



Módulo 2: Fundamentos de Desarrollo Front-End

**Desarrollador de aplicaciones
Full Stack**

Python Trainee



MÓDULO 2 – FUNDAMENTOS DE DESARROLLO FRONT-END

2.3.- Aplicación de estilos y responsividad

2.3.1.- Manejando hojas de estilo

2.3.1.1.- Qué es CSS, fundamentos y utilidad

En tópicos anteriores se ha visto cómo crear un documento HTML, organizando su estructura, determinando qué elementos son más apropiados para representar su contenido; esto, por cierto, permite definir el documento, pero no determina de qué forma se mostrará en pantalla. Desde la introducción de HTML5 esa tarea es responsabilidad de CSS, que es un lenguaje que facilita instrucciones que se usan para asignar estilos a los elementos web, como colores, tipos de letra, tamaños, etc. Los estilos se definen con CSS y luego se asocian a los elementos hasta que se obtiene el diseño visual que se desea aplicar en una página.

Por motivos de compatibilidad, los navegadores web aplican estilos por defecto a algunos elementos HTML. Esta es la razón por la que algunos algunos elementos en un sitio web puedan tener márgenes inesperados o saltos de línea, pero otros estando definidos de forma parecida muestran comportamientos dispares. Algunos de estos estilos son útiles, pero la mayoría deben ser reemplazados o complementados con estilos personalizados. En las hojas de estilo los diseños personalizados se declaran con propiedades; un estilo se define declarando el nombre de la propiedad y su valor separados por dos puntos.

```
font-size: 24px;
```

En el ejemplo anterior se declara una propiedad que cambia el tamaño de la letra a 24 píxeles, toda instrucción termina siempre con punto y coma “;”. Si esta propiedad se aplica a un elemento del sitio web, el texto contenido en él se mostrará en la pantalla con el tipo de letra definido por defecto, pero con un tamaño de 24 píxeles.

Respecto de los píxeles (px), aunque éstos son las unidades de medida que más se implementan, en algunas circunstancias, especialmente cuando se crean sitios web con diseño web adaptable a su contenedor, pueden ser más apropiadas otras unidades. En términos generales, las unidades más utilizadas son px (píxeles), pt (puntos), in (pulgadas), cm (centímetros), mm (milímetros), em (relativo al tamaño de letra del elemento), rem (relativo al tamaño de letra del documento), y % (relativo al tamaño del contenedor del elemento).



La mayoría de las propiedades se aplican a un solo aspecto del elemento. Si se quiere cambiar varios conceptos al mismo tiempo, es necesario declarar múltiples propiedades; el lenguaje CSS define una sintaxis que hace más sencillo el proceso de asignar múltiples propiedades a un elemento, característica que se conoce como “regla”. Una regla es una lista de propiedades declaradas entre llaves e identificadas por un selector; este último indica qué elementos se verán afectados por las propiedades. La siguiente regla se aplica a la etiqueta `p` y, por tanto, las propiedades que ahí se especifican se aplicarán a todos los elementos `<p>` del documento; esta regla incluye dos propiedades con sus valores agrupados por llaves. Si se aplica esta regla a un documento, el texto dentro de cada elemento `<p>` se mostrará en color rojo y con un tamaño de 24 píxeles.

```
p {  
  color: #FF0000;  
  font-size: 24px;  
}
```

2.3.1.2.- CSS y HTML

La gran diferencia entre el lenguaje de programación HTML y CSS son sus funciones. De hecho, ambos lenguajes no compiten entre ellos, pero sí se complementan. Por un lado el HTML permite armar la estructura de la web a través del etiquetado de hipertextos. Mientras que el lenguaje CSS permite asignar estilos y alterar las propiedades de los elementos HTML.

HTML ha tenido una serie de mejoras y actualizaciones desde que fue creado, llegando a la última versión de HTML5. Ningún lenguaje es perfecto, y si se profundiza en el desarrollo web, el usuario se podrá percatar de esto dependiendo del escenario en el que lo aplique.

En el caso del lenguaje HTML, estas son las ventajas y desventajas:

Ventajas de HTML:

- Fácil de usar.
- Sintaxis suelta (aunque ser demasiado flexible no cumplirá con los estándares).
- Bastante fácil de escribir.
- Permite el uso de plantillas, lo que facilita el diseño de una página web.
- Fácil de aprender y codificar incluso para programadores novatos.
- Compatible con casi todos los navegadores, si no con todos los navegadores.
- Ampliamente utilizado; establecido en casi todos los sitios web, si no en todos los sitios web.
- Muy similar a la sintaxis XML, que se utiliza cada vez más para el almacenamiento de datos.
- Gratis: no se necesita comprar ningún software.



Desventajas de HTML:

- No puede producir una salida dinámica por sí solo.
- A veces, la estructuración de documentos HTML es difícil de comprender.
- Los errores pueden resultar costosos.
- El tiempo que lleva para crear algunos elementos como listas y tablas puede ser excesivo.
- El desarrollador debe mantenerse al día con las etiquetas obsoletas y asegurarse de no utilizarlas. Las etiquetas obsoletas aparecen porque otro lenguaje que funciona con HTML ha reemplazado el trabajo original de la etiqueta; por lo tanto, es necesario aprender el otro idioma (la mayoría de las veces, es CSS).
- Las funciones de seguridad que ofrece HTML son limitadas.

CSS, al igual que HTML, tiene una función muy específica, y aunque hay herramientas que ayudan a aplicarlo de manera más productiva, el CSS en bruto tiene algunas áreas de oportunidad y sin duda grandes bondades.

Ventajas de CSS:

- Unificar todo lo referente al diseño visual en un solo documento.
- Se pueden hacer modificaciones en un solo lugar sin tener que recurrir a los archivos HTML por separado.
- Es menor la probabilidad de que exista duplicación de estilos en diferentes lugares, debido a esto es más fácil de organizar y hacer cambios. A esto se debe añadir que la información a transmitir es considerablemente menor y por tanto las páginas también se descargan más rápido.
- La creación de versiones para otros dispositivos: tablets o smartphones es muy simple.

Desventajas de CSS:

- Compatibilidad con algunos buscadores.

