Ingeniería en Desarrollo de Software
Semestre 8

Programa de la Unidad didáctica:
Programación web II

Unidad 1. Tecnologías para mi sitio web

Ciudad de México, mayo de 2024

Clave:
15144843

Universidad Abierta y a Distancia de México

Índice

Unidad 1. Tecnologías para mi sitio web 3
Presentación de la Unidad 3
Logros 3
Competencia específica 4
1. Maquetación y estructura de HTML5 4
1.1. HTML5 y CSS3 6
2. Diseño responsivo para HTML 15
2.1. Funcionamiento del diseño web responsivo 16
3. Administradores de contenido 23
3.1. Ingeniería orientada a la reutilización de componentes 23
3.2. Funciones básicas de un gestor de contenidos 28
3.3. Arquitectura general de un gestor de contenido 30
3.4. Instalación de gestores de contenido 32
4. Alojamiento web 35
Cierre de la Unidad 44
Para saber más 45
Fuentes de consulta 46
Bibliografía complementaria 49
Fuentes de imágenes 50

Unidad 1. Tecnologías para mi sitio web

Presentación de la Unidad

En el mundo moderno las organizaciones hacen del uso de las tecnologías, y su principal estrategia para competir con un mercado a nivel mundial es mediante el uso de la web, la cual, en ocasiones, llega a ser la tarjeta de presentación de las organizaciones ante el mundo, y se convierte en la primera impresión de las organizaciones sobre sus posibles clientes. Por ello, es importante que la creación de un sitio web sea planificada estratégicamente, para captar a los usuarios y obtener los beneficios que esto representa para la supervivencia y ampliación del mercado. En este contexto, se puede afirmar que diseñar y maquetar correctamente una página web es indispensable para atraer la atención de los internautas, así como para que permanezcan en ella.

Como ingeniero en Desarrollo de software, es indispensable que aprendas a manejar todas las herramientas y programas necesarios para hacer de tu creación un producto rentable. Para ello, esta unidad se estructura con cuatro temas. En el tema *1. Maquetación y estructura de HTML5* te capacitarás para que sepas utilizar toda una serie de aplicaciones informáticas dedicadas al diseño y la maquetación, en un nivel profesional, mediante HTML5; asimismo, se expondrá la funcionalidad de los comandos avanzados de los lenguajes de programación HTML y CSS. Adquirirás una formación multidisciplinar que te permitirá no sólo planificar el diseño de una página, sino también llevarlo a la práctica de manera eficiente y fluida. En el tema *2. Diseño responsivo para HTML* se revisará básicamente la forma de definir una pantalla en diseño responsivo. Por su parte, en el tema *3. Administradores de contenido* se revisará lo correspondiente a la ingeniería orientada a la reutilización de componentes. Y en el tema *4. Alojamiento web* se te proporcionarán los elementos necesarios para que puedas comenzar a trabajar tu sitio web en un host.

Logros

Al término de esta unidad lograrás:

- Utilizar etiquetas HTML5 en tu diseño de páginas web.
- Aplicar el diseño responsivo para adaptar una página web a cualquier tipo de dispositivo de cómputo.
- Alojar páginas web en un host para estructurar un sitio.

Competencia específica

- Maquetar un sitio web para identificar la ubicación de los recursos requeridos mediante las herramientas e instrumentos dirigido a los requerimientos de una organización.

1. Maquetación y estructura de HTML5

En la actualidad cuando se habla de realizar la planeación de la estructura y ubicación de cada elemento que contendrá un sitio web, también conocido como interfaz de usuario, se está hablando de maquetar dicho sitio. La maquetación en HTML4 se realizaba con etiquetas ** ,** y en versiones que le anteceden, se realizaba mediante el uso de **tablas**, que permitían dar un esquema y/o estructura a las páginas y sitios web de entonces.

Maquetar es el proceso de situar a todos y cada uno de los elementos visuales que compondrán el sitio web dentro de la estructura del mismo, y que se conoce finalmente como **interfaz de usuario** o **front-end** (Aubry, 2012).

Para obtener más información acerca de cada una de las herramientas mencionadas, visita el sitio Trazos web (2022), donde podrás observar las diversas herramientas de maquetación y practicar con alguna de ellas.

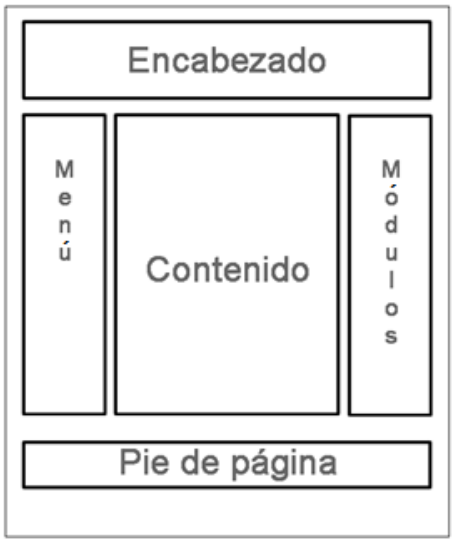
Según Rodríguez (2013, p. 35), “maquetar se puede definir como el acto de distribuir diferentes elementos, como el texto, imágenes, ilustraciones, etcétera, en cada página del documento designado, atendiendo a las necesidades del lector o del cliente”. La maquetación se puede realizar mediante herramientas libres y comerciales que permiten crear el prototipo gráfico del sitio, también conocido como *layout*, que brinda la facilidad de poder modificar o reubicar en cualquier momento algún elemento visual del sitio hasta que sea codificado.

Entre las mejores herramientas comerciales están **balsamiq**, **pidoco**, **mockflow** y **axure**; y entre las mejores herramientas libres están **prototyper** y **mockingbird.** Todas ellas permiten crear el prototipo gráfico del sitio, también conocido como ***layout***, que brinda la facilidad de poder modificar o reubicar en cualquier momento algún elemento visual del sitio hasta que sea codificado.

Otra opción con la que se puede crear el prototipo gráfico del sitio es de forma manual, es decir, en hojas blancas, en las que se planea y diseña la estructura del sitio, misma que es posible modificar en todo momento hasta llegar a una distribución final.

A continuación, se muestran ejemplos de maquetación que pueden tener los sitios o aplicaciones web.

Ejemplos de maquetado



Ejemplos de maquetado

Como se puede observar en las figuras anteriores, se hace uso de secciones o módulos que permiten estructurar el sitio. Entre los módulos o secciones más comunes se encuentran los siguientes:

- **Encabezado**. En esta sección se agregan, normalmente, el logotipo del sitio o aplicación, así como el nombre de la empresa o aplicación.
- **Menú**. Esta sección es la que brinda la navegabilidad del sitio; es un acceso directo a las diferentes secciones que componen en su totalidad el sitio.
- **Contenido.** En ésta se muestra la información concerniente a cada sección; es la sección principal del sitio.

- **Pie de página**. En esta sección se describen normalmente los derechos de autor y propiedad intelectual del sitio.

Los módulos mencionados pueden o no estar dentro de la estructura de un sitio web, mas no es obligatorio que deban estar o que deban integrarse los sitios con estos módulos, ya que la existencia de éstos depende de las necesidades y número de elementos visuales a incluir en el sitio a crear.

Una vez realizada la maquetación que tendrá el sitio o aplicación es necesario estructurarlo mediante el código, y para ello se hace uso del lenguaje HTML y CSS. En esta unidad se revisará el uso y estructura del lenguaje HTML5 y CSS3 para maquetar un sitio, temas que se exponen a continuación.

1.1. HTML5 y CSS3

El lenguaje HTML5, al igual que sus antecesores, es un lenguaje de marcado que permite crear páginas web, sólo que en esta versión se hace una definición de algunas etiquetas que permiten obtener un mejor diseño en la estructura del sitio.

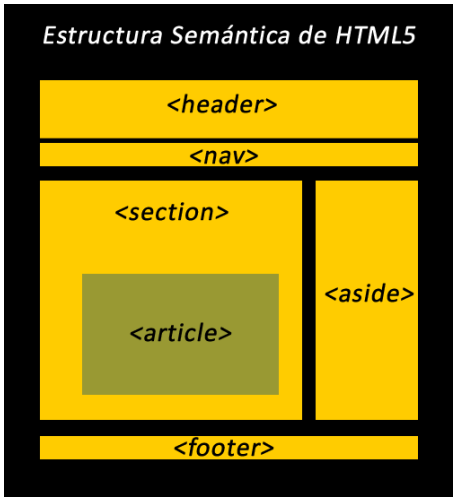
Esto no quiere decir que todas las etiquetas de HTML hayan sido cambiadas o modificadas, sino que existen nuevas etiquetas, y algunas fueron redefinidas, las cuales se explicarán a continuación.

Algunas etiquetas redefinidas de HTML para integrar HTML 5 (MSDN, 2014b) son las siguientes:

**Etiqueta **

La etiqueta en HTML es utilizada como una caja y permite agregar contenido en ella con el fin de crear una estructura para el sitio.

Ahora, con HTML5, existe una estructura de etiquetas semánticas que permiten definir el diseño de un sitio. La estructura básica de HTML5 es la siguiente:



Estructura básica de HTML5 (Desarrollo web.com 2012)

Donde:

<header>	**Sección que permite definir el encabezado o cabecera de la página.**
<nav>	Sección que permite establecer el navegador (Menú) principal de la página.
<section>	Establece una sección dentro de una página.
<article>	Especifica una unidad dentro de la página; es decir, un contenido.
<aside>	Establece contenido en un panel lateral de la página, también conocido como contenido tangencial.
<footer>	Establece el pie de página o información relacionada con los derechos de la misma.

Es muy importante mencionar que la etiqueta ** ** continua siendo de mucha importancia en HTML5, y que es la de mayor uso en conjunto con las ahora etiquetas semánticas de esta versión.

Para entender mejor el uso de estas nuevas etiquetas semánticas, a continuación se expone un ejemplo:

Se creará la definición básica de los elementos semánticos del lenguaje HTML5, con la finalidad de identificar la forma de declaración y uso dentro de un sitio.

```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4         <title> Título de la p&aacute;gina</tittle>
```

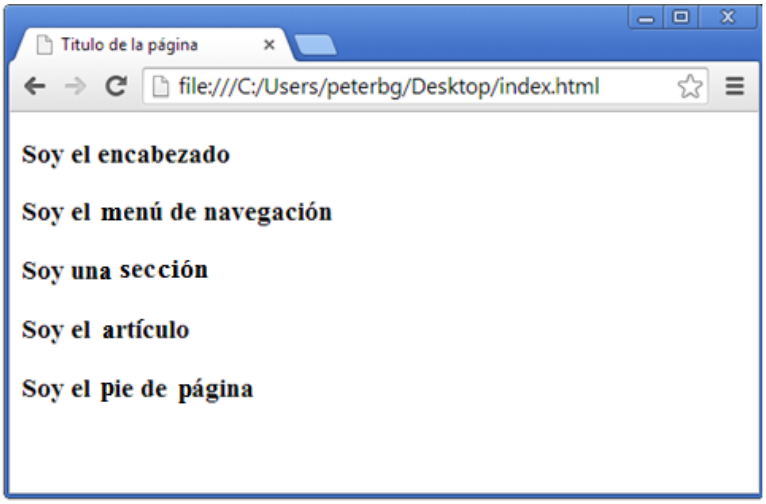
```
5 </head>
6 <body>
7          <header>
8.                  <h3>Soy el encabezado</h3>
9.         </header>
10.       <nav>
11.               <h3>Soy el Men&uacute; de navegaci&oacute;n</h3>
12.       <nav>
13.
14.       <div id="contenido">
15.             <section>
16.                <h3>Soy un Section</h3>
17.                <article>
18.                       <h3>Soy el Artículo</h3>
19.                       <br/><br/>
20.                 <article>
21.        </section>
22.        <aside>
23         <h3>Soy el ASIDE</h3>
24         </aside>
25         </div>
26
27         <footer>
28                 <h3>Soy el Pie de P&aacute;gina </h3>
29         </footer>
30         </body>
31         </html>
```

Ejemplo de utilización de etiquetas semánticas

En el ejemplo anterior se puede observar que la primera línea es:

<!DOCTYPE html>

Esta primera línea identifica el tipo de documento, mismo que representa uno de los cambios en esta versión de HTML5. Cuando se visualiza en el navegador, el ejemplo se observa de la siguiente forma:



Vista del resultado obtenido del ejemplo de utilización de etiquetas semánticas

Cómo se puede observar en la imagen anterior, se tiene la estructura de un sitio, es decir, se ha trasladado el esquema gráfico a código HTML5, sin embargo, éste aún no está completo, ya que no se ve como la imagen *Estructura básica de HTML5*, donde se observa la estructura semántica.

