar, tiene estilo para los atributos id y, además, para el enlace con la clase contenido. Posteriormente, se finalizan las capas: general, las capas de contenido y la de cv. Por último, se finaliza la división que hay dentro del párrafo y el documento en general.

Vea el código:

```
<div class="contenido">

    <div id="cv"><p><a href="documentos/cv.pdf" target="_blank">Pulse aquí para descargar CV</a></p></div>

</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

Lo siguiente que va a ver es el código HTML de la página de contacto.html, de manera que pueda identificar los elementos en la web:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página Fotógrafo</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>

<div class="general">

    <div class="titulo">Página Fotografía</div>

    <div class="menu">

        <a href="main.html">Inicio</a>

        <a href="fotos.html">Fotografías</a>

        <a href="cv.html">CV</a>

        <a href="contacto.html">Contacto</a>

    </div>

    <div class="contenido">
```

```
<div>

</div>

<span id="capitular">L</span>orem ipsum dolor sit amet, consectetur adipiscing elit.

    Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem,
    vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit,
    neque metus accumsan ligula, non tempor ante est ac ante.

<form method="post" action="mailto:correo@dominio.com" enctype="text/plain">

    <p>Comentarios</p>

    <textarea rows="6" cols="30" name="comentarios"></textarea>

    <p><button type="submit">Enviar</button>&nbsp; &nbsp;
        <button type="reset">Cancelar</button></p>

</form>

</div>

</div>

</body>

</html>
```

Lo mismo que ocurre en el resto de las páginas web que componen el sitio web, este documento también tiene las mismas partes al principio, es decir, el encabezado de la página y el cuerpo del documento que porta todos los elementos visibles, el cual se divide en dos capas, la primera tiene el título de la página y la segunda el menú de la misma:

```
<!DOCTYPE html>

<html>
```

```
<head>

<meta charset="utf-8" />

<title>Página Fotógrafo</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>

<div class="general">

    <div class="titulo">Página Fotografía</div>

    <div class="menu">

        <a href="main.html">Inicio</a>

        <a href="fotos.html">Fotografías</a>

        <a href="cv.html">CV</a>

        <a href="contacto.html">Contacto</a>

    </div>
```

Lo siguiente que va a ver es la parte del documento que tiene una división llamada fotocv, la cual tiene una foto del autor de la página y tiene un estilo de clase y de id, un texto con una letra capitular que también utiliza un estilo id y, por último, tiene un formulario para enviar un mensaje al correo del autor de la página. Lo último que encuentra es el cierre de los todos los divs del documento. Vea el código:

```
<div class="contenido">

    <div>

    </div>

    <span id="capitular">L</span>orem ipsum dolor sit amet, consectetur adipiscing elit.
```

Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem,  
vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit,  
neque metus accumsan ligula, non tempor ante est ac ante.

```
<form method="post" action="mailto:correo@dominio.com" enctype="text/plain">  
<p>Comentarios</p>  
  
<textarea rows="6" cols="30" name="comentarios"></textarea>  
  
<p><button type="submit">Enviar</button>&nbsp; &nbsp;  
 <button type="reset">Cancelar</button></p>  
  
</form>  
  
</div>  
  
</div>  
  
</body>  
  
</html>
```

## 2.4. Selección de contenidos para cada elemento de la página

Como ya ha visto, elegir los contenidos para una página web no es algo sencillo. Hay veces que tiene tanta información entre la que elegir que no sabe cómo hacer un filtro para que no acabe exponiendo demasiada información en la que el usuario pueda llegar a confundirse.

Lo siguiente que va a hacer es especificar los contenidos que ha añadido a la página que está creando y dar una razón de porqué ha elegido este tipo de elementos y no otros.

### Página Inicio

Vea mediante su código HTML el contenido que ha añadido para el documento main.html:

```
<div class="contenido">
```

```
<span id="capitular">L</span>orem ipsum dolor sit amet, consectetur adipiscing elit.  
  
Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem,  
vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit,  
neque metus accumsan ligula, non tempor ante est ac ante.  
  
Curabitur bibendum, quam vitae dapibus maximus, eros lacus congue massa,  
nec eleifend urna augue sit amet dui. Cras bibendum sit amet orci sit amet consectetur.  
  
Nam nec tortor et risus vehicula lobortis. Nam tristique ipsum non metus lobortis pharetra.  
  
Sed porta massa ut justo feugiat, sit amet tincidunt lectus varius.  
  
Nunc volutpat ullamcorper quam, vel molestie nisl suscipit sed.  
  
Suspendisse nisi purus, bibendum nec bibendum cursus, placerat luctus justo.  
  
Nullam mattis sit amet eros ac iaculis.  
  
</div>  
  
</div>  
  
</body>  
  
</html>
```

Como puede observar en esta página, se ha decantado por mostrar un texto de presentación. Pare realizar esta página por ahora ha seleccionado un texto por defecto, el típico Lorem Ipsum que puede encontrar en cualquier generador de textos por defecto.

Ahora, llegados a este punto, debe introducir el texto real que llevará esta página. Por lo tanto, deberá redactar un texto de presentación de la misma. Debe recordar que este texto va a ser el primer texto con el que el usuario se va a topar cuando entre a la web, por lo tanto, debe ser lo más atractivo posible para el usuario y fácil de leer.

Un texto que podría añadir a su página de inicio sería el siguiente:

```
<div class="contenido">  
  
<span id="capitular">L</span>es invito a visitar mi página web.
```

Soy una fotógrafa profesional que me dedico a retratar paisajes naturales que encuentro en los viajes que realizo.

<p>En esta página podrá ver todos mis trabajos. Para ello, solo tiene que pulsar el enlace Fotografías que se ubica en el menú de la página.

</p>

<p>Espero que lo disfruten.</p>

</div>

## Página Fotografías

En fotos.html ha escogido las fotos de paisajes del autor y las ha añadido a una tabla para que queden mejor estructuradas. La sintaxis que ha utilizado para ello es la siguiente:

```
<div class="contenido">

<table>

  <tr>

    <td><a href="imagenes/foto1.jpg" title="imagen 1"></a></td>

    <td><a href="imagenes/foto2.jpg" title="imagen 2"></a></td>

    <td><a href="imagenes/foto3.jpg" title="imagen 3"></a></td>

    <td><a href="imagenes/foto4.jpg" title="imagen 4"></a></td>

    <td><a href="imagenes/foto5.jpg" title="imagen 5"></a></td>

  </tr>

  <tr>

    <td><a href="imagenes/foto6.jpg" title="imagen 6"></a></td>
```

```
<td><a href="imagenes/foto7.jpg" title="imagen 7"></a></td>

<td><a href="imagenes/foto8.jpg" title="imagen 8"></a></td>

<td><a href="imagenes/foto9.jpg" title="imagen 9"></a></td>

<td><a href="imagenes/foto10.jpg" title="imagen 10"></a></td>

</tr>

</table>

</div>
```

Para crear esta página ficticia, ha tenido que buscar imágenes libres de derechos de autor que puede encontrar en diversas páginas de Internet. Cuando las haya descargado, las tiene que editar para redimensionarlas sin que pierdan calidad y, posteriormente, añadirlas a la tabla.

Debe recordar añadirle el atributo title a cada una de ellas. Para mejorar la navegabilidad del usuario, ha hecho que cuando el usuario pulse sobre la imagen, ésta se abra en otra ventana para que pueda verla a tamaño real, para ello, ha utilizado el valor \_blank. De esta manera, fuerza al usuario a permanecer en la página cuando abra las imágenes.

A continuación, vea uno de los <td> para poder observar mejor el contenido que ha añadido en la tabla y sus características:

```
<td><a href="imagenes/foto1.jpg" title="imagen 1"></a></td>
```

Como puede observar, los elementos que ha utilizado en la etiqueta <td> son los siguientes:

- Border: es el elemento que establece el borde de la imagen.
- A href: se utiliza para enlazar la imagen.
- Height: define la altura de la imagen.
- Width: define el ancho de la imagen.
- Title: Define el título de la imagen.
- Class: Este será el nombre de clase que utilizará CSS para aplicar los estilos.
- Target: Se utiliza para que el vínculo de la imagen se abra del modo que desee.
- Img src: Se definirá la ruta a la propia imagen.
-

## Página CV

En esta página, tan solo, ha añadido un texto con un enlace que permitirá a los usuarios descargar el CV de la fotógrafo. Se ha elegido este tipo de contenido porque es el más sencillo. La sintaxis que ha utilizado es la siguiente:

```
<div class="contenido">  
    <div id="cv"><p><a href="documentos/cv.pdf" target="_blank">Pulse aquí para descargar CV</a></p></div>  
</div>  
</div>
```

El documento cv.html es uno de los más básicos. Solo tiene un pequeño texto donde puede encontrar un enlace para descargar el cv en formato PDF.

Los elementos que ha utilizado para crear el contenido de este documento son:

- Target: con el valor \_blank para que se abra en una nueva ventana.
- A href: el elemento de enlazar el archivo pdf.

Muchas veces, también puede encontrar los cv en pantalla para que el usuario pueda visualizarlos sin necesidad de descargarlo. Sin embargo, puede ser algo poco atractivo y, además, no se recomienda este uso debido a la ley de protección de datos. Al insertar un enlace para descargar el cv, asegurarse de que la persona que lo pulse es porque realmente está interesada en el propio documento.

En estos casos, se recomienda no poner en el CV los datos personales como el DNI o la dirección de casa.

## Página Contacto

En el documento contacto.html ha introducido elementos como un pequeño texto, una foto personal y un formulario de contacto. Vea el código del documento:

```
<div class="contenido">  
    <div>  
          
    </div>  
    <span id="capitular">L</span>orem ipsum dolor sit amet, consectetur adipiscing elit.
```

Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem,  
vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit,  
neque metus accumsan ligula, non tempor ante est ac ante.

```
<form method="post" action="mailto:correo@dominio.com" enctype="text/plain">  
<p>Comentarios</p>  
  
<textarea rows="6" cols="30" name="comentarios"></textarea>  
  
<p><button type="submit">Enviar</button>&nbsp; &nbsp;  
 <button type="reset">Cancelar</button></p>  
  
</form>  
  
</div>
```

Como puede observar, lo primero que ha utilizado es una capa para añadir la imagen personal, después se ha introducido el texto, el cual queda a la izquierda de la imagen y, por último, se ha añadido el formulario de contacto.

Estos son los elementos básicos que se suelen utilizar en las páginas de contacto. El formulario es uno de los elementos más importantes, ya que, gracias a él los usuarios podrán contactar a través de la web sin tener que añadir teléfonos, ni correos electrónicos personales.

Como puede observar, se ha añadido un texto auxiliar que va a proceder a modificar para mostrar cómo quedaría esta página terminada:

```
<span id="capitular">M</span>uchas gracias por su visita.
```

Puede ponerse: en contacto conmigo a través de este formulario para decirme qué le ha parecido mi trabajo o si tiene alguna duda que yo le pueda solucionar.

## 2.5. Utilización del documento funcional para las especificaciones del diseño

Como ya vió anteriormente, se ha utilizado un documento HTML básico en el que ha aplicado los estilos deseados para obtener un resultado distinto de página web.