2.3.1.3.- Estilos en línea, embebidos, archivos externos

Las reglas y las propiedades definen los estilos que se asignarán a uno o más elementos, pero estos estilos no se aplican hasta que se agregan a un documento. Para ello existen tres alternativas:

- a) **estilos en línea:** usa atributo style para agregar estilos al elemento; toda etiqueta HTML cuenta con este atributo, y todos los estilos que ahí se apliquen se reflejarán posteriormente. ¿El problema? si se desea cambiar una característica gráfica del sitio, se debe hacer la modificación en todas las etiquetas, lo que incrementa el tamaño del archivo y complica su mantención posterior.



```
1  <!DOCTYPE html>
2  <html lang="es">
3    <head>
4      <title>Título del Documento</title>
5      <meta charset="utf-8">
6    </head>
7    <body>
8      <main>
9        <section>
10         <p style="font-size: 20px;">
11           Estilo aplicado directamente sobre un párrafo
12         </p>
13       </section>
14     </main>
15   </body>
16 </html>
```

- b) **estilos incrustados:** corresponde a agregar las reglas CSS dentro de la cabecera del documento usando atributos y valores que apliquen sobre las etiquetas a las que se desee aplicar un formato específico. Esto se debe hacer dentro de la etiqueta `<style>`. Lo que se declare dentro de la etiqueta es similar al caso del ejemplo que se indica antes; además, los estilos indicados aplican para todas las etiquetas correspondiente, no siendo necesario actualizarlas una por una. El inconveniente es que las mismas declaraciones se deberán repetir en cada documento si se desea mantener una gráfica uniforme.

```
1  <!DOCTYPE html>
2  <html lang="es">
3    <head>
4      <title>Título del documento</title>
5      <meta charset="utf-8">
6      <style>
7        p{
8          font-size: 20px;
9        }
10     </style>
11   </head>
12   <body>
13     <main>
14       <section>
15         <p>El estilo se aplica en todos los párrafos</p>
16       </section>
17     </main>
18   </body>
19 </html>
```

- c) **hojas de estilo:** esta acción corresponde a desplazar todos los estilos a un archivo con extensión CSS independiente y luego usar el elemento `<link>` para referenciarlo desde los documentos en los que estos estilos apliquen. Estos archivos se conocen como “hojas de estilo”, y solo contienen reglas provenientes del lenguaje aludido. Por cada elemento que se referencia, se pueden considerar parámetros dentro de la etiqueta `<link>`; en el caso de las



hojas de estilo solo se usan rel y href, que corresponden al valor "stylesheet" y la ruta del archivo CSS que contiene los archivos, respectivamente.

```
1  <!DOCTYPE html>
2  <html lang="es">
3    <head>
4      <title>Título del documento</title>
5      <meta charset="utf-8">
6      <link rel="stylesheet" href="hojadeestilos.css">
7    </head>
8    <body>
9      <main>
10        <section>
11          <p>
12            El estilo de este párrafo está declarado
13            en un documento independiente.
14          </p>
15        </section>
16      </main>
17    </body>
18  </html>
19
```

```
p {
  font-size: 20px;
}
```

2.3.1.4.- Referencias y selectores, por clase, por id

En un documento HTML, los selectores de clase buscan un elemento basado en el contenido de su atributo class. El atributo class está definido como una lista de elementos separados por espacio, y uno de esos elementos debe coincidir exactamente con el nombre de clase dado en el selector.

```
.classy {
  background-color: DodgerBlue;
}
```

```
<span class="classy">Aquí hay un span con algo de texto.</span>
<span>Aquí hay otro.</span>
```

Los selectores de ID, en cambio, buscan un elemento basado en el contenido del atributo id. El atributo ID del elemento seleccionado debe coincidir exactamente con el valor dado en el selector.

```
#identified {
```



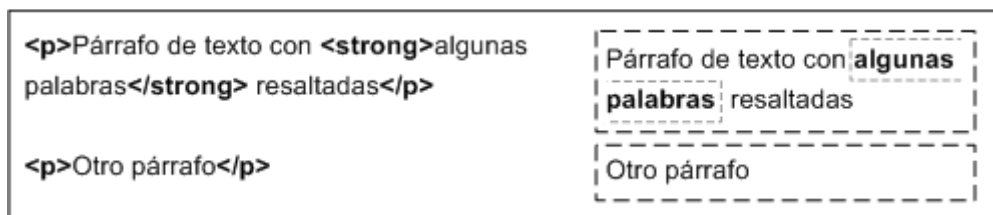
```
background-color: skyblue;  
}
```

```
<div id="identified">Este div tiene un ID especial!</div>  
<div>Este solo es un div regular.</div>
```

2.3.1.5.- El modelo de cajas

El modelo de cajas o "box model" es seguramente la característica más importante del lenguaje de hojas de estilos CSS, ya que condiciona el diseño de todas las páginas web. El modelo de cajas es el comportamiento de CSS que hace que todos los elementos de las páginas se representen mediante cajas rectangulares.

Las cajas de una página se crean automáticamente. Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento. La siguiente imagen muestra las tres cajas rectangulares que crean las tres etiquetas HTML que incluye la página:

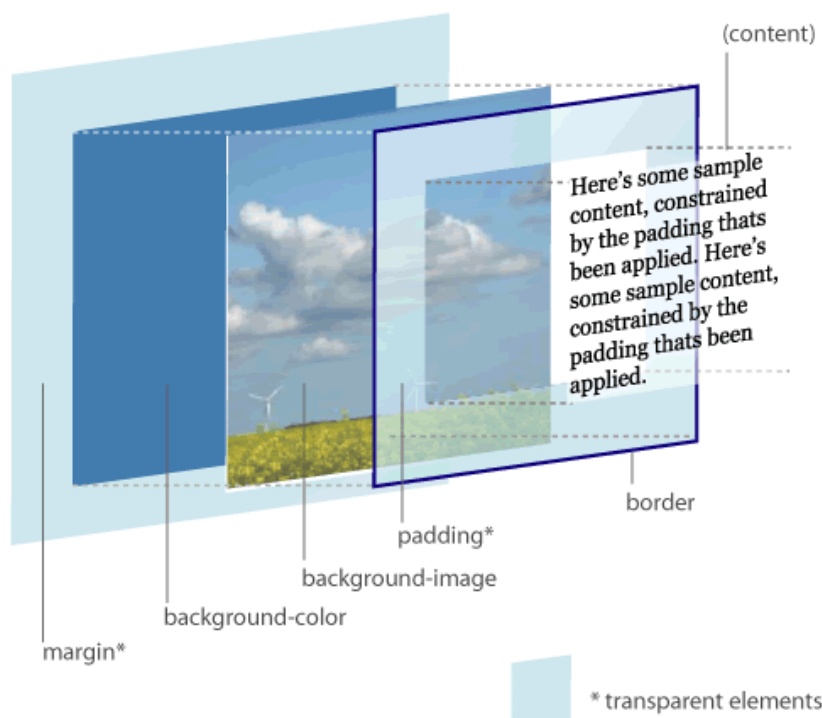


Las cajas de las páginas no son visibles a simple vista porque inicialmente no muestran ningún color de fondo ni ningún borde. La siguiente imagen muestra las cajas que forman la página web de <http://www.alistapart.com/> después de forzar a que todas las cajas muestren su borde:



Los navegadores crean y colocan las cajas de forma automática, pero CSS permite modificar todas sus características. Cada una de las cajas está formada por seis partes, tal y como muestra la siguiente imagen:

THE CSS BOX MODEL HIERARCHY





Las partes que componen cada caja y su orden de visualización desde el punto de vista del usuario son las siguientes:

- **Contenido (content):** se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)
- **Relleno (padding):** espacio libre opcional existente entre el contenido y el borde.
- **Borde (border):** línea que encierra completamente el contenido y su relleno.
- **Imagen de fondo (background image):** imagen que se muestra por detrás del contenido y el espacio de relleno.
- **Color de fondo (background color):** color que se muestra por detrás del contenido y el espacio de relleno.
- **Margen (margin):** separación opcional existente entre la caja y el resto de cajas adyacentes.

El relleno y el margen son transparentes, por lo que en el espacio ocupado por el relleno se muestra el color o imagen de fondo (si están definidos) y en el espacio ocupado por el margen se muestra el color o imagen de fondo de su elemento padre (si están definidos). Si ningún elemento padre tiene definido un color o imagen de fondo, se muestra el color o imagen de fondo de la propia página (si están definidos).

Si una caja define tanto un color como una imagen de fondo, la imagen tiene más prioridad y es la que se visualiza. No obstante, si la imagen de fondo no cubre totalmente la caja del elemento o si la imagen tiene zonas transparentes, también se visualiza el color de fondo. Combinando imágenes transparentes y colores de fondo se pueden lograr efectos gráficos muy interesantes.

2.3.1.6.- Estilos más utilizados

A veces es recomendable declarar los estilos en un archivo independiente y luego cargar ese archivo desde un documento que lo necesite, pero obliga al desarrollador a implementar diferentes mecanismos para establecer la relación entre las reglas CSS y los elementos dentro del documento que se verán afectados por éstas.

Existen varios métodos para seleccionar los elementos que serán afectados por una regla CSS. Ya se ha visto que es posible usar el nombre del elemento, pero también se pueden usar los valores de los atributos id y class para referenciar un solo elemento o un grupo de elementos, e incluso combinarlos para construir selectores más específicos.

<pre>span { font-size: 20px; }</pre>	Afecta a todos los elementos que estén contenidos dentro de la una etiqueta .
<pre>p, span { font-size: 20px; }</pre>	Asigna los mismos estilos a elementos con nombres diferentes; la regla afecta a todos los elementos <p> y encontrados en el documento.



<pre>main p { font-size: 20px; }</pre>	Afecta solo a los elementos <p> que se encuentran dentro de un elemento <main> , ya sea como contenido directo o insertados en otros elementos.
<pre>section > p { font-size: 20px; }</pre>	Usa el carácter > para referenciar un elemento que es hijo directo de otro. Indica que el elemento afectado es el elemento de la derecha cuando tiene como padre al elemento de la izquierda.
<pre>h1 + p { font-size: 20px; }</pre>	Este selector hace referencia a un elemento que está precedido por otro elemento; ambos deben compartir el mismo elemento padre. Afecta a todos los elementos <p> que se ubican inmediatamente después de un elemento <h1> .
<pre>p - p { font-size: 20px; }</pre>	Afecta a todos los elementos <p> que preceden a otro elemento <p> .
<pre>#mitexto { font-size: 20px; }</pre>	Se aplica al elemento identificado por el atributo id y el valor "mitexto". La ventaja de este procedimiento es que cada vez que se crea una referencia usando el identificador mitexto en un archivo CSS, solo se modifica el elemento con esa identificación, pero el resto de los elementos no se ven afectados. Esta es una forma muy específica de referenciar a un elemento y se usa comúnmente con elementos estructurales, como <section> o <div> .
<pre>.mitexto { font-size: 20px; }</pre>	Este atributo es más flexible y se puede asignar a varios elementos dentro del mismo documento. Para referenciar un elemento usando su atributo class , el selector debe incluir el valor del atributo precedido por un punto.

Las propiedades son la pieza central de un documento CSS. Todos los estilos que es posible aplicar a un elemento se definen a través de ellas; hay cientos de propiedades disponibles, y se pueden clasificar en dos tipos: propiedades de formato y propiedades de diseño. Las primeras se encargan de dar forma a los elementos y su contenido, mientras que las de diseño están enfocadas a determinar el tamaño y la posición de los elementos en la pantalla.

Las propiedades de formato se pueden clasificar según el tipo de modificación que producen. Por ejemplo, algunas propiedades cambian el tipo de letra que se usa para mostrar el texto, otras generan un borde alrededor del elemento, asignan un color de



fondo, etc. A continuación se indicarán algunas propiedades fundamentales de los elementos, clasificadas según su tipo.