Para definir las reglas de presentación de cada etiqueta es necesario hacer uso de las hojas de estilo en cascada, o más conocidas como CSS en la versión 3.

CSS3 es la versión más reciente de las hojas de estilo en cascada, las cuales permiten definir reglas de presentación a cada etiqueta de HTML5, ya sea de forma directa o mediante el uso de clases o identificadores (MSDN, 2014a).

Para poder incluir una regla en CSS3 existe una sintaxis que se debe cumplir, y es la siguiente:

```
Selector { propiedad  : valor; }
```

Donde:

- ***Selector:*** Es el elemento al que se desea aplicar la regla.
- ***Propiedad****:* Es un característica que aplica para el elemento en cuestión; como todos los elementos tienen diferentes propiedades, se pueden consultar en las tablas de propiedades incluidas más adelante.
- ***Valor:*** Es la cantidad o porcentaje que se aplicará a la propiedad especificada; éstos son propios de cada una y se especifican en las tablas de propiedades.

También es posible identificar a los elementos por medio de clase o identificadores, como en el siguiente ejemplo:

```
h1    {    color    : RED; }
```

Donde se tiene el **selector <h1>** y todas las propiedades que se definan en selector, se **aplicarán** a todas las etiquetas <h1> que se encuentren en la página. Pero, si se desea que éste sólo aplique a ciertas etiquetas <h1>, es posible hacer uso de clases e identificadores, por ejemplo, asignar un nombre de clase a la etiqueta **<h1>** con la propiedad **class**=”nombre_de_clase”, es decir, *<h1 class=”amarillo”>,* con lo que es posible definir propiedades y valores sólo para esta clase, lo que se realiza de la siguiente forma:

```
.amarillo { color : Yellow; }
```

Como se puede observar, ya no se está haciendo uso del selector **h1**, sino que se utiliza el nombre de la clase para referenciar el selector.

De igual forma es posible **definir identificadores** para aplicar reglas de presentación a un selector. Esto se hace mediante el uso de la propiedad **id=”identificador”**, es decir, que al utilizar el selector **<h1 id=”verde”>**, quedaría de la siguiente forma:

```
#verde { color : Green; }
```

Como se puede observar en el ejemplo anterior, se inició con el símbolo **#,** que sirve para referenciar al selector con identificador verde.

La diferencia principal entre una clase y un identificador es que un estilo definido como clase se puede aplicar a múltiples selectores, y el estilo definido como identificador sólo puede aplicarse a un selector.

A continuación se explicarán las diferentes formas en que se pueden incluir los estilos en una página o sitio utilizando CSS3.

Es posible incluir CSS en una página o sitio de tres formas diferentes (MSDN, 2014a):

1. En forma directa: Esto quiere decir que la regla es declarada en la misma etiqueta, lo que afecta sólo al dominio de la misma, por lo que es recomendable utilizar este tipo de declaración cuando se requiera afectar contenido muy específico, por ejemplo:

```
<h1 style{color:red;}>Texto Rojo </h1>
```

Donde:
style: Permite definir estilos.
color: Propiedad
red: Valor de la propiedad

2. **En la cabecera de la página:** Es posible incluir estilos en la cabecera de la página, haciendo uso de la etiqueta <style> para delimitar éstos, por ejemplo:

```
<style type=”text/css”>
     h1{
         color: green;
 size: 10pt;
             }
             .textoimportante{
                 color:  red;
                 font-align: center;
                 }
        </style>
```

Donde:
<style></style>: Delimitan el inicio y final de la declaración de los estilos.
3. Utilizando un archivo externo: Es posible incluir estilos mediante el uso de un archivo externo y totalmente ajeno a la página. Esto se logra con la creación de un archivo que debe tener la extensión **nombre.css**, que, por lo general, se define como **estilo.css**. La utilización de este tipo de declaración es la más recomendable hoy en día, y es la que se recomienda para todo tipo de sitio o aplicación, ya que con el uso de un archivo externo se tiene el control del estilo del sitio. Para invocar a dicho archivo se hace uso de la siguiente instrucción:

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

Donde:
href="estilo.css" define el nombre y ubicación del documento que contiene la declaración de las reglas CSS3 que se aplicarán, mismas que se observan más adelante.

A continuación se expone un ejemplo realizado utilizando código en HTML5, en tanto que para el estilo se hace uso de código CSS3. El objetivo de éste es realizar la maquetación del sitio con la estructura que se muestra en la siguiente imagen:

Muestra de maquetación de sitio

Paso 1. Se creará un archivo de nombre **columna.html** donde se especificará la estructura de ésta mediante HTML5 de la siguiente forma.

```html
<!DOCTYPE html>
<html lang="es">
    <head>
        <meta charset="UTF-8">
        <title>Pagina a una columna</title>
        <link rel="stylesheet" href="css/estilocolumna.css">
    </head>
    <body>
        <header>
            <p>CABECERA</p>
        </header>
        <section>
            <article>CONTENIDO</article>
        </section>
        <footer>
            PIE
        </footer>
    </body>
</html>
```

Contenido de archivo columna.html

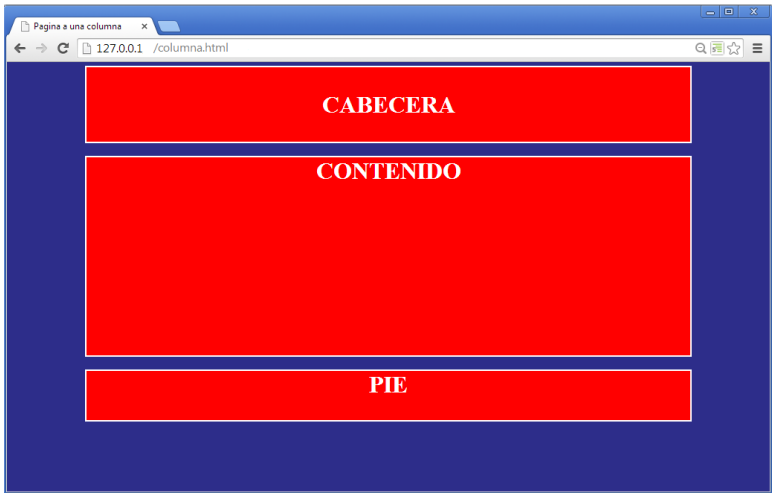
Donde es posible observar la definición de los elementos **<header>**, **<section>** y **<footer>.**

Paso 2. El archivo para la definición de estilos se nombrará como **estilocolumna.css**, y éste tendrá la siguiente declaración:

```css
body{
    background-color: #2D2D8A;
}
header, section, footer{
    background-color: #F00;
    width: 80%;
    margin:0px auto;
    text-align: center;
    font-size: 3em;
    color: #FFF;
    border: 4px solid #FFF;
    margin-bottom: 25px;
    height: 150px;
    font-weight: bold;
}
section{
    height: 400px;
}
footer{
    height: 100px;
}
```

Contenido de archivo estilocolumna.css

Paso 3. El resultado de los archivos anteriores es el siguiente:



Ejemplo de visualización archivo columna.html

Existe una gran cantidad de propiedades aplicables a los selectores de un sitio; las propiedades más comunes puedes consultarlas en el documento *U1. Propiedades de CSS*.

Retomando el ejemplo de **utilización de etiquetas semánticas** que se explicó anteriormente, se definirán a continuación las reglas de visualización de cada selector de la página, misma que se observó en la imagen **Ejemplo de utilización de etiquetas semánticas,** para poder llegar al formato deseado.

Se integrará un archivo externo de nombre estilo.css, en el que se definirán las reglas de los selectores con sus propiedades y valores correspondientes. El resultado de la definición de las reglas CSS3 es el siguiente:

```css
body{
    background-color: black;
    color: white;
}

header{
    width: 90%;
    background-color: #FFCC00;
    height: 80px;
    margin: 0px auto 0px;
    color:black;
    text-align: center;
    font-size: 2em;
}

nav{
    width: 90%;
    background-color: #FFCC00;
    height: 30px;
    margin: 0px auto 0px;
    color:black;
    text-align: center;
    font-size: 1em;
}

footer{
    width: 90%;
    background-color: #FFCC00;
    height: 30px;
    margin: 0px auto 0px;
    color:black;
    text-align: center;
    font-size: 1em;
}
```

```css
section{
    width: 55%;
    background-color: #FFCC00;
    color:black;
    text-align: center;
    font-size: 1em;
    border: 2px solid #FFF;
    margin-top: 15px;
}
aside{
    width: 38%;
    background-color: #FFCC00;
    color:black;
    text-align: center;
    font-size: 1em;
    border: 1px solid #FFF;
    margin-left: 15px;
    margin-top: 15px;
    position: absolute;
}
section, aside{
    display: inline-block;
}
article{
    width: 90%;
    border:1px solid #FFF;
    margin: 0 auto;
    background-color: #999933;
}
#contenido{
    width: 90%;
    margin: 0px auto 0px;
    color:black;
}
```

Ejemplo de código archivo estilo CSS

Como se puede observar en el ejemplo anterior, se hace uso de múltiples propiedades de CSS3, las cuales encuentran en el documento *U1. Propiedades de CSS.* El resultado obtenido del código que se ejemplifica anteriormente es el siguiente:

Resultado de maquetación con HTML5 y CSS3

De esta forma es como se concluye el tema de maquetación y estructura del lenguaje HTML5. Para obtener más información y ampliar los temas investiga diversos sitios en internet o en otras fuentes accesibles, por ejemplo, los siguientes sitios: Maquetando el monstruo web (Acámica, s/f) y Aprendiendo ando (Sancho, M., 2013).

2. Diseño responsivo para HTML

Según Frain (2012), el término diseño web adaptativo, conocido como RWD (por sus siglas en inglés, ***Responsive Web Design*),** fue acuñado por Ethan Marcotte en su artículo llamado *Responsive Web Design* (Marcotte, 25 de mayo, 2010). El término, señala el autor, se usa para hacer referencia a un diseño flexible, fluido, elástico, etc., y se refiere a la posibilidad de adaptar el contenido de una página o sitio web a las necesidades de visualización de los diversos dispositivos móviles.

Se ha nombrado también al diseño web responsivo como:

> "Diseño web adaptable o adaptativo […] es una técnica de diseño y desarrollo web que, mediante el uso de estructuras e imágenes fluidas, así como ***media-queries*** (módulo CSS3 que permite la representación de contenido para adaptarse a condiciones como la resolución de pantalla) en la hoja de etilo CSS, consigue adaptar el sitio web al entorno del usuario.
>
> La capacidad de la web de adaptarse a todos los dispositivos que la accedan (móviles, tabletas, PC, etcétera) y visualizarse de una manera óptima en cualquiera de ellos" (Ramos y Ramos, 2014, p. 145).

Con el surgimiento de los dispositivos móviles en el entorno web, se observó la necesidad de crear páginas web adaptadas y las URL **m.dominio**, primero con lenguajes de marcaje diferentes al HTML, y después, con la llegada de los Smartphones, adaptando los estilos CSS, sucedía que era necesario desarrollar dos sitios web distintos (Tajadura, 2013).

Mediante el diseño responsivo se pueden crear páginas que pueden ser visualizadas en todo tipo de dispositivos, como equipos de cómputo de escritorio, laptops, netbooks, tablets, smarthphones, entre otros, mediante el uso de estructuras e imágenes fluidas. Con este tipo de diseño no se necesita tener una versión desarrollada para cada dispositivo, ya que una sola página desarrollada, con una metodología responsiva, se adapta a todos ellos.

La principal ventaja de implementar el diseño responsivo es la reducción de costos en las fases de creación y mantenimiento (un solo diseño es válido para cualquier formato). No se duplica el contenido y se evita el redireccionamiento, ya que se utiliza sólo una URL en los navegadores. Con una sola versión en HTML y CSS se cubren todas las resoluciones de pantalla, es decir, el sitio web creado estará optimizado para todo tipo de dispositivos como, computadoras personales, tabletas, teléfonos móviles, etcétera. Esto mejora la experiencia de usuario a

diferencia de lo que ocurre, por ejemplo, con sitios web de ancho fijo cuando se acceden desde dispositivos móviles.