---

En este apartado, va a explicar cómo se ha utilizado el documento funcional y cómo se le han aplicado los estilos específicos que ha do relatando.

Va a empezar por el documento main.html:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>

</head>

<body>

<div><h1>Página web</h1></div>

<div>

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<div>

<span>L</span>orem ipsum dolor sit amet, consectetur adipiscing elit.

Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem,
vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit,
```

neque metus accumsan ligula, non tempor ante est ac ante.

Curabitur bibendum, quam vitae dapibus maximus, eros lacus congue massa,

nec eleifend urna augue sit amet dui. Cras bibendum sit amet orci sit amet consectetur.

Nam nec tortor et risus vehicula lobortis. Nam tristique ipsum non metus lobortis pharetra.

Sed porta massa ut justo feugiat, sit amet tincidunt lectus varius.

Nunc volutpat ullamcorper quam, vel molestie nisl suscipit sed.

Suspendisse nisi purus, bibendum nec bibendum cursus, placerat luctus justo.

Nullam mattis sit amet eros ac iaculis.

</div>

</div>

</body>

</html>

Como puede observar, el documento original no tiene ningún tipo de estilo, tan solo, las divisiones pertinentes.

Vea qué estilos se les ha aplicado a los diferentes elementos de main.html:

Para ello, va a ver cómo queda el documento final al habersele aplicado el nombre de los estilos para enlazarlos con la hoja de estilos. De esta manera podrá comparar ambos documentos:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">
```

```
</head>

<body>

<div class="general">

<div class="titulo">Página Fotografía</div>

<div class="menu">

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<div class="contenido">

<span id="capitular">L</span>es invito a visitar mi página web.

Soy una fotógrafo profesional que me dedico a retratar paisajes naturales que encuentro

en los viajes que realizo.

<p>En esta página podrá ver todos mis trabajos. Para ello, solo tiene que pulsar

el enlace Fotografías que se ubica en el menú de la página.

</p>

<p>Espero que lo disfruten.</p>

</div>

</div>
```

```
</body>
```

```
</html>
```

Como puede ver, lo primero que se le ha aplicado es un estilo en el div general. En cambio, en la hoja de estilos se ha aplicado al principio estilos al body para crear un estilo general en la página. Vea los estilos:

```
body{  
background-color: white;  
}  
}
```

El único estilo que se le ha aplicado al body es el background-color que lo que hace es que la página tenga un fondo de color blanco.

A continuación, vea los estilos que se les ha aplicado a la capa general del documento HTML:

```
.general{  
width: 800px;  
margin: 0 auto;  
text-align: center;  
margin-top: 100px;  
border-color: white;  
border-radius: 10px;  
background-color: white;  
}  
}
```

Los estilos que ha aplicado a esta capa son los siguientes:

- Width: especifica el ancho de la capa.
- Margin: especifica el margen de la capa.
- Text-align: indica que le texto se alinee en el centro.
- Margin-top: establece el número de píxeles para el margen superior.
- Border color: establece el color del borde.

- Border-radius: Redondea las esquinas al indicar los píxeles para hacerlo. Es un nuevo estilo de CSS3.
- Background-color: establece el color de fondo de la capa.

Lo siguiente que va a explicar es sobre cómo hacer las esquinas redondeadas. Cosa que en CSS3 es muy sencilla de hacer, sobre todo cuando haya experimentado con las dimensiones de las curvas.

Una ventaja de utilizar la técnica de CSS3 es que puede hacer estas esquinas, tan solo, utilizando CSS, por lo que no necesitará ayuda de HTML, ni tampoco ningún fondo de color que redondee el elemento, como se hacía antiguamente.

La desventaja principal de este método es que puede que encuentre navegadores que no sean compatibles.

Para poder crear el efecto de esquina redondeada se usa lo siguiente:

- Border-radius: 20px;

Sin embargo, si lo que quiere es que sus esquinas tengan forma de óvalo, se usan dos valores. Vea un ejemplo:

- Border-radius: 20px / 50px;

De esta manera, los arcos en vez de circulares, serán elípticos, ya que, utilizará los dos valores que le ha dado a la propiedad. En caso de no poner las barras diagonales, las esquinas seguirán teniendo tamaño circular, pero cada una de ellas tendrá tamaños diferentes. Esto ocurre porque las esquinas tendrán un radio desigual.

Para los navegadores que aún son incompatibles con este método, lo que debe hacer es añadir un prefijo específico y, además, dejar la propiedad sin prefijo para que los navegadores que sean compatibles puedan interpretarla. Vea un ejemplo:

-moz-border-radius: 20px;

-webkit-border-radius: 20px;

- Border-radius: 20px;

En cambio, estos navegadores aceptan propiedades que permiten que establezca cada esquina de manera individual. Esto puede hacerlo, por ejemplo, con webkit que, aunque no acepta valores dobles, sí acepta que establezca las esquinas de manera individual. Vea un ejemplo:

-webkit-border-radius: 20px;

-webkit-border-top-right-radius: 60px;

```
-webkit-border-bottom-left-radius: 60px;  
  
-moz-border-radius: 20px 60px;  
  
• Border-radius: 20px 60px;
```

Vea el siguiente estilo que pertenece a la clase `titulo` y se establece en el documento `main.html`. Este estilo es el que establece los estilos de la primera capa y que el usuario puede ver en la página, el título de la página pone Página Fotografía. Vea el CSS para estos estilos:

```
.titulo{  
  
font-size: xx-large;  
  
font-family: Verdana, Geneva, Tahoma, sans-serif;  
  
font-weight: bold;  
  
color: black;  
  
background-color: white;  
  
padding-top: 20px;  
  
padding-bottom: 20px;  
  
border-width: 2px;  
  
border-style: solid;  
  
border-color: black;  
  
border: 2px solid black;  
  
border-radius: 10px;  
}
```

Ahora, va a analizar uno por uno los estilos que ha establecido para el título de la página:

- Font-size: Añade la fuente extra grande para este título.
- Font-family: Añade la fuente Verdana como la principal para el título.
- Font-weight: Le ha añadido al texto el valor negrita.
- Color: Es donde ha definido el color del título.
- Background-color: Ha utilizado esta característica para fijar el color de fondo del título.

- Padding-bottom y padding-top: Esta característica es la que utiliza para determinar la anchura del relleno para el elemento. En este caso en su parte superior e inferior.
- Border-style: Lo utiliza para establecer los bordes del elemento.
- Border-width: Establece el grosor de los bordes.
- Border-color: Establece el color de los bordes.
- Border-radius: redondea las esquinas del borde.

Vea a continuación los estilos que se han establecido en las hojas de estilo para el documento main.html donde se encuentra el menú. Vea el código a continuación:

```
.menu{  
background-color: rgb(116, 234, 238);  
  
margin-top: 10px;  
  
margin-bottom: 10px;  
  
border-style: solid;  
  
border-color: darkslateblue;  
  
border-width: 2px;  
  
padding-top: 10px;  
  
padding-bottom: 10px;  
  
font-weight: bold;  
  
font-variant: small-caps;  
  
word-spacing: 10px;  
  
border-radius: 10px;  
}
```

Ahora, toca ver uno por uno los estilos que ha establecido para el menú:

- Background-color: Se ha establecido como fondo del menú un color turquesa.
- Margin-top y margin-bottom: establece un valor específico para los márgenes superior e inferior.
- Border-style: Para añadir un estilo al borde.
- Border-color: Para añadir un color al borde.

- Border-width: Para cambiar el grosor del borde.
- Padding-top y padding-bottom: Para añadir un relleno superior e inferior.
- Font-weight: Para añadir grosor a la letra.
- Font-variant: Mediante esta característica se puede añadir una variante del tipo de letra como, por ejemplo, versalitas.
- Word-spacing: Para añadir un espaciado entre cada una de las palabras mediante valores en píxeles.

Por otra parte, también se encuentra con los estilos a los enlaces del menú. Vea en el siguiente código CSS:

```
.menu a{  
color: darkred;  
text-decoration: none;  
}  
  
.menu a:hover{  
color: fuchsia;  
}  
  
.menu a:visited{  
color: blueviolet;  
}
```

Para los enlaces del menú, se han establecido los siguientes estilos:

- .menu a: En esta sección es donde se añade un color determinado al texto y que no esté subrayado por defecto.
- .menu a:hover: Es donde define bel color determinado para cuando pase el puntero del ratón sobre los enlaces éstos cambien de color.
- .menu a:visited: Esta sección de código se ha definido para que cambie el color a los enlaces ya visitados.

Otro estilo que puede encontrar en el documento main.html es la diferenciación de los contenidos. Concretamente, en la página de inicio, tan solo, se ha añadido un texto de bienvenida. Sin embargo, éste también tiene un estilo propio en la hoja de estilos. Vea el código CSS:

```
.contenido {  
    position: relative;  
    padding-top: 20px;  
    padding-right: 15px;  
    padding-bottom: 15px;  
    padding-left: 15px;  
    border-style: solid;  
    border-color: gray;  
    border-width: 2px;  
    text-align: justify;  
    min-height: 300px;  
    border-radius: 10px;  
}
```

Vea a continuación cada uno de los estilos paso por paso:

- Position: Esta característica se utiliza para especificar la ubicación de los elementos dentro de un div.
- Padding-left: Se utiliza para especificar la anchura del relleno de un elemento por su parte izquierda.
- Padding-right: Se utiliza para especificar la anchura del relleno de un elemento por su parte derecha.
- Padding-top: Se utiliza para especificar la anchura del relleno de un elemento por su parte superior.
- Padding-bottom: Se utiliza para especificar la anchura del relleno de un elemento por su parte inferior.
- Border-style: Se utiliza para añadir un estilo al borde.
- Border-color: Establece el color del borde.
- Border-width: Establece el grosor del borde.

- Text-align: Establece la alineación del texto
- Min-height: Se utiliza para definir una altura mínima del div.

Hasta aquí llegan los estilos que ha definido para la página main.html. Cuando los haya añadido al documento CSS, podrá ver el resultado de cómo han quedado los elementos con sus estilos.

Los estilos de los próximos documentos son idénticos en sus partes comunes. Las partes comunes de los documentos son las siguientes:

- Body
- Estilos para la capa general
- Estilos para la capa título
- Estilos para la capa menú
- Estilos para la capa contenido
- El último estilo que puede encontrar en el documento CSS es el estilo del inicio de la letra capitular de los párrafos. El código es el siguiente:

```
#capitular{
```

```
    font-size: xx-large;
```

```
}
```

Gracias a este pequeño código, puede hacer que la primera letra del párrafo de tamaño extra grande.