Texto

<code>font-family</code>	Declara el tipo de letra que se usa para mostrar en un texto. Se pueden declarar múltiples valores separados por coma para ofrecer al navegador varias alternativas en caso de que algunos tipos de letra no se encuentren disponibles en el equipo del usuario. Valores estándar son Georgia , "Times New Roman" , Arial , Helvetica , "Arial Black" , Gadget , Tahoma , Geneva , Helvetica , Verdana , Geneva , Impact y sans-serif .
<code>font-size</code>	Determina el tamaño de la letra; puede estar indicado en pixeles, puntos, porcentaje o cualquiera de las opciones que entrega el lenguaje.
<code>font-weight</code>	Indica si el texto se mostrará en negrita o no. Los valores posibles son normal y bold , pero también se puede indicar en múltiplos de 100 para grosores de letra.
<code>font-style</code>	Permite establecer el estilo de la letra. Valores posibles: normal , italic , y oblique .
<code>font</code>	Permite declarar varios atributos al mismo tiempo, separándolos por un espacio y en un orden adecuado. El estilo y el grosor se deben declarar antes que el tamaño, y el tipo de letra al final (por ejemplo, <code>font: bold 24px Arial,sans-serif</code>).
<code>text-align</code>	Alinea el texto de un elemento. Valores posibles: left , right , center , y justify .
<code>text-align-last</code>	Alinea la última línea de un párrafo. Valores posibles: left , right , center , y justify .
<code>text-indent</code>	Establecer el tamaño de sangría en un párrafo, que es el espacio vacío al comienzo de la línea.
<code>vertical-align</code>	Alinea elementos de forma vertical. Se usa para alinear texto con imágenes, y la propiedad se aplica a la imagen. Valores posibles: baseline , sub , super , text-top , text-bottom , middle , top , y bottom .
<code>line-height</code>	Define espacio entre líneas. Valor se declara en pixeles, porcentaje, puntos, o en cualquiera de las medidas establecidas en CSS.



Colores

<code>rgb (rojo, verde, azul)</code>	Define un color por medio de los valores especificados por los atributos (desde 0 a 255): nivel de rojo, nivel de verde y nivel de azul, en ese mismo orden.
<code>rgba (rojo, verde, azul, alfa)</code>	Similar a rgb() , pero con un parámetro adicional que define la opacidad; este valor va entre 0 (totalmente transparente) y 1 (totalmente opaco).
<code>hsl (matiz, saturación, luminosidad)</code>	Define un color desde valores especificados por los atributos. Los valores se declaran en números decimales y porcentajes.
<code>hsla (matiz, saturación, luminosidad, alfa)</code>	Cumple una función similar a hsl() , pero considera un parámetro adicional: alfa u opacidad.
<code>color</code>	Declara el color del contenido del elemento.

A continuación se indica un ejemplo de cómo aplicar estilo de fuente a un elemento y un color.

```
#titulo {  
  font: bold 26px Verdana, sans-serif;  
  color: #CCCCCC;  
}
```

El color indicado anteriormente, #CCCCCC, es otra forma de representar un valor. Sigue la misma idea que el color RGB, y cada par de valores considera un color, y cada letra puede ir entre 0 - 9 y A – F. A este tipo de color se le denomina “color hexadecimal”.

Tamaño

<code>width</code>	Esta propiedad declara el ancho de un elemento. Se puede expresar en pixeles, porcentaje, puntos, etc.
<code>height</code>	Declara la altura de un elemento; se representa de igual forma en pixeles, puntos, porcentaje, centímetros, etc.
<code>margin</code>	Declara el margen de un elemento, y corresponde al espacio que hay alrededor de la caja. Puede recibir cuatro valores que representan el margen superior, derecho, inferior, e



	izquierdo, en ese orden y separados por un espacio (por ejemplo, margin: 10px 30px 10px 30px;).
padding	Declara el relleno de un elemento. Este último es el espacio entre el contenido del elemento y los límites de su caja. Los valores se declaran de la misma forma que lo hacemos para la propiedad margin .
overflow	Indica cómo se mostrará contenido que sale el elemento. Valores disponibles: visible (por defecto), hidden (esconde lo que no entra dentro de la caja), scroll (muestra barras para desplazar el contenido), auto (navegador decide).

Fondo

background-color	Asigna un color de fondo a un elemento.
background-image	Asigna una imagen de fondo a un elemento, referenciando la imagen con el método url() .
background-position	Declara el inicio de una imagen de fondo; los valores se pueden indicar en porcentaje, pixeles, o con las palabras clave center , left , right , top , y bottom .
background-size	Define el tamaño de la imagen de fondo; puede ser en pixeles o porcentaje, o bien usando las palabras clave cover o contain .
background-repeat	Determina si la imagen de fondo se repetirá, y en qué direcciones. Usa los valores repeat , repeat-x , repeat-y y no-repeat .

Bordes

border-width	Ancho del borde; acepta hasta cuatro valores separados por un espacio para especificar el ancho de cada lado del borde.
border-style	Indica estilos de bordes, cuatro valores separados por espacio. Valores posibles: none , hidden , dotted , dashed , solid , double , groove , ridge , inset , y outset .
border-color	Color del borde, cuatro valores separados.
border	Permite declarar todos los atributos del borde al mismo tiempo.
border-radius	Define el radio del círculo virtual que el navegador utilizará para dibujar las esquinas redondeadas.



Estas son solo algunas de las propiedades que se pueden definir a través de CSS. Desde la llegada de CSS3, se han agregado muchas opciones diversas para generar sitios web con diseños adecuados a las demandas actuales. Existen otras funciones más que es necesario al menos mencionar:

- **Sombras:** genera sombras para contenedores o textos; opciones posibles: box-shadow, text-shadow.
- **Gradientes:** se forma mediante una serie de colores que varían continuamente con una transición suave de un color a otro. Opciones posibles: linear-gradient, radial-gradient.
- **Filtros:** agregan efectos a un elemento y su contenido. CSS incluye la propiedad filter para asignar un filtro a un elemento y las siguientes funciones para crearlo. Opciones posibles: blur(), grayscale(), drop-shadow(), sepia(), brightness(), contrast(), hue-rotate(), invert(), saturate(), opacity().
- **Transformaciones:** Esta propiedad realiza cuatro transformaciones básicas a un elemento: escalado, rotación, inclinación y traslación. Opciones posibles: scale(), rotate(), skew(), translate(); también existen en versión 3D.
- **Transiciones:** Con la pseudoclase :hover se puede realizar transformaciones dinámicas. Sin embargo, una animación real requiere una transición entre los dos pasos del proceso. Para este propósito, CSS ofrece las siguientes propiedades: transition-property, transition-duration, transition-timing-function, transition-delay, transition.
- **Animaciones:** La propiedad transition crea una animación básica, pero solo se con dos estados en el proceso: el estado determinado por los valores actuales de las propiedades, y el estado determinado por los nuevos valores. CSS incluye las siguientes propiedades para componer animaciones más complejas: animation-name, animation-duration, animation-timing-function, animation-delay, animation-iteration-count, animation-direction, animation-fill-mode, animation.