2.1. Funcionamiento del diseño web responsivo

Antes de que surgiera el diseño responsivo, los diseñadores de páginas web tenían que crear una página web para cada dispositivo, llámese tablet, laptop, netbook, Smart TV, entre otros. Con el diseño responsivo, sin embargo, en lugar de construir un sitio web para cada tipo de dispositivo, ahora se crea un único sitio web utilizando **CSS3 media queries** y un ***layout*** con imágenes flexibles. De esta forma, el sitio web detecta desde qué clase de dispositivo está accediendo el usuario y muestra la versión más optimizada para ese medio, reorganizando los elementos y ocultando algunos de ellos como imágenes o imágenes más ligeras, redistribuyendo las columnas en el diseño u ocultando texto, entre otras adaptaciones, según el dispositivo al que se tenga que adaptar dicho sitio web.

Mediante el lenguaje HTML5 es posible **ahorrarse** el costo que representa desarrollar una APP nativa para cada dispositivo (Quesada, 2013). Los estilos CSS son parte fundamental para visualizar las páginas web mucho más profesionales.

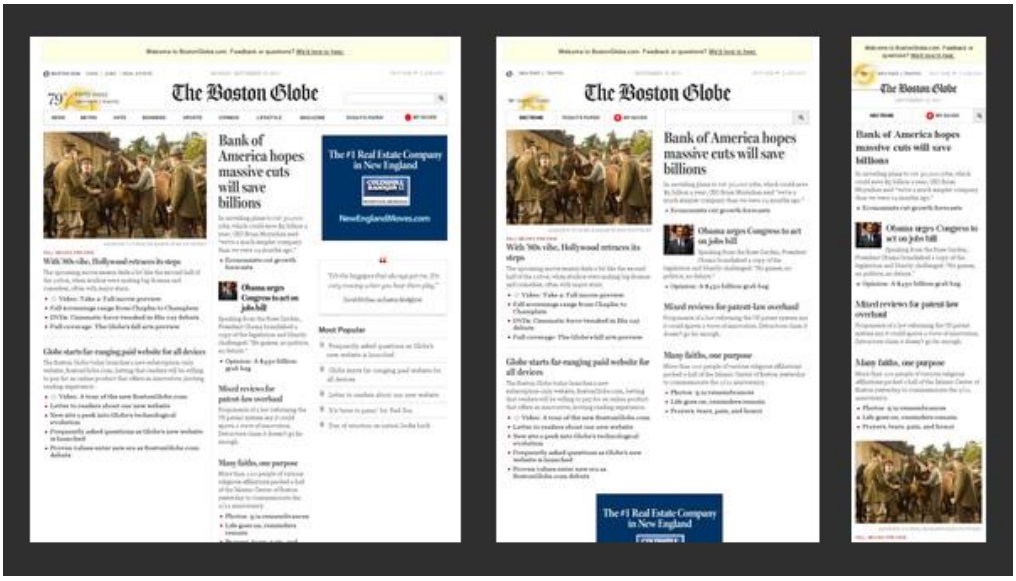
La nueva versión de CSS3 contiene unos elementos llamados **media queries** cuya función es indicarle al navegador cómo debe verse la página web en los diferentes tamaños de resolución (Tajadura, 2013).

Por lo general, se han establecido tres estándares:

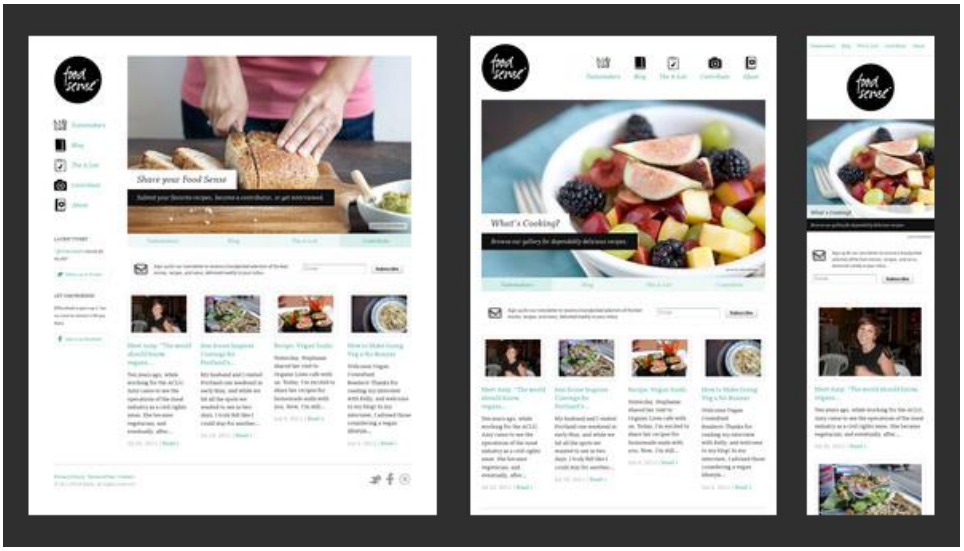
- Para **dispositivos móviles** entre 320px y 561px
- Para **tablet** entre 561px y 800px
- Para **computadoras personales** PC superior a 800px

A continuación se muestran algunos ejemplos de visualización de páginas web con diseño responsivo:

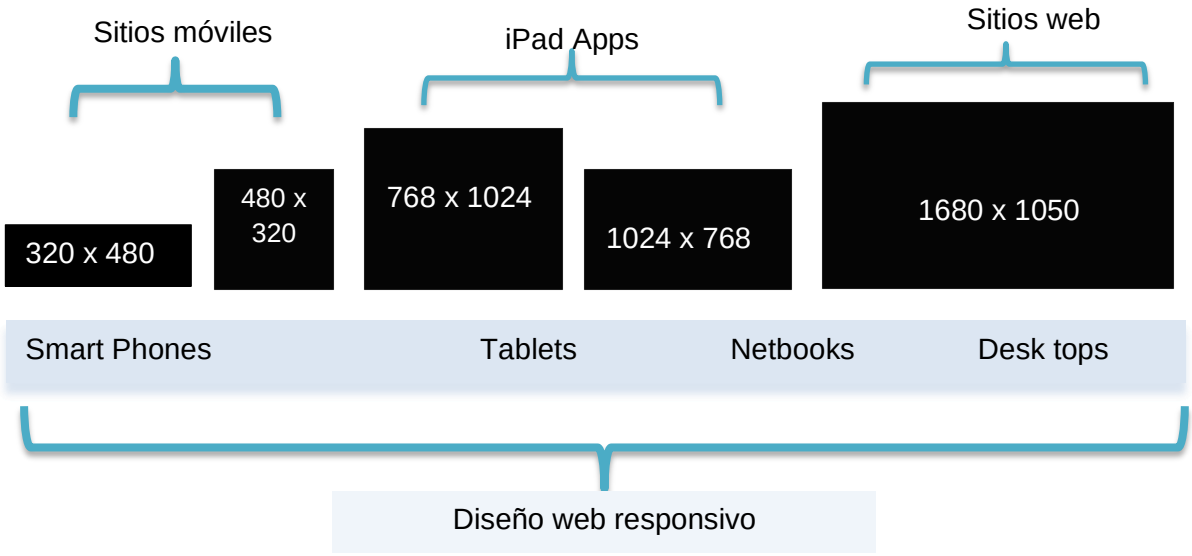
Diseño web responsivo en donde se observa la misma página web adaptada a diferentes tamaños para diferentes dispositivos (Llano, 2012).



Ejemplo de diseño responsivo del *Diario The Boston Globe*, en donde se observa la misma página web adaptada a diferentes tamaños para diferentes dispositivos (Llano, 2012).



Ejemplo de diseño responsivo de una página web de cocina (Llano, 2012).



Resoluciones de pantallas según el dispositivo (Diez Media, 2014)

Como ya se mencionó anteriormente, mediante el diseño web responsivo el mismo sitio web puede ser visto desde la pantalla de un BlackBerry Curve, cuyo tamaño y resolución son de 2.4 pulgadas, con 320 x 240 pixeles; de un Samsung Galaxy Note, con 5.3 pulgadas de tamaño y una resolución de 1280 x 800 pixeles; del nuevo iMac, con 27 pulgadas de tamaño y 2560 x 1440 pixeles; de un MacBook Pro Retina, con 13.3 pulgadas y 2560 x 1600 pixels; o un televisor Samsung de 65 pulgadas, con 1920 x 1080 (Gagliardo, 2014).

Unidades de medida

Para realizar un diseño responsivo es necesario conocer las **unidades de medida**. Todas las medidas pueden expresarse con un máximo de tres decimales, y no importa que en Sudamérica y Europa los decimales se separen por comas. Los estándares web son de Estados Unidos, así que, como se utiliza en ese país, en el procedimiento de diseño las medidas se **separan por puntos.**

- **Unidades absolutas.** Las unidades absolutas son independientes de la pantalla, de la ventana del navegador, y de cualquier otro elemento en la estructura de la página. No requieren calcularse con base en ningún otro valor de referencia. Según Peña (2013), "consisten en unidades de medida físicas y tienen una relación fija y bien definida unas relativas a otras. Son unidades no escalables, a excepción del porcentaje, que estrictamente hablando no sería una unidad absoluta estándar, aunque así es generalmente considerada." Estas son:
    - Centímetros: cm
    - Milímetros: mm
    - Pulgadas: in
    - Puntos: pt (equivale a 1/72 pulgadas)
    - Picas: pc (equivale a 12 puntos)

- **Unidades relativas**: Las unidades relativas son aquellas que se calculan tomando, como referencia, algún otro valor. Según Peña (2013), "especifican un tamaño relativo a otra longitud externa. Las unidades relativas son fácilmente escalables". Estas son:

Unidad relativa	Especificación
em	Relativo al tamaño de letra (equivale al ancho de la letra "M"; según el tamaño de letra que se use, este valor es directamente el tamaño de la fuente).
ex	Similar al em, pero equivale a la altura de la letra "x".
ch	Similar al em y al ex, pero equivalente al ancho del "0" (cero).
rem	Igual a em, pero siempre corresponde al tamaño de fuente del elemento **root**.
px	Relativo a la resolución de la pantalla.
%	Puede ser relativo al tamaño de letra, o al tamaño del elemento padre, dependiendo de su uso.
vw	Relativo al ancho del viewport (corresponde al 1% de su ancho).
vh	Similar a vw, pero relativo a la altura del viewport.
vmin	Corresponde al más chico de los valores, entre vw y vh.
vmax	Corresponde al más grande de los valores, entre vw y vh.

Unidades relativas (Gagliardo, 2014)

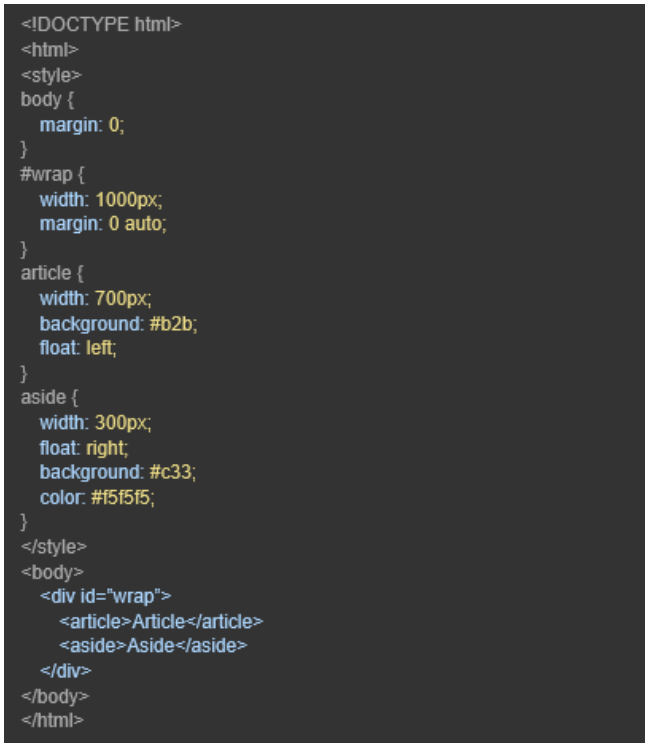
La principal ventaja de las medidas absolutas es que un elemento con una medida absoluta se verá exactamente del mismo tamaño, sin importar dónde se visualice. Se podría poner una regla sobre la pantalla y medir que, efectivamente, mide la cantidad de pulgadas que se desea. Peña (2013) los describe como unidades absolutas, ya que corresponden a 1/96 pulgadas, o lo que es lo mismo, 0.75pt. Sin embargo, esta medida sólo es correcta en equipos con 96 dpi. Por lo tanto, los pixeles siguen siendo unidades relativas (Gagliardo, 2014). En el diseño web responsivo no se utilizan con frecuencia estas unidades, pues interesa más la elasticidad del diseño. Las unidades relativas, en cambio, son las que se utilizan más.