Además, para los estilos de los documentos web restantes, se ha creado una capa específica que irá cambiando conforme al documento que sea. Va a ver cada una de las secciones que los diferencia:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>

<div>
```

```
<div><h1>Página web</h1></div>
```

```
<div>
```

```
  <a href="main.html">Inicio</a>  
  
  <a href="fotos.html">Fotografías</a>  
  
  <a href="cv.html">CV</a>  
  
  <a href="contacto.html">Contacto</a>
```

```
</div>
```

```
<div>
```

```
  <table>
```

```
    <tr>
```

```
      <td><a href="imagenes/foto1.jpg" title="Imagen 1"></a></td>
```

```
      <td><a href="imagenes/foto2.jpg" title="Imagen 2"></a></td>
```

```
      <td><a href="imagenes/foto3.jpg" title="Imagen 3"></a></td>
```

```
      <td><a href="imagenes/foto4.jpg" title="Imagen 4"></a></td>
```

```
      <td><a href="imagenes/foto5.jpg" title="Imagen 5"></a></td>
```

```
    </tr>
```

```
  <tr>
```

```
    <td><a href="imagenes/foto6.jpg" title="Imagen 6"></a></td>
```

```
<td><a href="imagenes/foto7.jpg" title="imagen 7"></a></td>

<td><a href="imagenes/foto8.jpg" title="imagen 8"></a></td>

<td><a href="imagenes/foto9.jpg" title="imagen 9"></a></td>

<td><a href="imagenes/foto10.jpg" title="imagen 10"></a></td>

</tr>

</table>

</div>

</div>

</body>

</html>
```

A continuación, va a ver el documento con los estilos aplicados:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>

<div class="general">

<div class="titulo">Página fotografía</div>
```

```
<div class="menu">

    <a href="main.html">Inicio</a>

    <a href="fotos.html">Fotografías</a>

    <a href="cv.html">CV</a>

    <a href="contacto.html">Contacto</a>

</div>

<div class="contenido">

<table>

    <tr>

        <td><a href="imagenes/foto1.jpg" title="Imagen 1"></a></td>

        <td><a href="imagenes/foto2.jpg" title="Imagen 2"></a></td>

        <td><a href="imagenes/foto3.jpg" title="Imagen 3"></a></td>

        <td><a href="imagenes/foto4.jpg" title="Imagen 4"></a></td>

        <td><a href="imagenes/foto5.jpg" title="Imagen 5"></a></td>

    </tr>

    <tr>

        <td><a href="imagenes/foto6.jpg" title="Imagen 6"></a></td>

        <td><a href="imagenes/foto7.jpg" title="Imagen 7"></a></td>

    </tr>

</div>
```

```
<td><a href="imagenes/foto8.jpg" title="imagen 8"></a></td>

<td><a href="imagenes/foto9.jpg" title="imagen 9"></a></td>

<td><a href="imagenes/foto10.jpg" title="imagen 10"></a></td>

</tr>

</table>

</div>

</div>

</body>

</html>
```

Como puede observar, en este documento están definidos los estilos de cada uno de los elementos que hay en él de manera específica. En cambio, aunque no se ha definido los estilos en los elementos, sí que se ha definido el estilo de la tabla.

Vea a continuación, los estilos que ha aplicado a la tabla del documento:

```
table{

width: 70%;

margin-left: auto;

margin-right: auto;

height: 300px;

border-style: none;

}
```

Los estilos que se han especificado específicamente para la tabla son los siguientes:

- Width: especifica el ancho de la tabla.
- Margin-left: especifica los márgenes en el lado izquierdo.
- Margin-right: especifica los márgenes en el lado derecho.

- Height: especifica la altura en píxeles.
- Border-style: especifica los bordes, pero en este caso el valor es none, por lo que, no tendrá ninguno.

Por otra parte, también se han establecido unos estilos específicamente para las celdas de la tabla.

Vea el código CSS:

```
td{  
    width: 30%;  
  
    margin-left: auto;  
  
    margin-right: auto;  
}
```

A continuación, vea las propiedades de manera específica:

Width: Es lo que establece el ancho de la celda.

Margin-left: Establece el margen izquierdo.

Margin-right: Establece el margen derecho.

Además, se han aplicado estilos a las imágenes contenidas en el archivo fotos.html. Vea a continuación:

```
.foto{  
    border-width: 1px;  
  
    border-color: limegreen;  
  
    border-style: solid;  
}
```

Estos estilos que ve son los siguientes:

- Border-width: Establece el grosor del borde.
- Border-color: Establece el color del borde.
- Border-style: establece el estilo del borde para que sea visible o no, en la página.

El siguiente documento que va a ver es el de CV. Este documento sin estilos aplicados se conforma con el siguiente código HTML:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>

</head>

<body>

<div><h1>Página web</h1></div>

<div>

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<div>

<div><p>Pulse aquí para descargar CV</p></div>

</div>

</div>

</body>
```

</html>

Al aplicarle los estilos, quedará como resultado el siguiente documento:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>

<div class="general">

<div class="titulo">Página Fotografía</div>

<div class="menu">

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<div class="contenido">

<div id="cv"><p><a href="documentos/cv.pdf" target="_blank">Pulse aquí para descargar CV</a></p></div>

</div>
```

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

El código CSS que se ha utilizado para dar estilo al contenido del archivo cv.html es el siguiente:

```
#cv {  
  
    text-align: center;  
  
    min-height: 300px;  
  
    background-color: white;  
  
}
```

Vea las propiedades más específicamente:

- Text-align: Se establece la alineación del texto.
- Min-height: Se establece la altura mínima de la capa.
- Background-color: Establece el color de fondo.

Además, se han aplicado otros estilos a los contenidos del documento cv.html y son los que va a ver a continuación:

```
#cv div{  
  
    display: table;  
  
    min-height: 300px;  
  
    width: 100%;  
  
}  
  
#cv div p{  
  
    display: table-cell;  
  
    vertical-align: middle;  
  
}
```

```
#cv a{  
  
    font-weight: bold;  
  
    color: darkred;  
  
    text-decoration: none;  
  
}
```

```
#cv a:hover{  
  
    color: fuchsia;  
  
}
```

```
#fotocv{  
  
    float: right;  
  
    width: 200px;  
  
    margin-left: 20px;  
  
}
```

Como puede ver, se han aplicado estilos a la capa del cv, otro al párrafo, otro al enlace y otro a la foto del cv.

Lo que ve dentro del div cv es lo siguiente:

- Display: Se ha utilizado para controlar el texto de mejor manera y permitir que el texto sea tratado como una tabla para poder centrarlo dentro de la capa.
- Min-height: Añade una altura mínima a la capa.
- Width: Se establece el ancho de la capa.
- Los estilos que se han establecido dentro del div p son los siguientes:
- Display: Esta propiedad se encarga de manipular el texto como si estuviera dentro de una celda para poder controlar la posición dentro de la capa.
- Vertical-align: Se utiliza para alinear el texto de manera vertical.

El estilo que se le ha aplicado al párrafo de manera específica es:

- Text-decoration: Se ha utilizado para quitar el subrayado por defecto que aparecen en los enlaces.
- Por último, se han añadido los siguientes estilos a la foto del cv:
- Float: Lo utiliza para posicionar la foto en el lado derecho, en este caso.
- Width: Para establecer el ancho de la foto.
- Margin-left: Para definir el espacio izquierdo, que en este caso sirve para que la foto no solape al texto.

Vea la estructura HTML del documento contacto.html:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página fotógrafo</title>

</head>

<body>

<div>

<div><h1>Página web</h1> </div>

<div>

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>
```

```
<div>
<div>
    
</div>
<span>L</span>orem ipsum dolor sit amet, consectetur adipiscing elit.
Maecenas maximus, nulla eu cursus consectetur, libero nisl tristique sem,
vitae pulvinar elit neque ac sem. Aenean dictum, nibh sed vulputate suscipit,
neque metus accumsan ligula, non tempor ante est ac ante.

<form method="post" action="mailto:correo@dominio.com" enctype="text/plain">
<p>Comentarios</p>
<textarea rows="6" cols="30" name="comentarios"></textarea>
<p><button type="submit">Enviar</button>&nbsp; &nbsp;
<button type="reset">Cancelar</button></p>
</form>
</div>
</div>
</body>
</html>
```

Una vez que haya aplicado los estilos quedará como resultado el siguiente documento:

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

```
<head>

<meta charset="utf-8" />

<title>Página Fotógrafo</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>

<div class="general">

    <div class="titulo">Página Fotografía</div>

    <div class="menu">

        <a href="main.html">Inicio</a>

        <a href="fotos.html">Fotografías</a>

        <a href="cv.html">CV</a>

        <a href="contacto.html">Contacto</a>

    </div>

    <div class="contenido">

        <div>

        </div>

        <span id="capitular">Muchas gracias por su visita. Puede ponerse en contacto conmigo a través de este formulario para decirme qué le ha parecido mi trabajo o si tiene alguna duda que yo le pueda solucionar.

    </div>


```

```
<form method="post" action="mailto:correo@dominio.com" enctype="text/plain">

<p>Comentarios</p>

<textarea rows="6" cols="30" name="comentarios"></textarea>

<p><button type="submit">Enviar</button>&nbsp; &nbsp;
   <button type="reset">Cancelar</button></p>

</form>

</div>

</div>

<div class="footer"><p><a href="mapaweb.html" target="_blank">Mapa del sitio</a></p></div>