2.3.1.7.- Buenas prácticas al construir una hoja de estilos

Al igual que otros lenguajes, CSS cuenta con un conjunto de “buenas prácticas” que se deben considerar al momento de asignar estilos a un sitio. Dentro de ellas se puede destacar las siguientes:

- **Limpiar el documento CSS:** en la red existen varias aplicaciones que permiten optimizar el tamaño de un documento CSS eliminando los caracteres que no son aporte de nuevos estilos. Ejemplo de ello son aplicaciones tales como Proccsor, Clean CSS y Unused CSS.
- **Combinar hojas de estilo:** cada hoja de estilos declarada como archivo independiente es una petición http. Independiente del tamaño de cada documento referenciado, cada llamado es tiempo utilizado. Por tanto, combinar todos los CSS en uno solo es una práctica que puede beneficiar bastante en lo que respecta al desempeño del sitio.



- **Agregar una tabla de contenido:** esto consiste en agregar en modalidad de comentario, en las primeras líneas de un documento CSS, las secciones que contendrá el archivo y la línea en la que comienzan, esto claro está será pertinente en documentos CSS con muchas líneas de código.
- **Escribir el CSS de manera estándar:** cada declaración de estilos se puede hacer de la forma estándar (cómo aparece en los ejemplos anteriores) o bien en una única línea; se recomienda usar la primera opción.
- **Usar un editor de texto:** se recomienda hacer las modificaciones del texto de un CSS a través de un editor de texto. En todo sistema operativo existen diversas alternativas.

2.3.1.8.- Manejo de assets e imágenes. Conociendo rutas absolutas y relativas

Un sitio web puede referenciar, para su correcto funcionamiento, archivos externos creados en otros lenguajes. Estos archivos tienen funciones específicas, y la distribución potencial (cantidad de archivos, largo de los archivos, tipo, etc.) lo define el programador. Para agregar un recurso externo a un sitio se puede usar la etiqueta `<link>`, aplicando una sintaxis similar al siguiente:

```
<link rel="value">
```

Dependiendo del tipo de recurso que se quiere importar, el atributo `rel` va tomando diversos valores:

- **icon:** importa un ícono a un documento.
- **stylesheet:** importa una hoja de estilos.
- **search:** entrega un enlace a un recurso que puede ser usado para buscar dentro de la página actual y en sus páginas relacionadas.
- **author:** entrega un enlace al autor del documento.
- **help:** provee un enlace para un sitio de ayuda.

En secciones anteriores se indicó que la etiqueta `` permite insertar imágenes en un sitio. Una imagen se puede referenciar a través de CSS, haciendo referencia al método `url()`. Por ejemplo, si se desea agregar una imagen de fondo a un sitio web a través de CSS, el código que se muestra a continuación puede ser de mucha ayuda.



```
1  body {
2      /* Referencia relativa a la imagen */
3      background-image: url(imagenes/fondo.jpeg);
4
5      /* Imagen de fondo centrada, tanto vertical como horizontalmente */
6      background-position: center center;
7
8      /* Posición de la imagen fija en la ventana */
9      background-attachment: fixed;
10
11     /* Sin repetición */
12     background-repeat: no-repeat;
13
14     /* Imagen de fondo se reescala automáticamente con el cambio del ancho
15     de ventana del navegador */
16     background-size: cover;
17
18     /* Se muestra un color de fondo mientras se está cargando la imagen
19     de fondo o si hay problemas para cargarla */
20     background-color: #cccccc;
21 }
22
```

Una imagen puede ser referenciada de dos maneras:

- **Referencia absoluta:** incluyen todas las partes de la URL (protocolo, servidor y ruta) por lo que no se necesita más información para obtener el recurso enlazado.
- **Referencia relativa:** no consideran algunas partes de las URL para hacerlas más cortas. Ya que son URL incompletas, es necesario disponer de información adicional para obtener el recurso enlazado; para que una URL relativa sea útil es imprescindible conocer la URL del origen del enlace.

2.3.1.9.- Orden jerárquico de aplicación de reglas CSS y el peso asociado a las reglas.

El sistema en cascada es una regulación para evitar conflictos entre diferentes hojas de estilo, ofreciendo jerarquía en función de su procedencia. Las instrucciones de estilo procedentes de fuentes diversas se guardan conforme a una jerarquía.

Puesto que las instrucciones de estilo se pueden vincular y combinar de muchas maneras distintas, hay que determinar el procedimiento a seguir en caso de producirse instrucciones contradictorias o conflictivas. Por ejemplo, si se tienen varias instrucciones asignando diferentes valores a una misma propiedad ¿a qué instrucción se debe dar prioridad?

Se denomina cascada al sistema jerárquico que otorga a las instrucciones de estilo de mayor o menor importancia en función de su procedencia. Es un sistema de regulación para evitar incongruencias entre diferentes hojas de estilo.



Cada navegador dispone de su propia hoja de estilo estándar (Browser-Style) que es la que utiliza para representar las páginas HTML. El tipo de letra, el color de fondo y el de la fuente, el color de los vínculos, el tamaño de los títulos, etc., los formatea el navegador.

Los navegadores modernos permiten al usuario crear sus propias hojas de estilo (hojas de estilo personalizadas, *userstyles*). El usuario puede, por tanto, configurar sus preferencias en el navegador y sobrescribir con ellas los ajustes de la hoja de estilo del navegador.

Las hojas de estilo creadas por nosotros, sobrescriben, por regla general, diferentes instrucciones tanto del navegador como del usuario (siempre que existan); las hojas de estilo se combinan. Si vincula varias hojas de estilo de autor a su página, en caso de conflicto siempre serán las instrucciones de la última hoja de estilo referenciada las que tengan preferencia. Su hoja de estilo personalizada, esto es, la hoja de estilo de autor, no puede ser sobrescrita por la del usuario, ya que la hoja de estilo de autor tiene mayor prioridad. Esta prioridad jerárquica sólo puede cambiarse mediante la regla *!important*.

Por otro lado, en el área **<head>** del documento se pueden incluir instrucciones de estilo definidas (incrustadas). Las instrucciones incrustadas sobrescriben las instrucciones externas o importadas, pero sólo en el documento actual (esto es, el documento en que están incrustadas las instrucciones).