Medidas relativas. Entre sus nuevas características, CSS3 tiene la posibilidad de **usar expresiones matemáticas** para **calcular** los **valores** que se necesiten. El uso de **calc()** es muy sencillo; únicamente se introducen entre los paréntesis todas las operaciones (acepta sumas, restas, multiplicaciones y divisiones) necesarias para conseguir la medida que se requiere. Además, los cálculos pueden hacerse con todas las otras unidades de medida. Por ejemplo:

```
width: calc(100% - 5em)
```

Con **calc()** pueden conseguirse *layouts* mucho más precisos e inteligentes. Y no tiene problemas en **combinar** unidades **absolutas** con **relativas**. Se puede conseguir que un ** ** mida siempre el 50% del ancho, más 200px, o cualquier otra medida necesaria. **calc()** hace posible **c**onseguir ubicaciones muy específicas, lo que se lograba con **paddings** y **anidamientos .**

Diseño fluido. El diseño web fluido se basa, principalmente, en **representar** las **medidas** como **proporciones**. De este modo, el diseño de la página se verá siempre **proporcionalmente** igual, sin importar el tamaño, o resolución, de la pantalla. Para conseguir esto, normalmente se utiliza el porcentaje como medida principal. El modo más sencillo de comenzar con un diseño fluido es tomar un diseño fixed (el que tiene sus medidas estáticas, en pixeles). Por ejemplo, un diseño pensado para un monitor de 1280px de ancho, que podría ser como se observa en el siguiente código**:**



```
<!DOCTYPE html>
<html>
<style>
body {
   margin: 0;
}
#wrap {
   width: 1000px;
   margin: 0 auto;
}
article {
   width: 700px;
   background: #b2b;
   float: left;
}
aside {
   width: 300px;
   float: right;
   background: #c33;
   color: #f5f5f5;
}
</style>
<body>
   <div id="wrap">
      <article>Article</article>
      <aside>Aside</aside>
   </div>
</body>
</html>
```

Diseño responsivo para un monitor de 1280px de ancho (Gagliardo, 2014)

Para convertir las medidas estáticas, se realiza el cálculo de la siguiente forma: `700 (medida de article) / 1000 (medida de #wrap)` y el resultado indica que `"article"` mide `0.7 (ó 70%)` de lo que mide `#wrap`.

Pasos para definir el tamaño de pantalla en diseño responsivo:

1. Identificar los dos componentes de la **query**:
   - Un media type (screen), y
   - La consulta entre paréntesis, conteniendo una característica a inspeccionar **(max-device-width),** seguida por el valor requerido (`480 pixeles`), donde se le está preguntando al dispositivo si su resolución horizontal, **`max-device-width`**, es igual o menor que `480px`.

2. Si la respuesta es correcta, si la resolución del dispositivo es <= 480 pixeles, quiere decir que se podrá visualizar la página en un dispositivo con una pantalla pequeña como el iPhone; entonces, el dispositivo cargará `shetland.css`.

3. Si la respuesta es que la resolución del dispositivo es >= a 480 pixeles, entonces, el `link` es completamente ignorado.

4. Es posible testear múltiples valores de las propiedades en una sola query, encadenándolos con la palabra clave **`and`**:

Ejemplo:

```
<link rel="stylesheet" type="text/css" media="screen and (max-device-width: 480px) and (resolution: 163dpi)" href="shetland.css" />
```

5. Incluir las **media queries** en el CSS como parte de una regla `@media`, sin limitarse a incorporarlas en el tag `link`. Por ejemplo:

```
@media screen and (max-device-width: 480px) {
  .column {
    float: none;
  }
}
```

Es posible incluir las media queries también como parte de una directiva **`@import`**, como se observa a continuación:

```
@import url("shetland.css") screen and (max-device-width: 480px);
```

En cada caso, el efecto es el mismo: si el dispositivo pasa el test planteado por la **media query,** el CSS que corresponda es aplicado al código (Marcotte, 2010).

Imágenes en diseño responsivo. En cuanto a las imágenes en la base de la página web que se desarrolle, en su *layout*, por defecto, el CSS se estructura de la siguiente forma:

```
.figure {
  float: left;
  margin: 0 3.317535545023696682% 1.5em 0;    /* 21px / 633px */
  width: 31.121642969984202211%;              /* 197px / 633px */
}
li#f-mycroft,
li#f-winter {
  margin-right: 0;
}
```

"Cada elemento **`.figure`** tiene un tamaño de aproximadamente un tercio de la columna que lo contiene, con el margen derecho en cero para las dos imágenes al final de cada fila (`li#f-mycroft, li#f-winter`). Y esto funciona bastante bien hasta que la vista es o

notablemente más pequeña o más ancha que el diseño original. Con las media queries se pueden aplicar arreglos específicos para una resolución, adaptando el diseño para responder mejor a los cambios en la pantalla" (Rigó, 2013).

Para revisar dos ejemplos de diseño responsivo, investiga en algunos recursos en internet; por ejemplo, los siguientes: *Diseño web responsivo* (Marcote, 2010) y *Diseño web adaptable* (Rigó, 2013).

3. Administradores de contenido

Un sistema de gestión o de administración de contenidos o CMS (*Content Management System*) por sus siglas en inglés, permite crear, organizar y controlar tanto la publicación como la presentación de documentos y otros recursos digitales en diferentes formatos multimedia a través del trabajo colaborativo de varios usuarios que tienen asignadas diferentes responsabilidades como autores, editores, revisores y administradores del sitio web, con el fin de mantener el contenido del sitio web dinámico y actualizado sin la necesidad de que estos usuarios tengan conocimientos avanzados de programación web.

"Un SGC es un software que se encuentra en un servidor web al que se le incorporan, comúnmente, una o varias bases de datos, lo que permite al desarrollador crear, administrar y actualizar páginas o sitios web de manera simple y dinámica, a través de interfaces gráficas y plantillas de diseño predefinidas" (Baltazar, 2006).

Los CMS son una opción que permitirá crear sitios web de manera muy rápida una vez que se conocen ampliamente los requerimientos del cliente y que se ha maquetado el sitio a desarrollar.

3.1. Ingeniería orientada a la reutilización de componentes

Respecto a las tecnologías web con las que se cuenta para el desarrollo web, una opción para crear un sitio es utilizar lenguajes básicos como HTML y las CSS. Para poder explotar al máximo estas herramientas se debe contar con conocimientos de diseño, ergonomía web y programación, ya que se comenzará desde cero, o tal vez se requiera modificar un sistema ya existente.

Otra opción es utilizar un software como **Dreamweaver** que, a través de herramientas gráficas, generará el código HTML y CSS. Nuevamente, será necesario tener conocimientos especializados en lenguajes de programación si se desea un sitio web dinámico, fácil de usar y flexible.

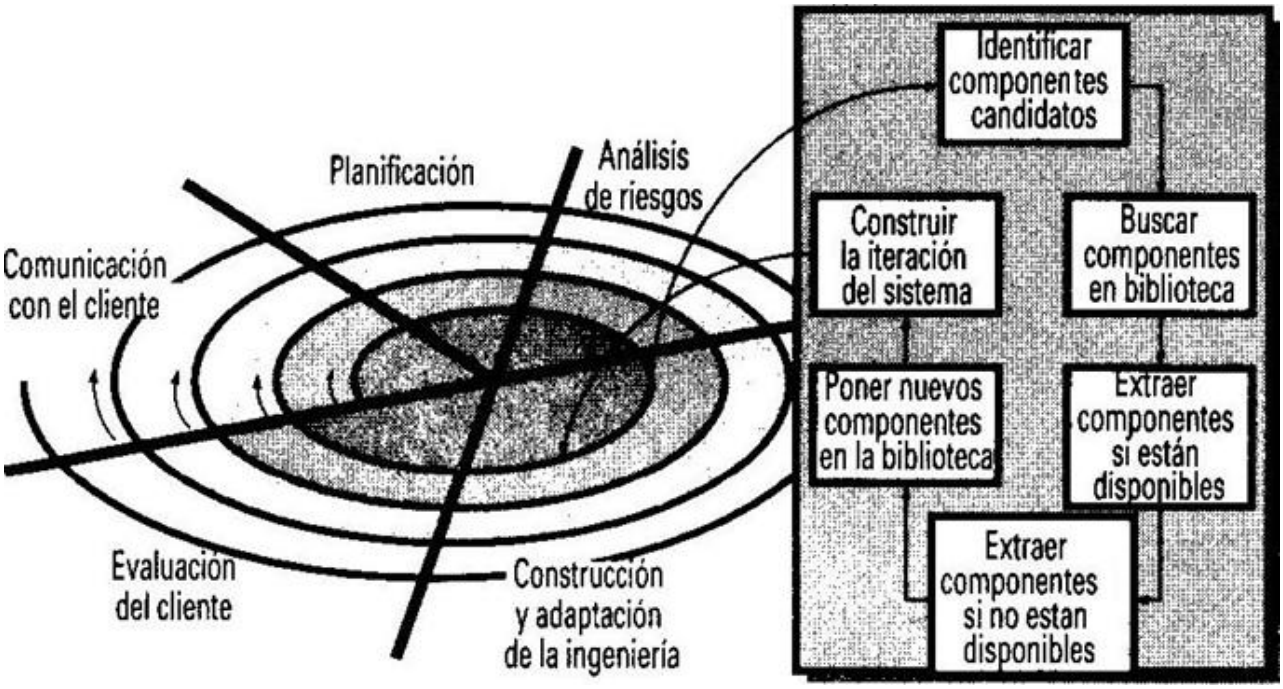
Estas dos soluciones requieren de un alto grado de especialización y de un proceso tradicional de software, ya sea mediante modelos basados en plan o modelos ágiles, como se abordó en la unidad didáctica *Análisis y diseño orientado a objetos*, pero existe un modelo de proceso de software conocido como ingeniería de software orientada a la reutilización, en la cual se utilizan componentes, paquetes o módulos preexistentes, para ser adaptados según se necesite y sean incorporados al sistema (Sommerville, 2011, pág. 35).

En el caso del desarrollo web existen bibliotecas de código de diversos lenguajes como PHP, JavaScript, Visual Basic Script, Perl, entre otros, que potencializarán el desarrollo de sitios web, pero existen también aplicaciones completas como los CMS que nos pueden dar la funcionalidad a un sitio web completo.

Sommerville (2011, p. 13) afirma que “la reutilización de software se ha convertido en el enfoque dominante para construir sistemas basados en la web. Cuando se construyen tales sistemas, uno piensa en cómo ensamblarlos a partir de componentes y sistema de software preexistentes”.

Otra recomendación al implementar la ingeniería de reutilización de componentes es que “en general no es práctico especificar por adelantado todos los requerimientos para tales sistemas. Los sistemas basados en web deben desarrollarse y entregarse de manera progresiva” (Sommerville, p.13). Por lo que es necesario utilizar metodologías ágiles, una comunicación constante con el cliente y usuarios finales, pues formarán parte del equipo de desarrollo.

La ingeniería de reutilización de componentes aplicada al desarrollo de software y aplicaciones web permitirá no iniciar una aplicación desde cero, algo que se viene haciendo desde hace tiempo en el ambiente empresarial, donde es necesario la reutilización extensiva de componentes y programas. En el caso de los CMS, es posible elegir el que mejor se adecúe a las necesidades del cliente para configurarlo y extender sus prestaciones a través de **componentes**, **plug-in** y **módulos probados**, que cumplen con estándares y las recomendaciones de la **W3C**, un consorcio internacional que emite recomendaciones para las *World Wide Web*, lo que garantizará la **compatibilidad** del sistema con los navegadores y servidores web que se apeguen a estos estándares, como el nuevo **HTML5**, **CSS3**, **XML**, **RRS** y **WAI** (Baltazar, 2006).