</body>

</html>
```

## 2.6. Tipos de página para la ubicación de contenidos

Mediante código HTML puede crear muchísimos estilos de página. Que las páginas sean de un tipo u otro, dependerá de cómo van a ir posicionados los elementos. Es por esta razón que en HTML se utilizan las etiquetas `<div>` y `<span>`, para crear diferentes capas donde ubicar los elementos. Estructurar un documento HTML por capas tiene ventajas como:

- De esta manera, puede saber cuándo un usuario pulsa sobre la página y en qué contenido ha pulsado.
- Puede crear efectos especiales para cada contenedor.
- Permite mucha más flexibilidad en el diseño.

Diseño de páginas existen tantos como creatividad pueda tener un diseñador. En cambio, existen diseños básicos que se utilizan muy a menudo, puesto que, son páginas funcionales, muy sencillas y rápidas de crear.

Dependiendo del diseño que vaya a plantear en la página, se diseñará la misma. Puede encontrar diseños de páginas donde el menú se ubica a la izquierda, por lo tanto, el contenido debería ir a la

derecha. Otras veces, puede encontrarse con que el menú está debajo de la cabecera de la página, por lo que el contenido se ubicaría debajo del mismo.

Podrá aplicar un diseño u otro dependiendo del tipo de página que quiera desarrollar. Por lo tanto, en estos casos debe guiarse por la funcionalidad de la página. Siempre debe tener en cuenta que son los usuarios quienes deben encontrarse cómodos navegando por la página y encontrar lo que buscan de manera rápida y sencilla. Es por esta razón que debe tener en cuenta qué diseños de adaptan a los contenidos que expone la página.

Debe planificar y organizar el estilo dependiendo de cómo irán estructurados los contenidos. Por ello, crear una página web requiere bastante tiempo, ya que, es mejor tener muy claras las cosas antes de empezar a realizar un diseño y, a mitad de camino, tener que volver a rehacerlo todo.

Por todo ello, va a ver algunos consejos de cómo empezar a organizar el contenido y la estructura de una página web:

Lo primero que se recomienda es crear una lista con los contenidos de los que se trate la web. El tipo de tema y la cantidad de los mismos, determinarán bastante la apariencia y la estructura del sitio web. Por ejemplo, no se utilizarán los mismos colores para una página sobre medicina y otra sobre mecánica de coches.

Muchas veces se denomina a la estructura de un sitio web como la arquitectura del sitio web. Esto ocurre porque los sitios webs se construyen como si fueran un edificio, es decir, con las piezas bien estructuradas, sus accesos y su estilo.

Por ejemplo, imagine que, para ir a uno de los dormitorios de una casa, tuviera que pasar por la cocina. Esto no sería una buena arquitectura. Es por ello, que los diseñadores deben construir un sitio web dependiendo de la función que se le va a dar al propio sitio web. El objetivo principal es conseguir que el sitio web sea accesible y cómodo para los usuarios y, para ello, lo que debe utilizar es el sentido común y la lógica.

A continuación, va a ver los elementos que debe tener presentes para diseñar su sitio web. Con los siguientes elementos puede estructurar de manera más precisa la estructura de la misma y darán una visión de qué debe tener en cuenta para ello:

Debe destacar qué es lo que va a ser más importante en la página. Además, deberá preguntarse si está visible y es fácil de acceder a ese elemento, ya que, si es el elemento donde debe recaer la atención de los usuarios, debe ser el elemento más visible.

Debe pensar qué puede hacer para que el acceso a los elementos sea lo más sencillo posible.

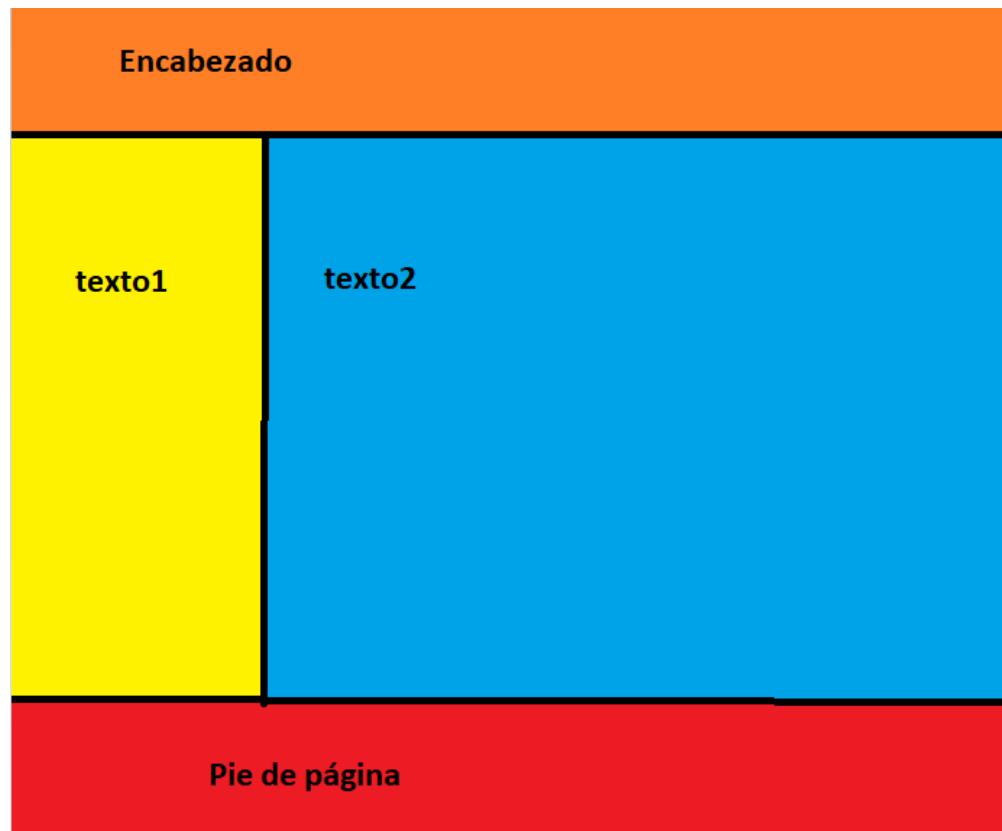
Otro punto a destacar es si la navegación es intuitiva para el usuario.

Si los nombres de los enlaces son fáciles de localizar y son explícitos.

Se recomienda utilizar un tamaño y color de texto fácil de visualizar, que tenga un buen contraste con el color de fondo para, de esa manera, asegurar una accesibilidad sencilla.

Por último, se recomienda utilizar imágenes con una versión adaptada a cada tipo de pantalla, incluyendo un vínculo hacia la versión del tamaño real de la imagen.

A continuación, dependiendo del posicionamiento de los elementos de la web, hará una estructura de la página de un tipo u otro. Para ello, puede dividir la página de múltiples maneras, cada una de las páginas tendrá su propia estructura y eso dependerá de a qué fin esté destinada la página. A continuación, va a ver dos ejemplos de estructuras típicas en las que se dividen las páginas web.



(imagen 21)

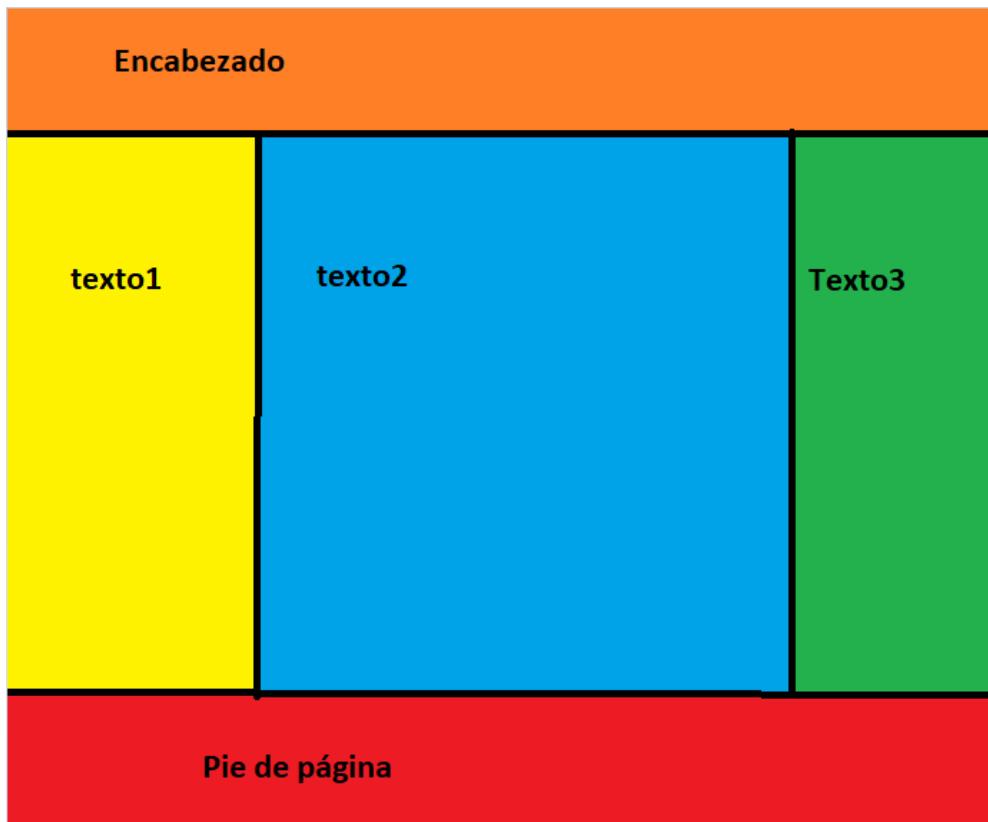
Como puede ver en esta imagen, la estructura de la web sería una estructura clásica donde se encontrarían los siguientes elementos:

Encabezado de color naranja.

Menú en color amarillo.

Contenido en color azul.

Pie en color rojo.



En esta otra división, puede observar que tiene una estructura muy similar a la anterior. Sin embargo, tiene un espacio más añadido. Los elementos de esta página serían los siguientes:

- Encabezado color naranja.
- Menú color amarillo.
- Contenido color azul.
- Submenú o banner vertical color verde.
- Pie en color rojo.

En este ejemplo, se ha añadido una nueva sección a la derecha para poder añadir un submenú o un banner informativo.

A continuación, verá otros dos ejemplos de divisiones clásicas que puede encontrar. Esta vez tendrá un estilo horizontal. Vea los ejemplos:



En este ejemplo puede ver cómo la estructura de página y la posición de todos sus elementos está en horizontal. Los elementos que puede encontrar en este ejemplo son los siguientes:

- Encabezado color naranja.
- Menú color azul.
- Contenido color morado.
- Pie en color rojo.

Si se fija, esta es la estructura que ha utilizado para el documento web que ha estado creando en epígrafes anteriores.

En el segundo ejemplo, tiene otro tipo de estructura muy parecida a esta que acaba de ver, pero tiene un elemento extra:



Los elementos que encuentra en este segundo ejemplo son los siguientes:

- Encabezado color naranja.
- Menú color azul.
- Contenido color morado.
- Submenú color verde.
- Pie de página color rojo.

En este ejemplo se añade un submenú junto al contenido. También puede añadir un banner o cualquier otro elemento que desee.

## 2.7. Definición de los tipos de página en base a los contenidos y funcionalidades

Al empezar a crear la estructura para una página web, se topa con varios estilos de página basados en la posición de los contenidos, los menús de navegación y los enlaces.

Como ya ha visto, existen diversos tipos de estructuras básicas para una página web, donde vienen especificadas las posiciones de los menús y del contenido, en la que la estructura y el diseño de la misma se va a mantener en todas las páginas que formen el sitio web.

Lo más común a la hora de diseñar una página es ubicar los menús en los márgenes de la página y el contenido en el centro de la misma. Las estructuras más básicas y estandarizadas de las webs es

posicionar los menús arriba o en los lados de la página, aunque puede encontrar otras muchas opciones de hacerlo:

#### **Bloque de navegación arriba.**

Utilizará esta estructura cuando la página web que está creando no tiene muchas secciones de contenido. Se situará el menú en línea y ocupará mucho espacio. La navegación será sencilla si los menús se despliegan en submenús o si son menús muy cortos.

#### **Bloque de navegación izquierda.**

Esta estructura permite presentar muchos menús y enlaces, puesto que, cada línea puede ser un enlace. En caso de que los menús sean muy extensos no va a haber ningún problema, ya que, el usuario podrá moverse mediante la barra de desplazamiento vertical.

Este tipo de estructuras se suelen utilizar cuando los menús son muy extensos, pues el diseño no se va a ver muy afectado por ello si el menú aumentara o disminuyera.

#### **Bloque de navegación derecha.**

Esta estructura se comportaría igual que la anterior. Aunque, realmente, estructura se utiliza menos que la anterior.

#### **Combinación de navegación arriba e izquierda/derecha**

Se utiliza esta estructura cuando la página tenga muchos enlaces que mostrar. Por ello, se pueden combinar cualquiera de las estructuras que ha visto anteriormente para que los menús se puedan agrupar en grandes grupos de enlaces.

Otra de las maneras de establecer un tipo de página basándose en sus funcionalidades es por su estructura de navegación, que vendrá definida dependiendo de la experiencia de usuario que se basará en cómo puede el usuario cambiar de una página a otra mediante sus enlaces.

La estructura de navegación se suele escoger dependiendo del tipo de contenido de la web. Vea a continuación los diferentes tipos que hay:

#### **Estructura jerárquica**

Este tipo de estructura tiene forma de árbol donde puede encontrar una página principal desde la que puede acceder al resto de páginas. Desde las otras páginas, se puede acceder a las diferentes subpáginas, de este modo se van creando niveles en jerarquía.

Este tipo de estructura es aconsejable para webs de centros o proyectos. Sin embargo, no se aconseja utilizar más de cuatro niveles, de esta manera evita que el usuario se desoriente durante la navegación.

Lo que sí se recomienda encarecidamente es que en todas las páginas haya un menú desde donde el usuario pueda pasar de manera fácil a otros niveles y páginas de cada uno de los niveles.

#### Estructura lineal

Esta es una de las estructuras más simples, puede compararla con las páginas de un libro. Desde una página, se puede navegar hasta la siguiente página o hacia la página anterior a ésta. Este tipo de estructuras se utilizan cuando quiera que el usuario siga un camino fijo y guiado sin que tenga el poder de acceder a otra página que pueda desorientarlo.

Esta estructura se suele utilizar para los sitios web que funcionan como tutoriales de aprendizaje, cursos o guías. No se recomienda utilizar esta estructura para un número de páginas muy grande, puesto que, puede hacer que el usuario se canse al no poder retomar la página que dejó vista, es decir, cada vez que el usuario entre, deberá hacer el mismo recorrido, ya que, no puede saltar a la página que deseé.

#### Estructura lineal con jerarquía

Este tipo de estructura combina la jerárquica, que ya ha visto, con la lineal. En esta estructura las páginas y las subpáginas se organizan de manera jerárquica, sin embargo, también tiene la opción de navegar de manera lineal por las páginas que están al mismo nivel.

#### Estructura en red

Esta estructura permite que, desde la página principal, se pueda navegar por cualquier página si ningún orden en concreto. Esta estructura es la más libre de todas, sin embargo, no es aconsejable utilizarla cuando tiene un gran número de páginas, puesto que, el usuario podría desorientarse y no saber cómo utilizar los recursos para navegar hacia una página específica.

### 2.8. Selección de los tipos de páginas para página web

Tomando como referencia la web funcional que está desarrollando, debe elegir un tipo de estructura específico. Como la web que está realizando es muy básica, la estructura más recomendable para la navegación es una estructura que tenga el bloque de navegación arriba de todo.



TOME NOTA

*La web que está desarrollando tiene pocas secciones en el menú, en cambio, si en un futuro quisiera aumentar el menú no tendría problema alguno, puesto que, al estar arriba, el usuario podría ampliar el menú rápida y fácilmente.*

Visto todo esto, la estructura que se escoge para la web será la estructura lineal con jerarquía. Como sus páginas ya están puestas de manera jerárquica, esta estructura será la más adecuada para su proyecto, de esta manera el usuario podrá navegar por las páginas del mismo nivel.

Basándose en las características de su web, ha elegido la estructura con bloque de navegación superior, en el que se hallan los cuatro botones que permitirán navegar por las cuatro páginas webs que componen el sitio, Inicio, Fotografías, CV y Contacto.

Debido a esta manera de navegación, sabe que la estructura de su sitio web es de manera lineal con jerarquía, puesto que es la opción más acertada basándose en el tipo de página que está desarrollando. Puesto que, lo que quiere es que el usuario pueda consumir la información de manera directa y sencilla. Utilizando este tipo de estructura, podrá hacerlo sin dar muchos rodeos.

## 2.9. Utilización del documento funcional para las especificaciones del tipo de página

En este epígrafe va a analizar el código de su página web y va a ver las diferentes capas que ha creado para su estructura. De esta manera, va a ver cómo ha dividido los diversos elementos que la componen:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta charset="utf-8" />
5  <title>Página fotógrafo</title>
6  <link rel="stylesheet" type="text/css" href="css/estilo.css">
7  </head>
8  <body>
9  <div class="general">
10 <div class="titulo">Página Fotografía</div>
11
12 <div class="menu">
13 <a href="main.html">Inicio</a>
14 <a href="fotos.html">Fotografías</a>
15 <a href="cv.html">CV</a>
16 <a href="contacto.html">Contacto</a>
17 </div>
18
19 <div class="contenido">
20 <span id="capitular">L</span>es invito a visitar mi página web.
21 Soy una fotógrafa profesional que me dedico a retratar paisajes
22 naturales que encuentro en los viajes que realizo.
23 <p>En esta página podrá ver todos mis trabajos. Para ello,
24 solo tiene que pulsar el enlace Fotografías que se ubica en
25 el menú de la página.
26 </p>
27 <p>Espero que lo disfruten.</p>
28
29 </div>
30 </div>
31 |
32 </body>
33 </html>
```