Y las *“but not least”*, en la etiqueta HTML también podrían aparecer directamente (inline) instrucciones de estilo. Esas instrucciones tienen entonces prioridad ante todas las demás instrucciones, aunque sólo en la instancia actual (esto es, en la etiqueta HTML en que se han escrito las instrucciones).

2.3.2.- Responsividad

2.3.2.1.- El concepto de responsividad

La llega del iPhone en el año 2007 cambió la forma en que las personas se conectaban a Internet. Hasta ese momento, los dispositivos móviles tenían pequeñas pantallas con pocas opciones de interacción; en resumen, si se deseaba acceder a una página web, debía ser desde un computador personal. En ese entonces las resoluciones más populares consideraban al menos 1024 píxeles de ancho en una pantalla, suficiente para un computador de estas características. Lo anterior, por cierto, fue un problema con los primeros dispositivos inteligentes, ya que los sitios se tenían que desplegar en pequeñas ventanas sin posibilidad a adaptar su contenido.

Un diseño web adaptable o “responsivo” es usar HTML y CSS para redimensionar, ocultar, reducir o ampliar automáticamente un sitio web, a fin que se vea bien en todos los dispositivos (computadores de escritorio, tabletas y teléfono). No olvidar: una página web debe verse bien en cualquier dispositivo.



Para hacer una página de diseño adaptable, se recomienda considerar los siguientes puntos.

Configuración de la ventana gráfica

Todos los sitios web con diseño adaptable deben considerar la etiqueta **<meta>** en la cabecera respectiva. Esta etiqueta permitirá definir la ventana gráfica del sitio, dando instrucciones al navegador sobre cómo manipular las dimensiones manteniendo una relación de aspecto.

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

Imágenes responsivas

Son imágenes que se escalan automáticamente para adaptarse a cualquier tamaño del navegador. Si la propiedad de ancho de CSS se establece en 100%, la imagen se adaptará y escalará en ambas direcciones. Considere que si la imagen es más pequeña que el máximo de la ventana, se agrandará y no se verá correctamente. Como respuesta, se puede usar la propiedad max-width; si se establece en 100%, la imagen se adaptará a la ventana, pero no más de su tamaño original. A continuación se indican ambas maneras de modificar una imagen.

```



```

Mostrar diferentes imágenes según ancho de ventana

La etiqueta HTML **<picture>** permite definir distintas imágenes para diversos tamaños de ventana.

```
<picture>
  <source srcset="img_smallflower.jpg" media="(max-width: 600px)">
  <source srcset="img_flowers.jpg" media="(max-width: 1500px)">
  <source srcset="flowers.jpg">
  
</picture>
```

Texto con tamaño adaptable



El tamaño del texto puede ser establecido con una unidad “vw”, que significa “tamaño de la vista” (“viewport width”). En general, un 1 vw = 1% del ancho del viewport.

```
<h1 style="font-size:10vw">Hello World</h1>
```

2.3.2.2.- Tipos de dispositivos y orientaciones

En CSS se pueden declarar puntos de interrupción para definir estilos. Estos son estructuras que establecerán comportamientos dado un ancho máximo de ventana. Una idea bastante común, es crear puntos de interrupción para diseño orientado a tablets y orientado a celulares.



En el ejemplo a continuación, note que se indican dos clases que son casi idénticas (col- y col-s-).



```
/* Para teléfonos móviles: */
[class*="col-"] {
  width: 100%;
}

@media only screen and (min-width: 600px) {
  /* Para tabletas: */
  .col-s-1 {width: 8.33%;}
  .col-s-2 {width: 16.66%;}
  .col-s-3 {width: 25%;}
  .col-s-4 {width: 33.33%;}
  .col-s-5 {width: 41.66%;}
  .col-s-6 {width: 50%;}
  .col-s-7 {width: 58.33%;}
  .col-s-8 {width: 66.66%;}
  .col-s-9 {width: 75%;}
  .col-s-10 {width: 83.33%;}
  .col-s-11 {width: 91.66%;}
  .col-s-12 {width: 100%;}
}

@media only screen and (min-width: 768px) {
  /* Para equipos de escritorio: */
  .col-1 {width: 8.33%;}
  .col-2 {width: 16.66%;}
  .col-3 {width: 25%;}
  .col-4 {width: 33.33%;}
  .col-5 {width: 41.66%;}
  .col-6 {width: 50%;}
  .col-7 {width: 58.33%;}
  .col-8 {width: 66.66%;}
  .col-9 {width: 75%;}
  .col-10 {width: 83.33%;}
  .col-11 {width: 91.66%;}
  .col-12 {width: 100%;}
}
```

```
<div class="row">
  <div class="col-3 col-s-3">...</div>
  <div class="col-6 col-s-9">...</div>
  <div class="col-3 col-s-12">...</div>
</div>
```

En la ilustración anterior se indica un código HTML en el que se llama a estilos que definirán el comportamiento de acuerdo al tamaño de la ventana. Frente a esto se deben definir dos casos:



- **Para escritorio:** La primera y tercera sección tendrán tres columnas cada una, mientras que la central tendrá 6.
- **Para tabletas:** La primera sección considerará 3 columnas, la segunda abarcará 9, y la tercera se mostrará bajo las dos primeras secciones y considerará 12 columnas.

Puntos de interrupción más comunes

Existen diversidad de pantallas y dispositivos, de diferentes anchos y altos, lo cual hace que sea difícil definir puntos de interrupción para cada uno de ellos. A fin de simplificar escenarios, se ha dividido en cinco grupos.

```
/* Dispositivos pequeños (teléfonos, 600px y menos) */  
@media only screen and (max-width: 600px) {...}  
  
/* Dispositivos pequeños (tabletas verticales y teléfonos  
largos, 600px y más) */  
@media only screen and (min-width: 600px) {...}  
  
/* Dispositivos medios (tablets horizontales, 768px y más) */  
@media only screen and (min-width: 768px) {...}  
  
/* Dispositivos largos (laptops/escritorio, 992px y más) */  
@media only screen and (min-width: 992px) {...}  
  
/* Dispositivos extra largos (grandes laptops and equipos  
escritorio, 1200px y arriba) */  
@media only screen and (min-width: 1200px) {...}
```

Orientación: vertical / horizontal

Las “media queries” pueden ser usadas también para cambiar el diseño de una página dependiendo de la orientación del navegador. Se pueden considerar un conjunto de propiedades CSS que solo se aplicarán cuando la ventana del navegador sea más ancha que su altura o, dicho de otro, en orientación horizontal.