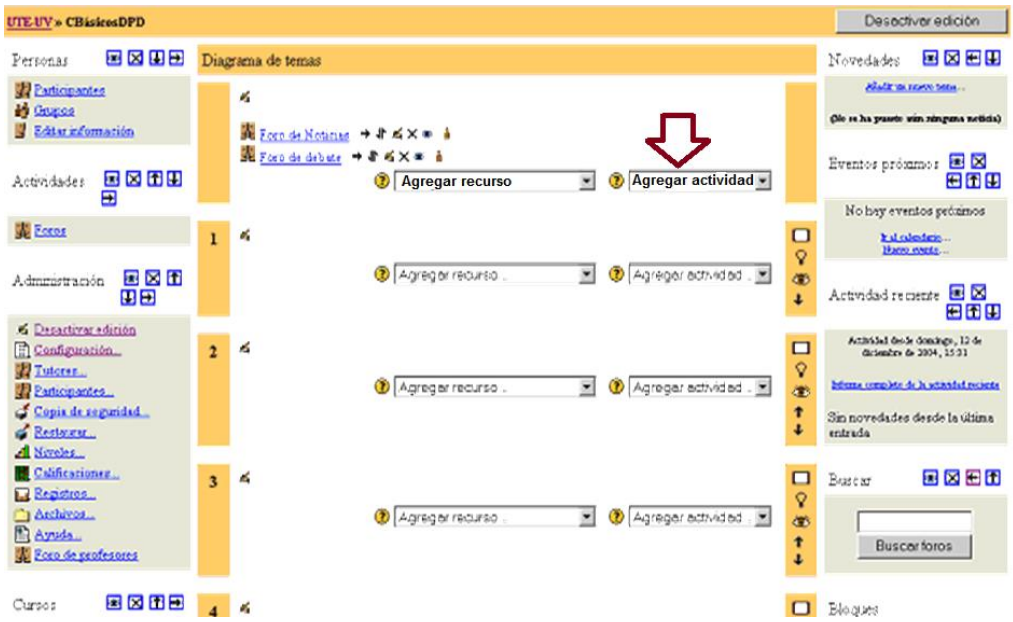
Desarrollo basado en componentes (Ariza y Molina, 2004)

Para observar la forma en que se utiliza el administrador de contenidos, es importante que localices materiales en internet. Se recomienda consultar la obra *Estudio de los sistemas de gestión de contenidos web* (Junta de comunidades de Castilla-La mancha, 2012), así como investigar recursos de video, por ejemplo, el nombrado *Tutorial administrador de contenidos (CMS) Mi página.net* (Mercadeo en internet, 1 de mayo, 2009).

Los ingenieros en desarrollo de software tienen un área de oportunidad en pleno desarrollo para crear y usar los gestores de contenido; sin embargo, deben estar preparados en conocimientos de múltiples herramientas, como **lenguajes de programación web**, **programación orientada a objetos**, **análisis y diseño de sistemas**, **algoritmos**, **sistemas operativos** y **bases de datos,** que se requieren para aplicarlas al desarrollo de páginas web.

Algunos ejemplos de usos de un LMS se observan en el ámbito educativo. Por ejemplo, la plataforma MOODLE, que es un administrador de contenidos utilizado ampliamente en el sector educativo; para integrar algunas actividades de aprendizaje, como rellenado de huecos, es necesario integrar código HTML en las instrucciones para la elaboración de estas actividades. A continuación se explican los pasos para integrar este recurso, como un ejemplo de edición o extensión a LMS, mediante códigos de programación. MOODLE es un sistema de distribución libre que puedes instalar en un servidor o en tu equipo. Para consultar la forma de instalar MOODLE, ingresa a su sitio oficial en internet.

Para editar la actividad de rellenar de huecos, se requiere ingresar a las herramientas de MOODLE, que generalmente se muestran en una pantalla como la siguiente:



Herramientas MOODLE (Universidad de Valencia, s/f)

El primer paso es ubicar las herramientas de actividades, como se observa en la imagen anterior.

Es necesario seleccionar la opción cuestionario y seleccionar el tipo llenado de huecos. Una vez seleccionado este tipo de actividad, se programa el llenado de huecos con el siguiente código HTML:

```
<p class="MsoNormal" style="margin: 0cm 0cm 0pt 19.85pt; text-align:
justify"><span lang="ES-MX" style="font-size: 12pt; color: black;
font-family: arial"><font color="#000099">Rellena los huecos y
presiona &amp;quot;Enviar todo y terminar&amp;quot; para verificar
respuestas</font>. </span></p>
<p></p>
<p>En el siglo {1:SHORTANSWER:#Respuesta errónea~=XV#Muy bien~%50%#}
se presenta una crisis económica que termina con los sistemas de
{1:SHORTANSWER:#Respuesta errónea~=producción feudal#Muy bien~%50%#}
y se da lugar a un nuevo sistema económico llamado capitalismo que
tiene como base excedentes en {1:SHORTANSWER:#Respuesta errónea~=la
producción#Muy bien~%50%#} y su principal propiedad es el
{1:SHORTANSWER:#Respuesta errónea~=capital#Muy bien~%50%#}. Con este
nuevo sistema de organización nace la sociedad
{1:SHORTANSWER:#Respuesta errónea~=burguesa#Muy bien~%50%#}.  </p>
<p align="justify"></p><br />
<p class="MsoNormal" style="margin: 0cm 0cm 0pt; line-height: 115%;
```

```
text-align: justify" align="justify"><span lang="ES-MX" style="font-
size: 12pt; color: black; line-height: 115%; font-family: arial"><font
face="Trebuchet MS,Verdana,Arial,Helvetica,sans-serif">Este nuevo
sector tomó en sus manos la economía y con ella una nueva forma de
producción: {1:SHORTANSWER:#Respuesta errónea~=un trabajador se
especializa en una sola parte del proceso#Muy bien~%50%#}.
</font></span></p>
<p align="justify"><font face="Trebuchet
MS,Verdana,Arial,Helvetica,sans-serif"></font></p>
<p class="MsoNormal" style="margin: 0cm 0cm 0pt; line-height: 115%;
text-align: justify" align="justify"><span lang="ES-MX" style="font-
size: 12pt; color: black; line-height: 115%; font-family: arial"><font
face="Trebuchet MS,Verdana,Arial,Helvetica,sans-
serif"></font></span></p>
<p align="justify"><font face="Trebuchet
MS,Verdana,Arial,Helvetica,sans-serif"></font></p>
<p class="MsoNormal" style="margin: 0cm 0cm 0pt; line-height: 115%;
text-align: justify" align="justify"><font face="Trebuchet
MS,Verdana,Arial,Helvetica,sans-serif"><span lang="ES-MX" style="font-
size: 12pt; color: black; line-height: 115%; font-family:
arial">Paralelamente hay </span><span style="font-size: 12pt; line-
height: 115%; font-family: arial; mso-ansi-language: es">cambios
significativos, el poder de la religión {1:SHORTANSWER:#Respuesta
errónea~=católica#Muy bien~%50%#} se cuestiona<span style="mso-
spacerun: yes">  </span>luego de mil 500 años de expansión y poderío,
enfrenta un debacle, propiciado por pensadores como
{1:SHORTANSWER:#Respuesta errónea~=Lutero#Muy bien~%50%#}. Este
fenómeno social que va de la mano con el proceso de transformación del
feudalismo al capitalismo, acelera el cambio producido por las nuevas
doctrinas y filosofías de la concepción de la<span style="mso-
spacerun: yes">  </span>religión.</span><span style="font-size: 12pt;
line-height: 115%; font-family: arial; mso-fareast-language: es-mx">
</span><span lang="ES-MX" style="line-height: 115%; font-family:
"></span></font></p>
<p align="justify"><font face="arial,helvetica,sans-serif"></font></p>
<p class="MsoNormal" style="margin: 0cm 0cm 0pt; line-height: 115%;
text-align: justify" align="justify"><font face="arial,helvetica,sans-
serif"></font></p>
<p align="justify"><font face="arial,helvetica,sans-serif"></font></p>
<p align="justify"><font face="arial,helvetica,sans-serif"></font></p>
<p align="justify"><font face="arial,helvetica,sans-serif"></font></p>
<p align="justify"><font face="arial,helvetica,sans-serif">
</font></p>
<p align="justify"><font face="arial,helvetica,sans-serif"></font></p>
<p align="justify"><font face="arial,helvetica,sans-serif"></font></p>
<p></p>
<p></p>
```

Ejemplo de código HTML utilizado como una extensión de MOODLE para actividades de llenado de huecos.

El código anterior se refiere a un tipo de pregunta flexible, en donde se permite incrustar, dentro de un texto, espacios que deben ser completados por los usuarios. Así, en cada **hueco** el usuario puede seleccionar entre varias opciones, o bien, escribir su propia respuesta corta o número, como se utiliza en las preguntas de opción múltiple. Se observa resaltada en el código la sentencia donde se indica que marque errónea una respuesta distinta a la indicada como correcta. Es importante especificar la pregunta como un texto con códigos insertados para indicar dónde van los huecos y qué respuestas se van a admitir en cada uno de ellos.

3.2. Funciones básicas de un gestor de contenidos

Conocer que existen tecnologías de componentes reutilizables es de gran utilidad, ya que son una opción para implementar rápidamente un sitio web completo que permita realizar las siguientes tareas a los usuarios (Baltazar, 2006):

1. Creación de contenidos. Un CMS implementa diferentes interfaces para que un usuario pueda editar publicaciones y agregar contenido multimedia como imágenes, videos y adjuntar cualquier otro tipo de documento sin necesidad de tener conocimientos avanzados de diseño web. Esto es posible debido a que se incorporan editores del tipo WYSIWYG, lo que ves es lo que obtienes (por sus siglas en inglés, *What You See Is What You Get*), con los cuales es posible conmutar entre el código HTML y la manera en que éste se visualizará en el navegador. Ejemplos de editores WYSIWYG pueden ser Quanta Plus, NVU.Amaya y Microsoft Frontpage Writer (de OpenOffice.org).

Los editores WYSIWYG se enfocan en proporcionar un editor de HTML incorporado en el CMS para publicar y editar contenido web sin tener conocimientos de HTML a través de botones que dan formato al texto, incorporan hipervínculos o cualquier otro elemento multimedia como imágenes o videos. El caso de usuarios avanzados, permite editar directamente el código HTML, con lo que es posible tener control total de lo que realiza el editor WYSIWYG.

Ejemplo de Editor WYSIWYG (Ajaxshake, s/f,a) http://www.ajaxshake.com/public/usersFiles/main/html-editor-00_user_1_a99e4.jpg.

El texto que se edita en el **Editor xHtml para WYSIWYG** es el texto que se **publicará**, integrando características comunes de edición a la web como las que tienen las aplicaciones de edición de escritorio como Microsoft Word y OpenOffice.

Ejemplo de editor WYSIWYG (Ajaxshake, s/f, b)

2. Administración de contenidos. Cuando un usuario **crea** una publicación, se **almacena** en una **base de datos** junto a otra información como el nombre del autor, la fecha de creación, la categoría del documento, permisos de acceso, entre otras. La gestión de contenidos suele repartirse en un equipo de trabajo donde habrá autores, revisores, editores y administradores del sitio. Cada **perfil** requiere **permisos** especiales y de **acceso**, por lo que es necesario que el CMS permita también la administración de usuario y permisos, así como un sistema de comunicación interna y de control del **flujo de trabajo** (**workflow).**

3. Publicación de contenidos. Una vez que el contenido de una publicación está editado, puede publicarse o programarse para publicarse en una fecha posterior o durante sólo un periodo determinado. Estas características permiten que el contenido de tu sitio web cambie constantemente, dependiendo de los permisos de acceso, del idioma del navegador, del lugar geográfico desde donde se visite el sitio, de las preferencias del usuario o de la fecha. Cada CMG proporciona las herramientas que permiten la administración de una manera gráfica, permitiendo crear, editar y borrar publicaciones.