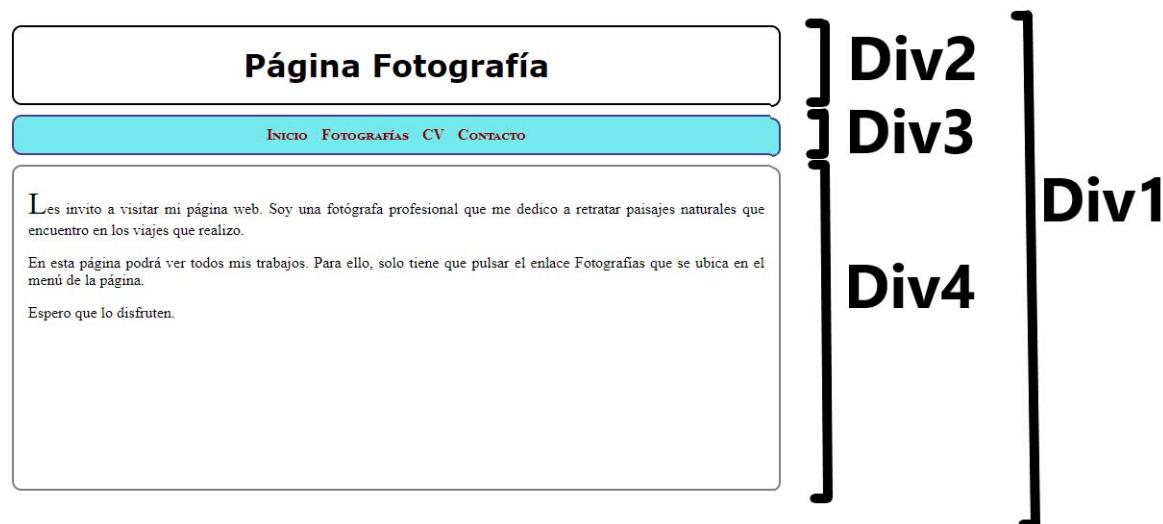


- Div1: Esta capa es la que contiene todos los elementos visibles de la página web, es decir, todos los elementos que podrá visualizar el usuario en el navegador.
- Div2: Esta capa contiene el título, logo o banner de la página. Esta es la primera capa que tiene en la página web.

- Div3: Esta capa contiene el menú de navegación.
- Div4: Esta es la última capa de la página y contendrá el contenido específico de cada una de las páginas, el cual irá cambiando conforme lo haga la página, es decir, el contenido que hay en main.html, no será el mismo de contacto.html.

Como puede observar en esta imagen, las capas están posicionadas de manera lineal, es por ello que cuando ve esta página en el navegador, puede ver las divisiones de cada una de las partes que van desde arriba hacia abajo.

Lo siguiente que va a ver es una imagen de la página web mostrada por el navegador, donde irá interpretando las mismas capas que se han codificado en HTML:



- Div1: Como puede observar en la imagen, la capa 1 será el fondo blanco de la página. Sin embargo, al no tener bordes parece que no hay ninguna división, puesto que, la función que tiene esta capa es la de contener todos los demás elementos.
- Div2: Esta capa, como puede ver, contiene el título de la página.
- Div3: Esta capa es la que contiene el menú de navegación de la página mediante el cual el usuario podrá navegar por la web.
- Div4: Por último, encuentra la cuarta capa donde puede observar que se encuentra el contenido específico de cada página. En este caso es un texto de bienvenida.

## 2.10. Especificaciones de navegación

Hay que tener en cuenta que la navegación de un sitio web es una de las cosas más importantes. No existe una página web que no tenga enlaces mediante los cuales el usuario pueda navegar por ella.

Si existiera una página sin enlaces, sería una página muy complicada para navegar y localizar el contenido que quiera encontrar.

Otra de las cosas muy importantes sobre la navegación es que debe ser intuitiva para el usuario. De ese modo, los usuarios tendrán una experiencia positiva al visitar ese sitio web y podrán encontrar el contenido que buscan de manera instantánea.

Crear un buen diseño para el menú de navegación no es algo sencillo de hacer, puesto que, tiene que tener en cuenta muchos puntos. Por ejemplo, la densidad del menú que necesita y el diseño que quiere que éste tenga. De este modo, podría diseñar un menú sencillo o complejo que pueda facilitar la búsqueda del usuario o, en cambio, complicarla más.

El diseño del menú de navegación va a depender del sitio web. No es lo mismo crear un menú para una tienda online que para un periódico online. Por ello, en la mayoría de las ocasiones, crear un menú sencillo va a servir para conseguir un sitio web correctamente estructurado. Sin embargo, otras veces, debe crear menús más complejos añadiendo submenús.

A continuación, va a ver algunos tips para mejorar la navegación de un sitio web:

### **Organización de la estructura de navegación web**

La organización de los contenidos del menú es la parte más importante a la hora de crear un buen diseño, de esta manera el menú será más sencillo e intuitivo para los usuarios y para encontrar la información que desean.

Para esta tarea, los desarrolladores web utilizan una conocida práctica denominada arquitectura de la información, que es la que se recomienda para mejorar la experiencia de usuario.

### **Esquema de menú de navegación**

Cuando realiza un esquema para el menú de navegación, debe siempre pensar en un futuro, ya que, el menú podría sufrir cambios. Otra de las cosas que debe hacer es hacer el menú de manera que sea sencillo para la navegación de los usuarios.

Para ello, lo mejor que puede hacer es visitar las webs de otros desarrolladores que sean de la misma temática que la web que va a crear. De esta manera, puede tomar ideas para, posteriormente, añadirlas a su web. Otra de las cosas que debe hacer es un test de usabilidad, si el sitio web va a tener mucho tráfico.

Cuando ya tenga claro cómo va a estructurar los contenidos del menú y de la web, pasará a crear el esquema de navegación web, el cual servirá para dar una visión clara de cómo se van a relacionar los contenidos entre sí. Se recomienda que los contenidos no estén a más de dos clics.

### **Menú principal y secundario**

Si tiene una página web con tal cantidad de contenidos que no tienen relación entre sí, debería crear un menú principal y un menú secundario.