En el ejemplo siguiente, la página web tendrá un fondo azul claro si la orientación está en modo horizontal.



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

2.3.2.3.- El concepto Mobile First

El concepto “Mobile First” significa hacer diseños de sitios web pensando en dispositivos móviles antes de crear diseños que apunten a equipos de escritorio o en cualquier otro; esto hará que la página se muestre más rápido en dispositivos más pequeños. Lo anterior, por cierto, implica hacer cambios a nivel de CSS.

Por ejemplo, en vez de cambiar estilos cuando la ventana sea menor que 800px, se debería cambiar el diseño para cuando el ancho sea mayor a 800px. Las instrucciones en CSS quedarían de la siguiente manera:

```
/* Para teléfonos móviles: */  
[class*="col-"] {  
  width: 100%;  
}  
  
@media only screen and (min-width: 800px) {  
  /* Para escritorio: */  
  .col-1 {width: 8.33%;}  
  .col-2 {width: 16.66%;}  
  .col-3 {width: 25%;}  
  .col-4 {width: 33.33%;}  
  .col-5 {width: 41.66%;}  
  .col-6 {width: 50%;}  
  .col-7 {width: 58.33%;}  
  .col-8 {width: 66.66%;}  
  .col-9 {width: 75%;}  
  .col-10 {width: 83.33%;}  
  .col-11 {width: 91.66%;}  
  .col-12 {width: 100%;}  
}
```




En la ilustración anterior se muestra un ejemplo de cómo se vería el sitio con las configuraciones del CSS indicadas en un computador personal. Si se accede al sitio desde un Tablet o desde un teléfono móvil, se notará que el diseño se adaptará al tamaño de la ventana original.

2.3.2.4.- Utilización de Media Query

Adicional a cambiar el tamaño de texto y de las imágenes, es común usar el concepto de “media queries”. Con este tipo de consultas se pueden definir estilos para distintos tamaños de navegador.

En el ejemplo que se indica a continuación, se crean tres contenedores de distintos colores. Si el ancho de la pantalla es menor a 800px, los contenedores tendrán un nuevo ancho de 100%, y se ubicarán uno sobre otro; en caso contrario se mostrarán de forma horizontal.



```
<style>
.left, .right {
  float: left;
  width: 20%; /* El ancho de 20%, por defecto */
}

.main {
  float: left;
  width: 60%; /* El ancho es 60%, por defecto */
}

/* Use un "media query" para agregar un punto de quiebre
a los 800px de ancho de la ventana: */
@media screen and (max-width: 800px) {
  .left, .main, .right {
    width: 100%; /* El ancho es 100%, cuando el
contenedor es de 800px o menor. */
  }
}
</style>
```

En el ejemplo anterior se crean tres clases, una para cada celda que se desplegará. En un computador personal, se verían tres elementos situados uno al lado del otro, usando 20% del ancho de la ventana los pequeños, y 60% el más grande. En una Tablet y en otros dispositivos móviles y en caso que la ventana contenedora mida menos de 800px, las etiquetas <div> se posicionarán de forma vertical, una sobre otra, usando el ancho actual del contenedor.

2.3.2.5.- Cómo probar los distintos dispositivos

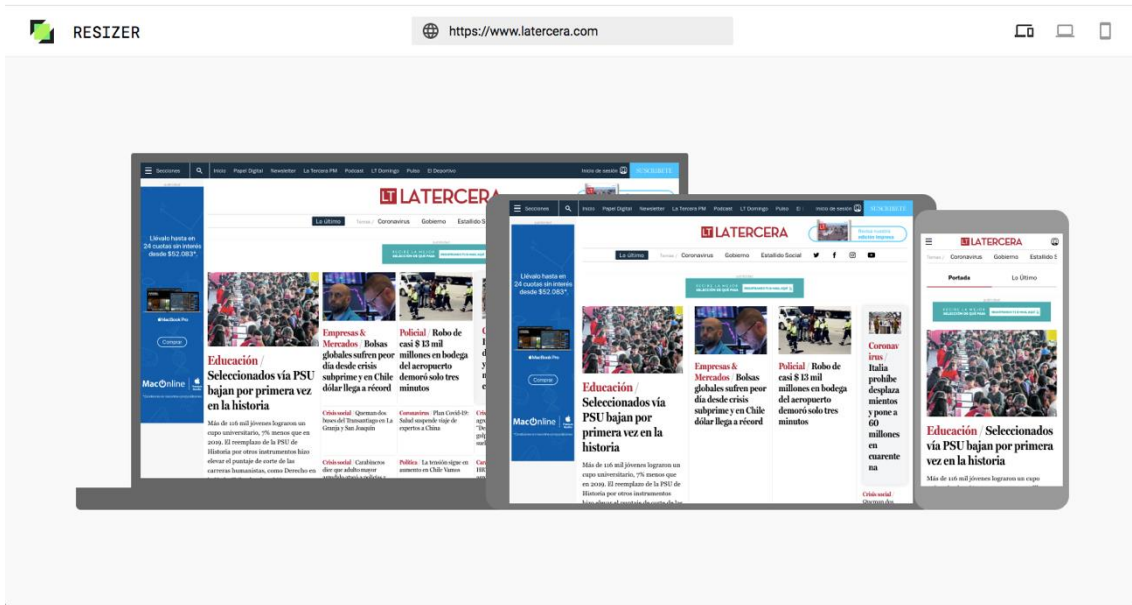
Hasta ahora se han analizado las bondades que provee el diseño adaptativo, y la forma en la que se debe implementar en un sitio. Una vez que se implementa un diseño de este tipo, se debe verificar que se adapte automáticamente a cualquier tamaño de dispositivo posible, en especial de aquellos que son más comunes en su uso.

En las herramientas que existen en la red, se destacan las siguientes:

Resizer

Este es un sitio web que permite validar el diseño de un sitio creado bajo la lógica responsiva. Para poder hacer uso de ella, el sitio a validar debe estar en la red, y se debe acceder a él ingresando su dirección en la barra de navegación, y presionar "Enter". Con esto se mostrará el sitio en los tres recuadros que aparecen más abajo. Cada recuadro es un tipo de dispositivo genérico.