De esta manera el trabajo de publicación de la información se simplifica y se garantiza que el CMS tendrá información fresca y se mantendrá un sitio web dinámico.

4. Presentación de los contenidos. Es muy sencillo cambiar el aspecto del sitio web o del formato de los documentos sin necesidad de editarlos uno a uno, ya que se utilizan plantillas con tecnología CSS. Una vez que se ha seleccionado una plantilla, puede aplicarse a todo el sitio, una sección o a un artículo en particular, con lo que el formato de título, subtítulos o el texto del párrafo puede cambiar con tan sólo cambiar de plantilla.

Para consultar los pasos para hacer un gestor de contenidos web, investiga en diversos recursos que se ofrecen en internet u otras fuentes a tu alcance, por ejemplo, el nombrado *Cómo hacer un gestor de contenidos web* (Pérez, 16 de septiembre, 2010).

3.3. Arquitectura general de un gestor de contenido

Los administradores de contenido proporcionan un marco de trabajo o *framework* que posibilita la actualización, mantenimiento y ampliación de contenidos en la web, con la colaboración de múltiples usuarios sin requerir para estas actividades conocimientos avanzados de diseño o programación, ya que estas tareas estarán supeditadas al desarrollador y al administrador del sitio (*webmaster*) (Ramos y Ramos, 2014, p. 39).

Los administradores de contenido regularmente tienen una arquitectura de software basada en **capas o *layaouts***. Cada capa se encarga de alguna característica o funcionalidad del sistema. En la capa más baja, también llamada **núcleo** o ***core***, suelen colocarse funcionalidades de seguridad, de acceso a base de datos, de administración de extensiones y plantillas (*templates*), también puede incluir la administración de publicaciones y de usuarios.

En una siguiente capa suelen colocarse las extensiones que potencializan la funcionalidad del núcleo y permiten realizar tareas más específicas, como el control de acceso a usuarios, el registro a la comunidad, administración de publicaciones, calendarios, transmisión de audio y video, administración de archivos y una gran infinidad de tareas.

Una característica muy importante del diseño en capas es que separa la funcionalidad del sistema (lógica de negocio) de la manera en que se visualiza la información (capa de presentación de datos), con lo que es posible cambiar totalmente la apariencia del sitio web a través de un formulario que permitirá seleccionar entre diferentes plantillas instaladas. Una plantilla (*template*) es un archivo que describe la maqueta del sitio y su formato a través de hojas de estilo CSS, por lo que pueden existir varias plantillas instaladas y activas para todo el sitio web, o diferentes plantillas para diferentes secciones del sitio.

Existen plantillas de licencia libre y gratuita, mientras que otras se distribuyen con un costo en sitios de diseño gráfico profesional. Si se desea dar una apariencia única al sitio web, puede desarrollarse una plantilla a la medida.

Según Ramos y Ramos (2014, p. 94), para que una plataforma CMS tenga éxito debe de reunir las siguientes características.

- Debe ser intuitiva, fácil de entender, utilizar y administrar, sin opciones complicadas.
- Flexible en la personalización, debe ser fácil cambiar el diseño.
- Extensible a través de *plugins* y módulos, que permitan añadir nuevas funcionalidades.
- No se necesitan conocimientos de programación.
- Debe estar preparada y probada para ofrecer un alto rendimiento y velocidad.
- Debe ofrecer elementos de seguridad que garanticen las entradas y permisos.
- Debe tener una buena documentación y apoyo comunitario, es decir, debe haber un soporte público para resolver cualquier tipo de problema.
- Debe hacer hincapié en estándares web y buenas prácticas que aseguran la compatibilidad con navegadores y nuevas tecnologías que surjan.

Tipos de gestores de contenidos

Existen diferentes taxonomías para diferenciar a los gestores de contenidos, ya sea por licencia, funcionalidad o ámbito de la aplicación. Inicialmente se pueden diferenciar entre Administradores de Contenidos Empresariales (ECM), Administradores de Contenido Web (WCM), Administradores de Documentos Multimedia (DMS) y Administradores de Contenidos para el Aprendizaje (LCMS).

Ramos y Ramos (2014, pág. 96) divide los CMS en tres categorías:

- Según el lenguaje de programación empleado, PHP, ASP, Java, ASP, NET, Ruby On Rails, Python.
- Según el tipo de licencia
    - *Open source* o código fuente abierto. Que permiten la modificación del código fuente original para adaptarlo a nuestras necesidades.
    - Código fuente propietario. Que no permiten que se modifique el código fuente aun cuando se proporcione con la aplicación. Cualquier modificación o adecuación de la aplicación debe ser pedida al desarrollador o esperar por una actualización.

- Según el tipo de uso o funcionalidad
    - Blogs. Para crear páginas personales como WordPress, CMS Made Simple, Symphony.
    - Foros. Pensados para compartir opiniones como phpBB, MyBB.
    - Wikis. Diseñados para el desarrollo de contenidos de manera colaborativa como DokuWiki, MediaWiki, TikiWiki.
    - E-Learning. Son plataformas para contenidos de enseñanza en línea como Moodle, eCollege, Claroline.
    - E-Commerce. Plataformas para la gestión de usuarios catálogos, compra y pagos como osComerce, PrestaShop y Open Cart.
    - Workflow. Brinda las herramientas para gestionar el trabajo de grupos especializados. Ejemplos de estos sistemas son Collabtive, eGroupeware y phpCollab
    - Publicaciones digitales. Permiten crear periódicos o revistas como cofax, Open Journal Systema y ePrints.
    - Difusión de contenido multimedia. Permiten difundir imágenes, videos, podcast y prácticamente cualquier contenido multimedia junto con texto e imágenes. Algunos ejemplos son Drupal. Mambo y Joomla. que también permiten la creación de portales web.

3.4. Instalación de gestores de contenido

Lo normal es que un CMS esté soportado por un servidor web o HTTP como Apache, una base de datos, por ejemplo, de MySQL, y un lenguaje de programación como PHP. Actualmente existen diferentes proyectos con licencia libres, como GNU y GPL, por lo que en este tema sólo se enfocará en las principales actividades a tener en cuenta para instalar y configurar un CMS, y quedará en ti buscar la información de alguno en particular que sea de su interés.

Lo **primero** será **conseguir** un **servicio de hospedaje** web (*hosting*) que permita la administración del servidor web, así como la creación de al menos una base de datos. El **segundo** paso es **crear la base de datos**, por lo general, a través de una interfaz que provee el mismo prestador del servicio de *hosting*, como lo es PhpMyAdmin. Será necesario tener a la mano los datos de:

- Nombre del usuario de la base de datos.
- Nombre de la base de datos.
- Dirección web del servidor de la base de datos (normalmente **localhost** para servidores gratuitos).
- Clave de la base de datos.

Una vez que se cuenta con la base de datos lista es posible subir los archivos del CMS al servidor web. Se debe verificar que la versión del CMS funcione sin problemas con la configuración y la versión del servidor web, del servidor de base de datos y de PHP que se tiene en el servicio de *hosting*; de otro modo será necesario buscar otra versión del mismo o de diferente CMS.

Para subir los archivos al servidor de hosting puede usarse una interfaz que provee el prestador del servicio, o a través de un cliente **FTP,** si este servicio se permite en la cuenta de hosting. Algunos servicios gratuitos no permiten la conexión a través del **puerto 21** usado por el cliente FTP, por lo que es necesario usar los mecanismos que el mismo proveedor brinde para subir los archivos necesarios, y descomprimirlos en la carpeta **HTTP** del **servidor,** si fuera necesario.

Lo siguiente será abrir un navegador web en la dirección por omisión de tu nuevo sitio web, con lo que se abrirá el proceso de instalación del CMS. A continuación se deben seguir los pasos que menciona la guía del proceso de instalación del CMS en cuestión y seleccionar los datos adecuados según corresponda, así como llenar la información de la conexión a la base de datos.

Configuración del gestor de contenidos

Según la estructura de un CMS, consta de los siguientes elementos (McGraw Hill, s/f):

- **Frontend**. Es la parte pública para los usuarios que acceden como invitados o usuarios registrados.
- **Backend**: Es el área de administración del CMS, donde se llevan a cabo todas las tareas de actualización, configuración y administración del sitio.
- **Módulos de configuración y personalización del sitio web:** Permiten distribuir el contenido en diferentes posiciones.
- **Administrador**: Controla los permisos de acceso a usuarios, así como la asignación de perfiles.
- **Módulos de gestión de contenido**: En este elemento se controla el sitio web, la creación, edición, publicación de contenido, entre otros.
- **Plantillas:** Por medio de ellas se personaliza el aspecto o el diseño del sitio web.
- **Extensiones:** Integración de funcionalidades del gestor de contenidos como sistemas de e-Comerce, comunidades en línea, reservación de citas, e-Learning, entre muchas otras.
- **Selección de idiomas:** Es la posibilidad de utilizar múltiples idiomas y localización.

Dependiendo del CMS que se instale será necesario identificar estas secciones para poder configurarlas adecuadamente.

Asimismo, será necesario buscar la documentación de las extensiones que instalemos para potencializar la funcionalidad del sitio web que se esté desarrollando y configurarlas adecuadamente.

Si fuera necesario agregar una funcionalidad o característica especial al sitio web, y que no ofrezca el mismo CMS o alguna extensión, será necesario crearla mediante algún lenguaje de programación como PHP, utilizando la API que el desarrollador del CMS proporciona, lo que implicaría tener que invertir tiempo en conocer cada API de desarrollo, ya sea para Joomla, Drupal, Mambo, Mediawiki, Moodle o el CMS que selecciones, lo cual puede volverse una tarea por demás difícil para usuarios sin experiencia en tecnologías como HTML, CSS, PHP y la administración de servidores web y de base de datos.

En el caso de desarrolladores de software y de aplicaciones web, el desarrollo de extensiones para CMS se vuelve una oportunidad de negocio, ya que estas aplicaciones suelen venderse para proporcionar una funcionalidad que el CMS no incorpora inicialmente y que debe ser utilizado, instalado y administrado usualmente por usuarios no profesionales para sitios web personales o de pequeñas empresas.

Ventajas y desventajas del uso de administradores de contenido en la programación web. La ingeniería orientada a la reutilización tiene la ventaja de reducir costos y riesgos en el proceso de software, ya que la cantidad de software a desarrollar también se reduce (Sommerville, 2011). Sin embargo, es posible que la funcionalidad del componente o extensión no cubra con todos los requerimientos del usuario, por lo que será necesario ajustar los requerimientos iniciales o modificar el código fuente del componente siempre y cuando se tenga acceso al mismo y se cuente con los permisos explícitos en la licencia para tal efecto. De otro modo puede pedírsele al desarrollador que implemente esa funcionalidad especial y pagar por las adecuaciones, buscar un nuevo componente o programarlo nosotros mismos.

En resumen, las ventajas son:

1. **Reutilización del software.** Lleva a alcanzar un mayor nivel de reutilización de software.
2. **Simplifica las pruebas.** Permite que las pruebas sean ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados.
3. **Simplifica el mantenimiento del sistema.** Cuando existe un débil acoplamiento entre componentes, el desarrollador es libre de actualizar y/o agregar componentes según sea necesario, sin afectar otras partes del sistema.
4. **Mayor calidad.** Dado que un componente puede ser construido y luego mejorado continuamente por un experto u organización, la calidad de una aplicación basada en componentes mejorará con el paso del tiempo.
5. **Ciclos de desarrollo más cortos.** La adición de una pieza dada de funcionalidad tomará días en lugar de meses o años.

6. **Mejor ROI.** Usando correctamente esta estrategia, el retorno sobre la inversión puede ser más favorable que desarrollando los componentes uno mismo.
7. **Funcionalidad mejorada.** Para usar un componente que contenga una pieza de funcionalidad, sólo se necesita entender su naturaleza, más no sus detalles internos. Así, una funcionalidad que sería impráctica de implementar en la empresa, se vuelve ahora completamente asequible **(**Ariza y Molina, 2004).

Una desventaja es que si no se cuenta con los permisos para modificar el código fuente es posible perder el control en cuanto a la evolución del sistema, ya que será necesario esperar a que el desarrollador del componente lance una nueva actualización.