Menú principal:

Este menú deberá contener las opciones principales de cara al usuario del sitio. Debe saber que dependiendo lo que la web quiera destacar, tendrá los enlaces en el menú principal o en el secundario.

Menú secundario:

En este menú, encuentra enlaces de valor que no están relacionados con los demás o no son lo demasiado importantes para estar en el menú principal.

### Agrupación de contenidos

En caso de que tenga mucho contenido en su web, debe preguntarse si debe organizarlo por orden alfabético o por prioridad. Vea a continuación una lista de los métodos de clasificación más utilizados:

Orden cronológico: Este tipo de orden es una buena elección en caso de que la fecha de publicación de los contenidos sea importante como, por ejemplo, en un blog.

Orden alfabético: Este tipo de orden es muy útil utilizarlo cuando el usuario entra en la web para encontrar un contenido de manera rápida.

Popularidad: Este tipo de estructura es útil cuando el usuario busca contenidos dependiendo de los intereses del propio usuario. Aunque, esta manera de estructurar los contenidos deja de lado otros contenidos que pueden ser interesantes destacar.

Pasos a seguir: Este tipo de estructura es útil cuando el contenido responda a una pregunta. Como, por ejemplo, un bot automático de ayuda.

### Horizontal VS Vertical

Los desarrolladores web tendrán múltiples factores por los cuales elegir un diseño del menú horizontal o vertical. Por ejemplo: la usabilidad o cantidad de contenidos, el diseño o añadir elementos visuales que hagan la navegación más sencilla para el usuario. Además, puede que el desarrollador escoja una tipografía específica para dar un toque más personal al diseño.

Otra de las cosas que debe tener en cuenta es si el sitio web va a contar con más de un idioma. Esto se debe a que no será igual de extenso un menú en español, en árabe o en francés. En un idioma el menú quedará bien y en otro habrá que ampliarlo.

### Submenús

---

Si la página web tiene muchos contenidos, los submenús serán una pieza clave. En caso de que quiera hacer la navegación más sencilla para el usuario, debe estructurar bien los contenidos y ser creativos para que el usuario no se pierda entre tanta información.

Se recomienda no añadir un pequeño margen entre el elemento siguiente cuando se trata de un menú horizontal, puesto que, dificultaría la navegación.

Vea a continuación unas ideas que puede tener en cuenta cuando cree submenús para mejorar la experiencia de usuario en su sitio web:

Utilización de categorías y subcategorías. Debe saber que la clasificación en un submenú es algo muy importante. Por ello, se recomienda realizar un esquema de navegación tipo lógico antes de seleccionar el tipo de diseño que va a añadir. Esto ayudará a la navegación.

Utilización de imágenes, tipografía e iconos. Si quiere mejorar la calidad del submenú, se recomienda utilizar espacios, imágenes, iconos o tipografías, en vez de la letra pequeña o poner mucha información en un espacio reducido.

Diferenciar cada submenú. Cada categoría debe tener un submenú diferente a los demás, pero sin perder la coherencia del diseño. Para ello, puede utilizar distintos colores.

## 2.11. Creación de un mapa de navegación de páginas

Mediante un mapa de navegación, puede presentar de manera gráfica y plana la estructura de los enlaces que existe en un sitio web. De esta manera, puede indicar los conceptos primordiales que hay en un sitio web y las relaciones que tienen los contenidos entre sí.

Por lo tanto, un mapa del sitio sirve para representar el sitio web de manera plana y, de ese modo, orientar al usuario mientras navegue por la web o para facilitarle un acceso directo hacia la página que le interese.

En definitiva, el mapa del sitio debe mostrar la estructura de la web median los enlaces de la misma que van redirigidos a los nodos principales, los cuales puede también desarrollar para mostrar los subnodos. El mapa podrá representarse de manera textual, o de manera gráfica, incluso combinando las dos maneras.

Un mapa de navegación funciona como un índice o una tabla de contenido que se utiliza para moverse a través de los documentos. Este mapa puede encontrarlo en las webs antiguas con forma de menú.

En caso de que quiera que los usuarios tengan una visión general del contenido, será muy útil generar un mapa con todos los contenidos de la web y que los usuarios puedan visualizarlo en una pantalla donde puedan acceder al contenido mediante el nodo del documento.

En cambio, si el sitio web es demasiado extenso, mostrar todo el contenido en un mapa y en una sola pantalla es prácticamente imposible y, tampoco es recomendable crear un gran entresijo de letras o imágenes.

Por lo tanto, si tiene una web muy extensa, crear un mapa del sitio que se pueda visualizar en una única pantalla, va a suponer una tarea muy complicada y, en caso que logre realizarla, va a ser algo muy confuso para el usuario.

Como solución a esto, puede crear diagramas que muestren de manera local los nodos más importantes para, posteriormente, mostrar cada subnodo en diagramas locales más detallados. Además, puede mostrar estos diagramas tan grandes mediante mapas de ojo de pez. Con este tipo de mapas ser capaz de mostrar todo el espacio web en un solo gráfico que se divide en niveles diferentes. De este modo, puede mostrar más información sobre dónde se encuentra el usuario y ocultar detalles de los sitios donde se encuentre más alejado.

Crear este tipo de nodos suele ser bastante trabajoso, puesto que, debería tratar cada nodo de manera específica.

Puede encontrar en Internet muchísimos programas que ayudan a realizar mapas de sitio que muestran el nodo activo y su contexto. Además, también muestran de manera gráfica la estructura total del documento de hipertexto. En cambio, estas herramientas no son del todo de utilidad, puesto que, solo funcionarán dentro del programa constructor de páginas y no cuando esté publicada la web.

En definitiva, un mapa de navegación del sitio web es una representación textual o gráfica de un sitio web que es comparable con el índice de un libro. Este tipo de mapas están compuesto por una página HTML que muestra la estructura del sitio:

- Cómo están enlazados los contenidos.
- Archivos.
- Directorios.

Este tipo de mapas de navegación hace que el usuario pueda navegar de manera más sencilla, también ayuda a los buscadores, puesto que, son capaces de diferenciar entre dos tipos de mapas, los mapas HTML y los mapas XML. Los mapas XML se utilizan para posicionar el sitio web.

En consecuencia, con que estos mapas se utilizan para el posicionamiento web, se hicieron populares y suelen utilizarse para hacer un acceso a todas las secciones de la página web más sencillo y que estos documentos se incluyan en los resultados de búsqueda de los buscadores.

A continuación, va a volver a los tipos de mapas HTML y va a ver sus características:

- Deben estar resumidos
- Deben estar jerarquizados
- Deben ser claros

- Deben contener títulos descriptivos
- Deben ser visualmente estéticos

## 2.12. Utilización del documento funcional para integrar el mapa de navegación

A continuación, va a ver las modificaciones que ha realizado en su web para añadir un mapa del sitio:

Primeramente, lo que ha hecho es crear un nuevo documento con los enlaces a cada una de las páginas. Por lo que el documento HTML ha quedado de la siguiente manera:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Mapa del sitio</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>

<p class="posicionmapasitio">

<ul class="mapasitio">

<li><p><a href="main.html" target="_blank">Inicio</a></p></li>

</ul>

<ul class="mapasitio">

<li><p><a href="fotos.html" target="_blank">Fotografías</a></p></li>

<ul class="mapasitio">

<li><p><a href="imagenes/foto1.jpg" target="_blank">Foto 1</a></p></li>

<li><p><a href="imagenes/foto2.jpg" target="_blank">Foto 2</a></p></li>

<li><p><a href="imagenes/foto3.jpg" target="_blank">Foto 3</a></p></li>
```

```
<li><p><a href="imagenes/foto4.jpg" target="_blank">Foto 4</a></p></li>
<li><p><a href="imagenes/foto5.jpg" target="_blank">Foto 5</a></p></li>
<li><p><a href="imagenes/foto6.jpg" target="_blank">Foto 6</a></p></li>
<li><p><a href="imagenes/foto7.jpg" target="_blank">Foto 7</a></p></li>
<li><p><a href="imagenes/foto8.jpg" target="_blank">Foto 8</a></p></li>
<li><p><a href="imagenes/foto9.jpg" target="_blank">Foto 9</a></p></li>
<li><p><a href="imagenes/foto10.jpg" target="_blank">Foto 10</a></p></li>
</ul>
</ul>

<ul class="mapasitio">
    <li><p><a href="cv.html" target="_blank">CV</a></p></li>
    <ul class="mapasitio">
        <li><p><a href="documentos/cv.pdf" target="_blank">CV PDF</a></p></li>
    </ul>
</ul>
<ul class="mapasitio">
    <li><p><a href="contacto.html" target="_blank">Contacto</a></p></li>
</ul>
</p>
</body>
</html>
```

Como puede observar, este documento contiene una lista no ordenada donde puede encontrar que los elementos principales están compuestos por los enlaces del menú y los secundarios por los enlaces que se ubican dentro del contenido de cada una de las páginas.

---

Lo segundo que ha hecho es aplicar dos estilos de clase para darle un formato al documento. Recuerde que, al añadir el mapa de sitio dentro del cuerpo del documento, heredará el color de fondo.

Se ha añadido algunos estilos al mapa del sitio, por ejemplo, un tipo de fuente específico, una anchura y una posición de relleno de 15px. Vea el siguiente código CSS:

```
.mapasitio {  
    font-family: Verdana, Geneva, Tahoma, sans-serif;  
    border-style: none;  
    width: 800px;  
    margin: 0 auto;  
    position: relative;  
    padding-left: 40px;  
}  
  
 .
```

```
.posicionmapasitio {  
    padding-top: 40px;  
}
```

Vea los valores que ha modificado para añadir el pie de página al documento HTML. Para ello, se han tenido que reducir los píxeles de la altura de la capa a la que pertenece el bloque del contenido de la página a 200px. Esta modificación no ha tenido ningún tipo de efecto en la página, puesto que, son páginas con muy poco contenido. En caso de que fueran páginas con mucho contenido, se deberían modificar las proporciones de las imágenes o las proporciones relativas de la web.

Vea en el código HTML que verá a continuación cómo se ha integrado el pie de página en los diferentes documentos HTML:

```
<!DOCTYPE html>  
  
<html>  
  
<head>
```

```
<meta charset="utf-8" />

<title>Página fotógrafo</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>

<div class="general">

<div class="titulo">Página Fotografía</div>

<br><br><br><br><br>

<div class="menu">

<a href="main.html">Inicio</a>

<a href="fotos.html">Fotografías</a>

<a href="cv.html">CV</a>

<a href="contacto.html">Contacto</a>

</div>

<br><br><br><br><br>

<div class="contenido">

<span id="capitular">L</span>es invito a visitar mi página web.

Soy una fotógrafa profesional que me dedico a retratar paisajes

naturales que encuentro en los viajes que realizo.

<p>En esta página podrá ver todos mis trabajos. Para ello,

solo tiene que pulsar el enlace Fotografías que se ubica en

el menú de la página.