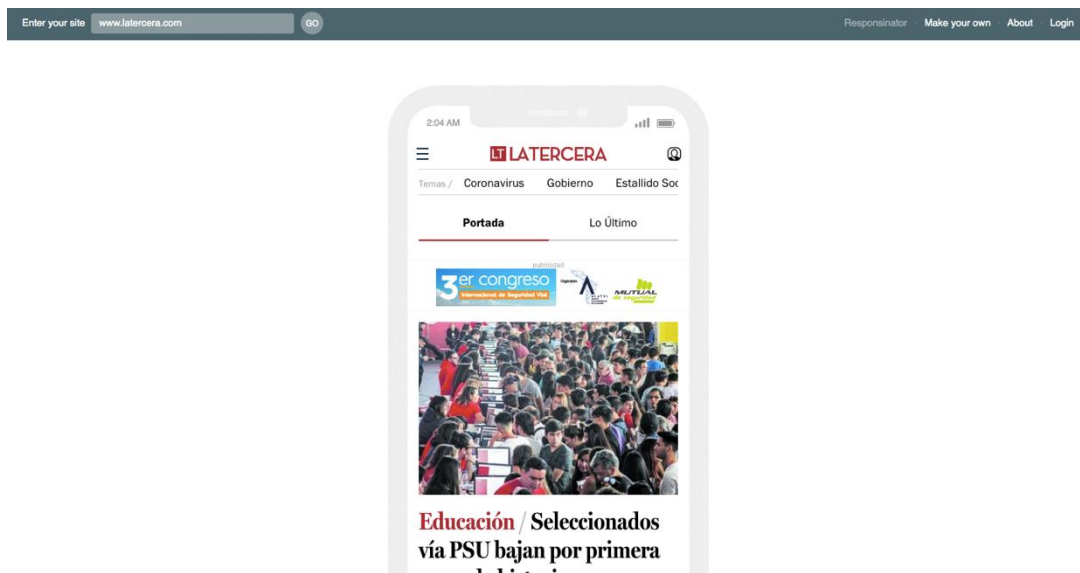
Dirección: <https://material.io/resources/resizer/>



Responsinator

Simula la representación de la web en un total de seis dispositivos móviles diferentes, en todos los casos considerando opción vertical y horizontal. Sigue la misma idea de la página indicada en el párrafo anterior.

Dirección: <http://www.responsinator.com/>

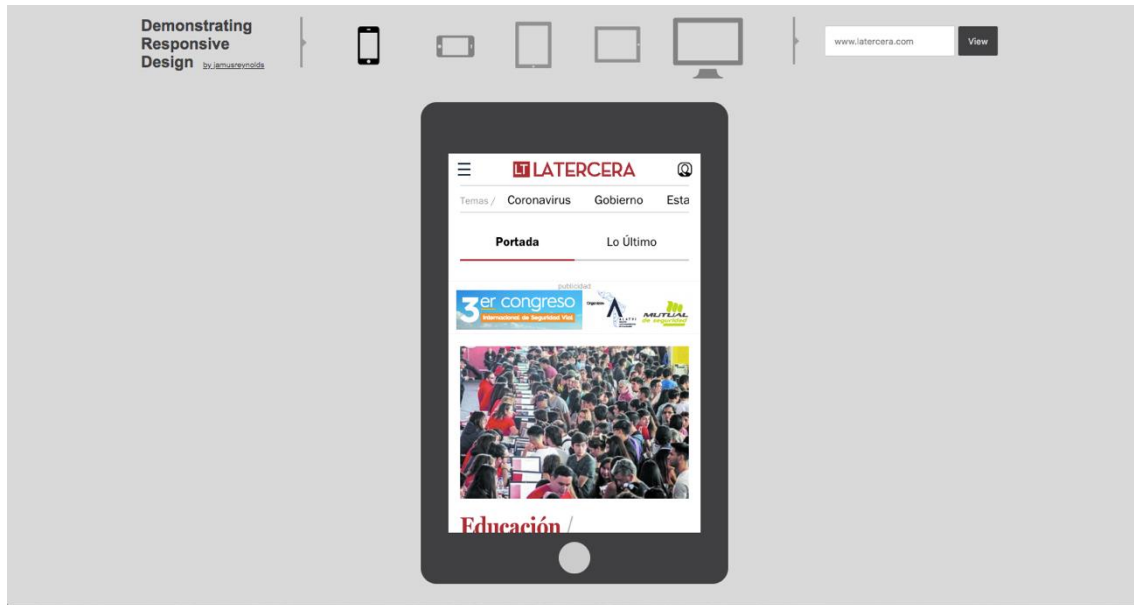


Demonstrating Responsive Design



En esta plataforma el usuario tiene solo a su disposición tres tamaños de pantalla. Independiente de esto, esta herramienta facilita el cambio rápido y claro entre diversos formatos.

Dirección: <http://www.jamus.co.uk/demos/rwd-demonstrations/>





Referencias

[1] ¿Sabes qué es CSS y cómo se relaciona con HTML?

Referencia: <https://www.crehana.com/cl/blog/web/sabes-que-es-css-y-como-se-relaciona-con-html/>

[2] Selectores de clases

Referencia: https://developer.mozilla.org/es/docs/Web/CSS/Class_selectors

[3] Selectores de identificadores

Referencia: https://developer.mozilla.org/es/docs/Web/CSS/ID_selectors

[4] Modelo de cajas

Referencia: <https://uniwebsidad.com/libros/css/capitulo-4>

[5] Jerarquía de sistema en cascada CSS para instrucciones

Referencia: <https://disenowebakus.net/jerarquia-de-estilos-en-css.php>

[6] Curso de CSS3 (en inglés)

Referencia: <https://www.w3schools.in/category/css3/>

[7] Animaciones en CSS3

Referencia: https://www.w3schools.com/css/css3_animations.asp

[8] w3schools - Diseño Web HTML adaptable

Referencia: https://www.w3schools.com/html/html_responsive.asp

[9] Responsive Test

Referencia: <https://www.ionos.es/digitalguide/paginas-web/disenio-web/herramientas-para-realizar-un-responsive-test/>