Sin embargo, el desarrollo de extensiones para los CMS representa un nicho de oportunidad para desarrolladores web que ofrecen de manera gratuita una versión reducida de su software, y por un costo generalmente asequible, la versión completa.

Para revisar la forma en que se personaliza un componente ya existente. Investiga diversos recursos en internet u otras fuentes que tengas a tu alcance, fuentes confiables y propias del área, por ejemplo, el recurso nombrado *Crear componente para Joomla 2.5*

4. Alojamiento web

Para poder alojar el sitio web una vez que se ha realizado la **maquetación de la página**, se tienen que analizar los diferentes tipos de servicios de hospedaje web que ofrecen las diferentes compañías.

Según el Glosario de informática e internet (2013), el *hosting* “consiste en el almacenamiento de datos, aplicaciones o información dentro de servidores diseñados para llevar a cabo esta tarea.” Es otorgar espacios dentro de la red en los cuales las organizaciones pueden colocar su sitio web; el hospedaje también conocido como hosting, alojamiento, hospedaje web, alojamiento web, web site, hosting, web hosting o webhosting, es un negocio público o privado que consiste en brindar un servicio para mantener archivos de uno o más sitios web. Quienes brindan este servicio son empresas de alojamiento web, quienes almacenan los archivos web. “Permite la publicación de sus archivos en Internet y […] brinda otras funciones, como cuentas de correo electrónico y soporte de bases de datos. Por lo general, se establece una tarifa mensual por el alojamiento. […] Puede cambiarse fácilmente a un paquete de mayor tamaño si fuera necesario” (Harnett, 2014 p. 24).

Como se menciona anteriormente, el alojamiento puede ser gratuito en servidores que, a cambio del espacio, colocan publicidad en las páginas, o comercial, dependiendo del espacio que se requiera, la privacidad, límite de cuentas de correo asignadas a usuarios, etcétera. Más

adelante se explicarán las características que se requieren analizar para elegir una empresa para alojar el sitio web.

Pasos para alojar un sitio web (basado en Webconsultoría, 2013).

Pasos para alojar un sitio web:

1. Obtener un nombre de dominio. Para alojar un sitio web, en primer lugar, se requiere obtener un nombre de servidor de dominio DNS (por sus siglas en inglés, *Domain Name Server).* “El DNS es un servicio de búsqueda de datos de uso general, distribuido y multiplicado. Su utilidad principal es la búsqueda de direcciones IP de sistemas anfitriones

(hosts[1]) basándose en los nombres de éstos. El estilo de los nombres de host utilizado actualmente en Internet es llamado ´nombre de dominio´" (Fernández, 1999-2000). El nombre de dominio es el enlace entre el dueño del sitio web y el resto de usuarios de la red.

El uso de un dominio propio permite que tanto la dirección de las páginas web como el correo electrónico tengan un nombre significativo que haga alusión a la empresa o usuario que lo posee. Una vez adquirido un nombre de dominio, éste puede ser utilizado independientemente de la localización física del servidor web o proveedor. Utilizar un nombre de dominio relacionado con el nombre de la empresa ofrece una imagen de profesionalidad, por ejemplo: www.miSitio.com.mx, www.unadmexico.mx. Como puedes observar, el nombre de dominio se integra con el nombre de la institución.

Estos dominios pueden ser de dos tipos: los dominios genéricos y los códigos por países. "Dentro de los dominios genéricos se distinguen dos tipos dependiendo de que su aprobación haya sido o no patrocinada por un sector de actividad, una organización o un gobierno, por ejemplo, en cuyo caso, su uso quedará restringido a las entidades o personas que incluyan dentro del ámbito de patrocinio" (Arias 2006, p. 307-308). El siguiente es un listado de algunos dominios genéricos:

Nombre de dominio	Patrocinado	Patrocinador
.com	NO	Uso comercial
.net	NO	Uso comercial
.org	NO	Uso comercial
.mil	SI	Usos militares de U.S.A
.int	NO	Organizaciones derivadas de acuerdos entre gobiernos
.edu	SI	Instituciones educativas
.gov	SI	Gobierno de U.S.A
.biz	NO	Uso comercial
.info	NO	
.pro	NO	Profesionales
.name	NO	Personas físicas
.coop	SI	Cooperativas
.aero	SI	Industria Aeronáutica
.museum	SI	Museos
.jobs	SI	Recursos humanos

Tipos de dominios (Arias 2006, p. 307-308).

[1] Esta definición ha ido cambiando a lo largo del desarrollo de Internet, pues si bien antes el host era definido como una sola computadora conectada a Internet, ahora también entra en esta definición el "hosting" virtual, que permite que a una sola computadora hospedar varios espacios en la Red. Es el caso de Geocities, por ejemplo, cuyo servicio se basa en "brindar" estos espacios a cambio de la colocación de publicidad en los mismos

Dominios de países. Son aquellos que incorporan un sufijo que indica la nacionalidad por ejemplo: **.ar** para Argentina, .**cl** para Chile .**it** para Italia, .**mx** para México, etcétera. Todos los países tienen su propio dominio territorial, excepto Estados Unidos, que sólo utiliza dominios genéricos. Es importante aclarar que no es obligatorio el uso de dominios territoriales, ya que pueden contratarse dominios genéricos sin importar el país de procedencia. (Gaitan y Pruvost, 2001, p.44).

Para adquirir un nombre de dominio se requiere realizar los siguientes pasos:
a. Seleccionar una empresa registradora acreditada de dominios. El nombre de dominio es otorgado por organizaciones llamadas registradoras acreditadas. La selección de esta organización se realiza de acuerdo con los requisitos establecidos por las mismas. Puedes revisar las empresas registradoras certificadas por ICANN en México, o a nivel internacional, en el sitio oficial InterNIC, que es un servicio reconocido por el departamento de comercio de Estados Unidos, avalado por ICANN (INTERNIC, 2014) http://www.internic.net/origin.html.

La misma empresa en la cual se registra el nombre de dominio también brinda el servicio de hospedaje o alojamiento web. Es por ello que se debe analizar qué tipo de servicios ofrece. Estos elementos de análisis de una organización o empresa registradora se detallarán más adelante.

b. Realizar el registro en una organización registradora acreditada. En un segundo momento, para obtener el nombre de dominio, es necesario registrarse en alguna de las organizaciones que se encargan de gestionar estos nombres, las cuales consultaste en la página de ICANN.

Muchas de las organizaciones registradoras de nombre de dominio incluyen herramientas que permiten comprobar si un nombre de dominio ha sido registrado previamente, así como registrar dominios .com, .net, .org y .edu; además, dispone de la utilidad *Who is* para comprobar si un dominio ha sido dado de alta.

En el caso de México, para registrar nombres de dominio **.mx,** las organizaciones registradoras acreditadas por ICANN se muestran en la siguiente tabla.

Servicio: Servicio a usuarios finales División: NIC México Web: **www.nic.mx**
Servicio: Servicio a Registrados Acreditados Division: Registry .MX Web: **www.registry.mx**

Servicio: Asignación de Recursos de Internet División: NIC México Web: **www.nic.mx**

Divisiones de NIC México (LatinoamerICANN, 2008).

La ICANN o Corporación de internet para la asignación de nombres y números (por sus siglas en inglés, *Internet Corporation for Asiggned Names and Numbers)* es la organización **responsable a nivel internacional de asignar espacio de direcciones numéricas** de protocolo de internet (IP), identificadores de protocolo, de las funciones de gestión del sistema de nombres de dominio de primer nivel genéricos (gTLD), y de códigos de países (ccTLD), así como de la administración del sistema de servidores raíz (ICANN, 2013). Su sitio oficial es https://www.icann.org/es

Más adelante se expondrá un ejemplo de un registro de nombre de dominio en una organización registradora gratuita.

Ahora se expone el siguiente paso para realizar el alojamiento web.

2. Análisis de las características de las organizaciones registradoras de dominio y alojamiento web. Para seleccionar una empresa de alojamiento es necesario analizar los servicios que cada proveedor otorga por el costo que se va a pagar y, en el caso de que sea gratuito, observar las características y servicios que ofrece. Algunos de estos servicios y características son, según Diamond (2013, p. 21-22):

Memoria: Es la capacidad de almacenamiento que te ofrece el servicio. La memoria se mide en megabytes. Una página web sencilla necesitara de 5 a 10 megabytes. Sin embargo, encontrarás muchas compañías de *hosting* que ofrecen un número ilimitado de memoria.

Velocidad (*bandwidth*): Es la velocidad con la que el servidor puede enviar y recibir información en un determinado momento. Por ejemplo, cuando la nueva computadora británica que costaba sólo veintidós euros salió al mercado, el número de visitas fue tan alto que provocó que el sitio web de la compañía quedara fuera de servicio y se bloqueara.

Capacidad de la base de datos (*database capability*): Si tu página requiere de una base de datos para el inventario de productos, debes asegurarte de que este servicio no requiera un pago extra; y si lo tiene, debes conocer cuál es el costo adicional.

Transferencia de datos (*data transfer*): Hoy en día el término de transferencia de datos es más común gracias al servicio de teléfonos celulares como los iPhone y Androides, por los que se paga un costo adicional si se bajan fotografías, videos u otros documentos.

Sistema operativo (*operating system*): Así como existen computadoras que operan en el sistema Windows, como las llamadas PC, y otras Machintosh o MAC, de la misma manera existen dos tipos de sistemas operativos para los sitios web.

Windows: Para páginas en lenguajes más complejos como ASP (*Active Server Pages*)
Linux: Para páginas hechas en PHP (*Hypertext Preprocesor*). Para saber cuál sistema operativo es mejor para tu sitio web, puedes consultar con la empresa de *hosting* y te informarán con más precisión en qué consiste cada uno y qué beneficios te brinda

El siguiente es un listado de empresas que brindan el servicio de *hosting* gratuito. Para saber cuál es la empresa que brindará un servicio de acuerdo con las características del sitio que se quiera alojar, hay que ingresar a cada uno de estos sitios y analizar los servicios que ofrecen.

Copia de la información (*backup*): Es muy importante preguntarle a la compañía de *hosting* cada cuándo hacen una copia de tu sitio web. En el caso de que pase algo con tu página y pierdas uno o más archivos, la compañía usará la última copia que ha hecho para tu sitio web. Algunas compañías hacen copia una vez a la semana, y otras lo hacen todos los días.

Análisis web (*web analytics*): El análisis web te va a permitir obtener estadísticas acerca del comportamiento y preferencias de los usuarios de tu sitio: qué páginas son más visitadas, de dónde vienen las personas que visitan tu sitio, etcétera.

Consideraciones para elegir una empresa de alojamiento de web
El sitio web Soluciones media (2012) propone considerar algunas de las siguientes características.

- Buscar un hospedaje que se adapte a tus necesidades reales.
- El servidor deberá operar con una IP del área geográfica en la que realizas tu actividad empresarial.
- Valora qué cantidad de espacio de almacenaje (megabytes o gigabytes) vas a necesitar.
- Determina el ancho de banda mensual necesario para el buen funcionamiento del sitio web.
- Límite de asignación de cuentas de correo (POP3).
- Alojamiento con opción a bases de datos y scripts.

3. Integración de las páginas al servidor web o alojamiento web. Una vez que se ha analizado la empresa que se **adecúa** a las **necesidades** de la **organización**, hay que seguir los pasos que indica cada una de las empresas registradoras de dominio para poder registrar el ***hosting*** y, posteriormente, alojar el sitio web que se refiere al paso de integrar toda la maquetación del sitio en el servidor seleccionado.

A continuación, se detalla un **ejemplo** de **alojamiento** web mediante uno de los diferentes tipos de servidores que prestan sus servicios de hospedaje web sin costo en una organización pública de alojamiento. Asimismo, se explica paso a paso cómo registrar el *hosting* en el dominio de Geocities.

Paso 1. Ingresa al navegador web y teclea la siguiente dirección: www.geocities.ws, una vez que realices este paso, observarás la siguiente pantalla:

Pantalla principal de (GEOCITIES, 2014b)

Paso 2. Haz clic en **SIGN UP NOW** para crear tu cuenta de usuario.
Una vez validada tu cuenta, anota tu usuario en el siguiente recuadro:

http://www.geocities.ws/

Anota tu contraseña en el siguiente recuadro

Contraseña:

Paso 3. Ingresa como usuario registrado haciendo clic en **Members,** ubicado en la parte superior derecha de la pantalla, como se muestra en la imagen anterior.