</p>
```

```
<p>Espero que lo disfruten.</p>

</div>

<div class="footer"><p><a href="mapaweb.html" target="_blank">Mapa del sitio</a></p></div>

</div>

</body>

</html>
```

Como puede ver, tan solo se ha añadido una nueva capa al final del documento, estando dentro de la capa general.

En el documento contacto.html, se ha tenido que realizar cambios en el formulario para que haya podido caber el pie de página y le dé armonía al documento. Va a ver a continuación los cambios que se han tenido que realizar:

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title>Página Fotógrafo</title>

<link rel="stylesheet" type="text/css" href="css/estilo.css">

</head>

<body>

<div class="general">

<div class="titulo">Página Fotografía</div>

<div class="menu">
```

```
<a href="main.html">Inicio</a>  
<a href="fotos.html">Fotografías</a>  
<a href="cv.html">CV</a>  
<a href="contacto.html">Contacto</a>  
</div>  
  
<div class="contenido">  
    <div>  
          
    </div>  
  
    <span id="capitular">Muchas gracias por su visita. Puede ponerse  
    en contacto conmigo a través de este formulario para decirme qué le ha parecido mi trabajo  
    o si tiene alguna duda que yo le pueda solucionar.  
  
<form method="post" action="mailto:correo@dominio.com" enctype="text/plain">  
    <textarea rows="6" cols="30" name="comentarios"></textarea>  
    <p><button type="submit">Enviar</button>&nbsp; &nbsp;  
    <button type="reset">Cancelar</button></p>  
</form>  
</div>  
</div>  
  
<div class="footer"><p><a href="mapaweb.html" target="_blank">Mapa del sitio</a></p></div>  
</body>
```

</html>

Como puede observar en el documento HTML, ha tenido que eliminar el texto que ponía comentarios, ya que, no va a dejar subir la capa hasta 200px. Como es un texto que no es muy relevante, no ha sido una gran pérdida para la funcionalidad de la web. Vea a continuación el siguiente código CSS con los estilos de contacto:

```
.contacto{  
min-height: 200px;  
}  
Creado por
```

Como puede ver, se ha añadido un estilo para reducir el bloque de contacto a 200px. Vea en la siguiente imagen el documento mapasitio.html mostrado por el navegador:

- [Inicio](#)
- [Fotografías](#)
  - [Foto 1](#)
  - [Foto 2](#)
  - [Foto 3](#)
  - [Foto 4](#)
  - [Foto 5](#)
  - [Foto 6](#)
  - [Foto 7](#)
  - [Foto 8](#)
  - [Foto 9](#)
  - [Foto 10](#)
- [CV](#)
  - [CV PDF](#)
- [Contacto](#)

Como puede observar, es una lista no ordenada que porta los elementos que componen su página. Cada uno de los enlaces que se muestra conducirá a ese recurso en concreto. Los enlaces de las fotos van directamente a las fotografías correspondientes. Todos los recursos se abrirán en una nueva ventana.

A continuación, va a ver cómo queda el pie de la página de inicio:

# Página Fotografía

[INICIO](#) [FOTOGRAFIAS](#) [CV](#) [CONTACTO](#)

Les invito a visitar mi página web. Soy una fotógrafa profesional que me dedico a retratar paisajes naturales que encuentro en los viajes que realizo.

En esta página podrá ver todos mis trabajos. Para ello, solo tiene que pulsar el enlace Fotografías que se ubica en el menú de la página.

Espero que lo disfruten.

Mapa del sitio

Como puede ver, el pie de página tiene un pequeño número de píxeles en donde puede integrar el texto de enlace, para que ese enlace lleve al documento que muestra el mapa del sitio. Vea los estilos específicos que se le han aplicado al pie de página:

```
.footer{  
margin-top: 10px;  
position: relative;  
padding-top: 10px;  
padding-right: 15px;  
padding-bottom: 15px;  
padding-left: 15px;  
border-style: solid;  
border-color: gray;  
border-radius: inherit;  
border-width: 2px;
```

```
text-align: left;  
  
min-height: 30px;  
  
}
```

```
.footer a{  
  
text-decoration: none;  
  
color: darkred;  
  
}  
  
.footer a:hover{  
  
color: fuchsia;  
  
}  
  
.footer a:visited{  
  
color: blueviolet;  
  
}
```

## 2.13. Elementos utilizados para la navegación

Los elementos que utiliza para navegar por la web son un grupo de enlaces que van enlazados a cada uno de los elementos que componen el menú del sitio web.

El menú de navegación, podrá verse durante toda la navegación web, por lo tanto, el usuario puede en cualquier momento dirigirse al menú para desplazarse por la página sin perderse por el camino. Vea los elementos con los que se compone el menú de navegación a continuación:

- Inicio
- Fotografías
- CV
- Contacto

Cada uno de los enlaces redirigirá al documento concreto. Además, en cada uno de los documentos a los que enlaza, puede seguir viendo el mismo menú para poder seguir navegando por la web. Lo único que cambiará en cada uno de los documentos, será el contenido específico del mismo. En cambio, el título y el menú permanecerán estáticos.

Vea a continuación cómo se muestra el menú de la web:

**INICIO FOTOGRAFÍAS CV CONTACTO**

### **2.13.1. Definición de los elementos utilizados para navegar**

Como ya se vio anteriormente, los elementos que ha seleccionado para navegar por la web, se encuentran en el menú. Vea a continuación los estilos con los que se define este menú:

```
.menu{  
background-color: rgb(116, 234, 238);  
margin-top: 10px;  
margin-bottom: 10px;  
border-style: solid;  
border-color: darkslateblue;  
border-width: 2px;  
padding-top: 10px;  
padding-bottom: 10px;  
font-weight: bold;  
font-variant: small-caps;  
word-spacing: 10px;  
border-radius: 10px;  
}  
  
 .menu a{
```

```
color: darkred;  
  
text-decoration: none;  
  
}
```

```
.menu a:hover{
```

```
color: fuchsia;  
  
}
```

```
.menu a:visited{
```

```
color: blueviolet;  
  
}
```

Va a ver los estilos que ha seleccionado para su menú:

- Backgorund-color: el color que ha seleccionado de fondo del menú es turquesa.
- Margin: Se ha aplicado un margen de 10px.
- Border-style: Se le ha aplicado un borde sólido para que se muestre en el navegador.
- Border-color: El color del borde del menú es darkslateblue.
- Border-width: Se le ha aplicado un ancho en el borde de 2px.
- Padding-top: Se le ha aplicado un espacio superior de 10px.
- Padding:bottom: Se le ha aplicado un espacio inferior de 10px.
- Font-weight: El ancho de la fuente es bold.
- Font-variant: Se le ha aplicado a la fuente un estilo versalitas.
- Word-spacing: El espacio entre letras es de 10px.

Además, se le ha aplicado también estilos al texto como enlace:

- Color: Se le ha aplicado el color darkred para que tenga un buen contraste con el fondo.
- Text-decoration: Se le ha aplicado el valor none para que no aparezca el texto subrayado.
- A:hover: Se le ha aplicado un color fucsia al texto para cuando el ratón pase por encima tenga un efecto.
- A:visited: Se le ha aplicado el color blueviolet al enlace visitado.

## 2.13.2. Utilización del documento funcional para especificar los elementos de navegación

Lo siguiente que verá son las especificaciones de navegación del documento que ha estado creando. Por lo tanto, va a ver el tipo de menú que ha creado en su web:

```
<div class="menu">  
  
    <a href="main.html">Inicio</a>  
  
    <a href="fotos.html">Fotografías</a>  
  
    <a href="cv.html">CV</a>  
  
    <a href="contacto.html">Contacto</a>  
  
</div>
```

Como puede ver, es un menú bastante simple, solo tiene cuatro enlaces que redirigen a los cuatro documentos de los cuales se compone el sitio web.

Se ha escogido este tipo de menú tan simple, debido al tipo de página que está creando, puesto que, como se trata de una página personal con muy poco contenido, no se va a necesitar utilizar un menú muy extenso con muchos submenús ni nada de eso.

Además, se ha añadido dentro de la página el menú horizontal y superior, de este modo. La navegación va a ser más sencilla e intuitiva para el usuario, porque podrá encontrar la información más sencilla y rápidamente.

Se ha utilizado un texto sencillo como enlace del menú. Además, se ha evitado introducir imágenes o iconos para que los usuarios puedan comprender fácilmente el menú.

Vea el orden que se ha utilizado para los elementos del menú:

- Inicio
- Fotografías
- CV
- Contacto

Con este orden pretende que el usuario, al leer de izquierda a derecha, empiece a navegar por el inicio y vaya poco a poco descubriendo el contenido de la página. Poniendo lo más interesante más a la izquierda.

Sin embargo, como es un menú muy básico, el usuario se podrá dirigir a cualquiera de los enlaces de manera aleatoria.

Por otro lado, en la hoja de estilo se ha especificado un color diferente a los enlaces que ya han sido visitados, de esta manera el usuario sabrá qué documentos ha visitado ya y cuáles no.

## 2.14. Elaboración de una guía de usuario

Los manuales de usuarios son guías que ofrecen instrucciones de cómo utilizar o hacer algo. Lo normal es que los programas tengan su propia guía de usuarios, aunque también hay guías para electrodomésticos, televisores, altavoces, móviles, y a todos los equipos electrónicos que puede encontrar. Un manual de usuario útil es aquel que educa al usuario sobre las características de un producto a la vez que se le enseña cómo utilizarlo de manera efectiva. Además, las guías están escritas para que puedan consultarse de manera rápida y fácil en cualquier momento. Vea algunas cosas que debe tener en cuenta para escribir un contenido útil y efectivo para un manual de usuarios:

### Primer método:

Lo primero que debe hacer es especificar quién es el usuario. Para escribir una guía eficaz, debe saber el perfil de usuario al que va dirigida. Para ello, puede crear un perfil escrito del usuario. Un perfil de este tipo es muy útil cuando es un equipo de trabajo el que está escribiendo la documentación y, además, también este perfil puede ayudar a desarrollar el producto desde su concepto hasta su desarrollo final. Vea qué elementos tiene que tener en cuenta al desarrollar un perfil de usuario:

El lugar donde los usuarios utilizarán la guía, por ejemplo, en casa, en la oficina o en un lugar de trabajo remoto. Esto podrá determinar tanto el estilo como el contenido de la guía.

Lo siguiente que debe tener en cuenta es la manera en la que los usuarios van a utilizar la guía. Es decir, si va a ser una guía que los usuarios van a consultar muchas veces o solo para buscar una información específica, ya que, en relación a esto, la guía debería tener un formato diferente. En caso de que en un principio el usuario fuera a consultar esta guía asiduamente, la sección consulta debería tener una sección llamada Cómo comenzar y en esta sección, se deberían añadir las instrucciones más sencilla y simples que utilizará el producto.

Otra de las preguntas que debe hacerse es cuánta experiencia tendrá el usuario con ese artículo o con artículos parecidos. Si el artículo es una novedad o es muy diferente a otros artículos que haya en el mercado, debe añadir una pequeña explicación de sus diferencias comparándolas con los demás artículos parecidos y, también, lo acompaña en cómo comenzar a utilizar el artículo. En caso de que sepa que este tipo de artículos son un poco problemáticos para algunos usuarios como, por ejemplo, algunos programas de ordenador, se deberá ofrecer una información detallada y muy fácilmente comprensible.

Debe escribir la guía conforme ayude al usuario con sus necesidades y de la manera más comprensible para el mismo. En caso de que el usuario al que va dirigida la guía no sea un usuario del campo técnico, debe evitar el lenguaje técnico para, de ese modo, conseguir que las explicaciones de la guía sean lo más claras posible. Por otra parte, debe organizar los textos de manera que estén acordes a cómo piensa el usuario, es más intuitivo para el usuario que liste las características del producto por grupos funcionales que listar solo aquellas que se utilizan de manera más cotidiana.

Es verdad que, en algunas ocasiones, debe utilizar lenguaje técnico para explicar las cosas, por ejemplo, en los programas de ordenador, muchas veces se utilizarán gráficas de Fibonacci, gráficos circulares o de barras. En estas ocasiones, debe definir algunos términos y ofrecer alguna información básica y detallada sobre el tema como, por ejemplo, explicar lo que es un gráfico de Fibonacci y para qué se utiliza.

### **Segundo método:**

Deberá añadir la portada y cubiertas correctas. Se deberá añadir una portada a cualquier guía de usuario mayor a una hoja, y en caso de que la guía sea más grande que la superficie de una hoja de papel, se deberá añadir una cubierta.

En caso de que el manual o el artículo en sí provea algunas condiciones de uso, debe ubicarlos en la portada interior del manual y en la cubierta.

Debe poner referencias sobre las documentaciones que van relacionadas en el prefacio. En caso de que la documentación sea más de una guía, se deberán colocar las referencias a los demás documentos con los números de versión correspondiente. En el prefacio también es donde pondrá una sección de cómo utilizar este manual, en caso de que la hubiera. Si la guía tiene una amplitud mayor a las 10 páginas, deberá incluir un índice de contenidos.

Debe añadir las instrucciones y las materias a consultar, dentro del cuerpo del manual. Normalmente, las instrucciones y las materias a consultar tienen sus propias secciones específicas, sin embargo, dentro de una sección podría aparecer una indicación al usuario para que consulte una parte específica de otra sección. De esta manera, el usuario podrá encontrar más rápidamente la información que busca.

Debe escribir los pasos a seguir en la guía con una estructura coherente a lo largo de la misma. Debe empezar por escribir un resumen del procedimiento, después, debe especificar qué es lo que debe hacer el usuario y cuáles son los resultados de sus acciones. Se deberán enumerar todos los pasos a seguir y empezar con acciones verbales al igual que se mostrarán los pasos en cada una de las secciones.

Además, todos los materiales a consultar deberán tener añadido una lista de opciones, consejos en caso de que haya que resolver cualquier incidencia y preguntas frecuentes. También, se podrán añadir índices y glosarios al final de la guía, aunque también puede posicionar una lista resumida al principio del mismo. Si el manual tiene menos de 20 páginas, podrá omitirse el índice del mismo.

Utilizará imágenes para aclarar el texto cuando así lo necesite. Las imágenes o capturas de pantalla pueden ser muy útiles para mostrar algunos procesos de mejor manera que solo con el texto, sobre todo si esos procesos son complejos y el usuario puede confundirse. Con las imágenes podrá comprobar de manera visual que está haciendo correctamente el procedimiento. Las capturas o imágenes se pueden crear utilizando programas de diseños asistidos, o programas de diseño gráfico y programas para edición de fotografía. Además, para crear capturas de pantalla, puede hacerlo

mediante la función que viene preinstalada en el ordenador o con cualquier programa de diseño gráfico que tenga la función de hacer capturas de pantalla.

Cuando haya terminado de crear la imagen, debe guardarla en un formato que sea lo más comprimido posible. Además, debe reducir el tamaño de la imagen para que pueda encajar correctamente en la página del manual, siempre permitiendo al usuario que pueda visualizarlo correctamente. En caso de que la imagen no pueda ser fácilmente visualizada, debe pasar a dividirla por partes, para que de ese modo el usuario pueda visualizar las partes más importantes del proceso.

En caso de que se utilicen una gran cantidad de imágenes en cada procedimiento, debe realizarlas todas con el mismo tamaño para que tengan una coherencia. De este modo, el manual será más atractivo a la vista. También, hay que tener en cuenta que, si toma una captura de pantalla y capture una serie de colores, debe capturar los mismos colores en las imágenes siguientes. Programas como Paint Shop o Photoshop ofrecen funciones de captura de pantalla muy buenas, sin embargo, el programa Snagit, que está especializado en esta tarea, permitirá modificar, comentar y catalogar de una manera mucho más sencilla las capturas de pantalla.

### **Tercer método:**

Debe optar por una fuente fácilmente legible. Es cierto que los ordenadores tienen multitud de tipos de fuente, pero entre ellas debe escoger la más legible. Para ello, debe escoger un pequeño grupo de fuentes en el cual las letras se vean bien al estar juntas. Los tipos de fuentes se pueden dividir en dos grupos sans serif y serif.

Las fuentes que pertenecen al grupo serif se diferencian en que tienen pequeños adornos al final de los trazos de las letras. Las fuentes serif más conocidas son Book Antiqua, Times New Roman y Baskerville. Este tipo de fuentes se utilizan de mejor manera en los fragmentos de textos que son muy amplios y que están dentro del cuerpo principal de la guía con un tamaño de letra de 10 a 12 puntos.

En cambio, las fuentes sans serif no tienen ningún tipo de decoración en los trazos de las letras. Las fuentes más conocidas de este grupo son Helvetica, Arial, Calibri y Century Gothic. Este tipo de fuentes se pueden utilizar en fragmentos grandes de texto con un tamaño entre 8 y 10 puntos. Sin embargo, al ser sans serif, este tipo de letras se hace difícil de leer cuando tienen un tamaño de 12 puntos o superior. Por otro lado, este tipo de letras puede ser utilizado para escribir grandes títulos y encabezados, y son muy útiles para utilizar en los pies de página y números en tablas y columnas.

En definitiva, para escribir manuales de usuarios, debería escoger tipos de fuentes simples como Times New Roman o Arial y, también puede utilizar una fuente tipo serif para escribir citas o títulos, por ejemplo, si escribe manuales de videojuegos de ciencia ficción o fantasía.

Cuando ya tenga la fuente elegida, debe escribir una página del manual y comprobar que las fuentes elegidas tienen una coherencia estética y son lo suficientemente legibles. Una recomendación es enseñarle a otra persona o a la persona encargada de aprobar la estética del manual.

Cuando haya terminado de establecer las fuentes de la página, tendrá que establecer la estructura que va a tener el manual y debe decidir dónde colocar cada uno de los elementos.

## RESUMEN

En esta Unidad Formativa ha aprendido, primeramente, la sintaxis de CSS y sus partes. También, ha aprendido a utilizar sus seudoclases y seudoelementos para crear, de este modo, páginas más dinámicas.

Además, ha visto las diferentes maneras que tiene para utilizar CSS junto con HTML, cómo relacionar HTML con CSS mediante selectores id y de clase.

Ha profundizado en los elementos que se utilizan para la navegación de una web cuáles son imprescindibles y cuáles no.

Otra de las cosas muy importantes que ha aprendido es a crear documentos accesibles para todo tipo de usuarios. A crear documentos específicos para cada tipo de medio.

También, ha aprendido a modificar los recursos para poderlos insertar en una página web de manera homogénea y con coherencia para que no altere el estilo de la misma.

Ha visto cómo crear bloques para delimitar las capas de un documento y así puede posicionar el contenido donde desee. También, cómo aplicar estilos heredados a diferentes elementos.

En esta Unidad Formativa, ha visto qué tipo de herramientas de edición de código puede utilizar para ayudar con el desarrollo web y qué herramientas ayudan con la optimización y con la corrección de errores de la misma.

Ha visto los atributos específicos que puede utilizar para los párrafos, los bloques, las tablas y las fuentes. Un atributo muy importante para el posicionamiento de los elementos que ha visto es z-index.

Las seudoclases más importantes y que más se utilizan que ha aprendido son :hover, y :visited. Y, una propiedad muy útil para los estilos de los bordes es border-radius, que facilitará mucho la tarea de redondear las esquinas de los bordes.

En definitiva, en este tema ha aprendido todo lo necesario para crear la parte de HTML y la CSS de un sitio web.

## GLOSARIO

**Bloque:** Un bloque es una especie de espacio que puede crear en HTML donde puede añadir imágenes, textos, y otros recursos.

**Bookmarklet:** Es un marcador que, en vez de apuntar hacia una url, apunta a una pequeña parte de código JavaScript que se encarga de ejecutar tareas específicas de manera automática.

**Cabecera:** Esta es la primera parte de un documento HTML y es donde reside toda la información acerca del documento.

**Cascada:** Esto significa que en CSS el orden de las reglas importa. Cuando dos reglas tengan la misma especificidad, se aplicará la que aparezca en último lugar. De ahí a que se le llame cascada.

**Clase:** Las clases permiten a CSS y JavaScript seleccionar y acceder a elementos específicos mediante este tipo de selectores.

**Declaración:** Es la unidad básica de CSS. Esto quiere decir que no se puede utilizar ninguna estructura más pequeña que ésta en CSS.

**Extensión:** Es la parte final que se encuentra en un nombre de archivo. Por ejemplo: .css, .html, .js y sirve para que el ordenador reconozca el tipo de archivo y, de ese modo, pueda tratarlo de una manera u otra.

**FTP:** Sus siglas significan Protocolo de transferencia de archivos. Por lo tanto, un FTP es un protocolo que se utiliza para transferir archivos por toda la red y permitirá gestionar archivos que se hallen en un servidor.

**Herencia:** La herencia se entiende como que los valores que se establecen en los elementos padre, pasan a los elementos hijos, es decir, a los elementos contenidos en el elemento padre.

**Hojas de estilo:** CSS o Cascading Style Sheets es el lenguaje que utiliza para crear hojas de estilo. Estas hojas se utilizan para desarrollar el aspecto y el formato de un documento HTML.

**Id:** Los selectores id son iguales que los de clase, la diferencia es que solo puede seleccionar un solo elemento en todo el documento, ya que, los id deben ser únicos.

**Mapa del sitio:** Es una lista de las páginas y los recursos que contiene un sitio web.

**Navegador:** Es un programa que permite navegar a través de Internet y otras redes.

**Propiedad:** Esta es la primera parte de una declaración en CSS. Las propiedades indican al navegador qué característica de estilo de un elemento se debe modificar.

**Reglas CSS:** Las reglas son conjuntos de propiedades que se aplican a un elemento específico.

**Selectores:** Es un identificador en la hoja de estilo que identifica los elementos del DOM a los que se van a aplicar esos estilos.

**Sitio web:** Es un conjunto de documentos electrónicos y páginas web que tienen relación con un tema en concreto.

**Valor:** Un valor es una colección de subvalores en CSS. Por ejemplo, si utiliza una propiedad color, sus valores serán todos los códigos de color.

**Versalitas:** Es una variación de cualquier tipo de letra donde las letras mayúsculas adquieren un tamaño similar al de las minúsculas.