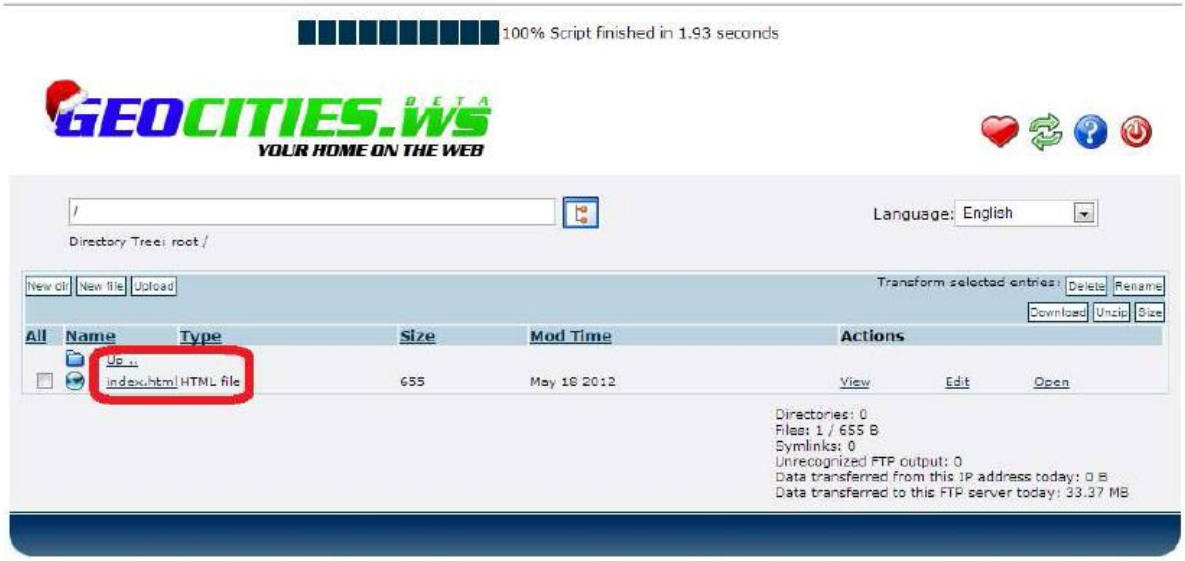
Paso 4. Una vez ingresado al panel de control, haz clic en **File Manager** para empezar a alojar tu sitio web.

Cpanel de GEOCITIES (2014a).

Paso 5. El directorio te mostrará los archivos existentes dentro de éste, y deberás ubicar fácilmente el archivo index.html, el cual es tu archivo principal, e iniciará cuando ingreses tu dominio. Un ejemplo sería como el siguiente:

http://www.geocities.ws/TU_USUARIO

Este archivo se genera automáticamente al momento de crear tu usuario, el cual deberás eliminar y subir la maquetación de tu sitio ya creada. También es posible construir tu sitio desde este servidor gratuito.



Pantalla de nuevo usuario.

Con este paso 5 se finaliza la descripción de registro en un ***hosting* gratuito,** tomando en consideración que los servicios gratuitos siempre van a adjuntar su publicidad en todos los sitios que se construyan en su servidor, es decir, la privacidad es vulnerable y no cuenta con servicios de seguridad en el manejo de datos en la web.

Para subir los archivos a un servidor web es recomendable leer el manual de uno de los programas de transferencia de archivos, que es multiplataforma y que te permitirá subir los archivos del sitio web de una forma sencilla, éste es **Filezilla.** Puedes localizar este manual en el sitio de la CECAIP (2014). Para subir tus archivos se recomienda utilizar **Filezilla** por contar con características de accesibilidad, soporte y uso en multiplataforma, o bien, puedes utilizar cualquier otro software que cuente con estas mínimas características.

Es momento de hacer las pruebas necesarias y tomar en consideración los temas analizados para poder elegir el *hosting* adecuado para el sitio web, así como el software **Filezilla** para poder subir los archivos de tu sitio web.

Cierre de la Unidad

Al finalizar la unidad se enfatiza la importancia de la nueva definición de las etiquetas en HTML5, que facilitan el poder maquetar un sitio de una manera más sencilla con la nueva estructura semántica del lenguaje, así como la integración de estilos CSS3 en los sitios, ya que éstos son los que brindan la oportunidad de definir reglas que hacen que el sitio trabaje y se adecúe a las necesidades requeridas.

Debido a HTML5 y CSS3 es que los sitios web, y la web en general, están tomando una nueva definición para creación de sitios para la web, haciendo que estos sitios puedan ser visibles en cualquier tipo de dispositivo móvil con acceso a internet.

En esta unidad se han estudiado los sistemas de gestión de contenido para crear sitios web de contenido dinámico a través del modelo de ingeniería de software, orientada a la reutilización. Se explica cuáles son las funciones y características básicas que tienen en general, como la administración de publicaciones, de usuario, extensiones y plantillas.

Se explicaron los diferentes criterios que se utilizan para clasificar a los CMS respecto al tipo de licencia, a su funcionalidad y al lenguaje de programación y tecnologías que utilizan, como el servidor web y de base de datos. Se abordó la arquitectura general de un CMS, que tiene un diseño por capas donde la capa más interna es el núcleo del CMS; le siguen la capa de extensiones, que se encarga de la lógica de negocio de la aplicación; y finalmente, la capa de presentación que se encarga del formato de las publicaciones e información en general.

Finalmente se presentaron los requerimientos generales para instalar un CMS y se expusieron las ventajas y desventajas que su uso conlleva. La página de la UnAD México está implementada sobre uno de los administradores de contenido más populares de código abierto, que es Joomla, mientras que la plataforma de e-Learning que utilizas en tu aprendizaje, la cual se conoce como Moodle, es otro ejemplo de un CMS aplicado a la educación en línea (LMS) de código abierto.

A lo largo de esta unidad has logrado conocer las características principales para considerar al momento de alojar un sitio en cualquiera de los servidores de tipo comercial o gratuito que brindan los servicios de *hosting*. Se expuso un ejemplo paso a paso del uso de un servidor gratuito para que pudieras observar la forma de registrar el dominio y, posteriormente, alojar la maquetación previamente realizada.

Estos temas han servido para que vayas conociendo cada uno de los pasos que debes seguir para poner a funcionar el sitio web que diseñarás. La siguiente unidad se enfocará en la seguridad con la que debe contar un sitio web, lo cual es importante para identificar y afrontar sus vulnerabilidades.

Para saber más

Ya que CSS3 cuenta con una gran cantidad de propiedades, para revisarlas es necesario que consultes los siguientes recursos:

- W3Schools (2014). *ASP.NET Tutorial*. Recuperado de: https://www.w3schools.com/css/DEFAULT.asp
- Sintes, B. (2013). *Lista de propiedades CSS*. Recuperado de: http://www.mclibre.org/consultar/htmlcss/css/css-propiedades.html
- Román, I. (2012). *6 aplicaciones online para maquetar tus diseños*. Recuperado de http://www.trazos-web.com/2012/08/13/6-aplicaciones-online-para-maquetar-tus-disenos/

Para saber más sobre el administrador de contenidos Joomla, su instalación, configuración, administración y personalización consulta los siguientes recursos:

- Mazier, D. (2013). *Mejore su sitio Joomla. Las mejores extensiones para las versiones 2.5 y 3.0*. Barcelona: Ediciones ENI. Recuperado de: https://tinyurl.com/y6tbzbfu

- Mazier, D. (2012). *Joomla 2.5. Cree y administre sus sitios web.* Barcelona: ENI Ediciones.

- Mazier, D. (2018). *Joomla! 3.8 Cree y administre sus sitios Web*. Ediciones Eni. Recuperado de: https://tinyurl.com/y3g2qggv

Para saber más sobre el administrador de contenidos Wordpress, su instalación, configuración, administración y personalización, consulta los siguientes recursos:

- Aubry, C. (2012). *Wordpress 3.5. Un CMS para crear y administrar blogs y sitios web*. Barcelona: ENI Ediciones.
- Aubry, C. (2015). *WordPress 4: un CMS para crear y administrar blogs y sitios web*. Ediciones ENI. Recuperado de: https://tinyurl.com/yxexlo8l

- Aubry, C. (2012). *Cree su primer sitio Web. Del diseño a la realización*. Barcelona: ENI Ediciones. Recuperado de: https://tinyurl.com/y2ylbrwq

Puedes revisar más información sobre cómo instalar diferentes CMS en el siguiente recurso:

- Ramos, A. y Ramos, M. de J. (2014). *Aplicaciones Web*. Madrid: Paraninfo.

Para saber más sobre el desarrollo de diseño responsivo, revisa el siguiente sitio:

- Mejía, J. C. (23 de enero, 2012). *Guía de Responsive Web Design: todo lo que necesita saber sobre Responsive Web Design.*

Para saber más sobre imágenes fluidas y flexibles, revisa el siguiente recurso:

- Garro, A. (22 de febrero, 2014). Capítulo 3 imágenes flexibles. En: *Responsive web design.* Recuperado de http://www.arkaitzgarro.com/responsive-web-design/capitulo-3.html

Es importante que revises el siguiente documento para saber más acerca del surgimiento e historia de las compañías registradoras de dominio, en relación con el *tema 4. Alojamiento web.*

- ICANN (s/f). *Nuevo programa gTLD.* Disponible en https://archive.icann.org/es/topics/new-gtlds/factsheet-new-gtld-program-sep09-es.pdf

En el sitio se expone un ejemplo de compañía mexicana registradora de dominios, pero, además, hay una gran cantidad de compañías registradoras en México. Esta compañía se recomienda porque está registrada ante la ICANN.

- MexHosting (s/f). *Hospedaje de Páginas Web. Web Hosting en México. Registro de Dominios*. Recuperado de http://www.mexhosting.com.mx/

El siguiente documento se recomienda para saber más respecto al diseño de páginas web.

- Álvarez, G. A. (1996). *Guía práctica para usuarios de HTML. Creación de páginas Web.* Madrid: Anaya Multimedia.

En el siguiente documento puedes consultar más términos sobre hospedaje o alojamiento web.

- Jacobsen, O. J. y Lynch, D. C. (Marzo, 1991). *A glossary of networking terms.* CA: Interop, Inc. Recuperado de http://tools.ietf.org/html/rfc1208

Fuentes de consulta

- Acámica (s/f). *Maquetando el monstruo web.* Disponible en http://www.acamica.com/cursos/11/maquetando-el-monstruo-web
- Arias, P. M. (2006). *Manual práctico de comercio electrónico*. Madrid: Las Rozas.
- Ariza, M., y Molina, J. C. (septiembre 2004). *Introducción y principios básicos del desarrollo de software basado en componente.*
- Aubry,C. (2012). *HTML5 y CSS3 Revolucione el diseño de sus sitios web*. Barcelona: Ediciones ENI. Disponible en https://tinyurl.com/yxg4vd6z

- Baltazar, L. (2006). *Sistemas gestores de contenido SGC*.
- CECAIP Comisión Estatal para el Acceso a la Información Pública (2014). *Mini manual de uso y configuración Filezilla. Manual sobre el uso y configuración del servidor Web filezilla versión 3.5.2.* Zacatecas: CECAIP, Servicio de Sistemas e Informática.
- Centro de Apoyo Tecnológico a Emprendedores, Fundación Parque Científico y Tecnológico de Albacete. (2012). *Estudio de los sistemas de gestión de contenidos web. Análisis de las mejores soluciones del mercado*. Albacete: Junta de Comunidades de Castilla-La mancha. Publicado bajo licencia Creative Commons. Disponible en http://www.bilib.es/uploads/media/estudio_sistemas_gestion_contenidos_web_cms.pdf
- Diamond, F. (2013). *Tu negocio online ¡Hecho Fácil!: La guía paso a paso para lograr ¡el sueño del negocio propio!* Nueva York: Penguin Group.
- Diez Media (9 de febrero, 2014*). Diseñando sitios web responsivos.*
- Fernández, C. R. (1999-2000). DNS. Madrid: ATI Asociación de técnicos en informática. *Glosario básico inglés-español*. Disponible en http://www.ati.es/novatica/glosario/glosario_internet.html#Domain%20Name%20System
- Frain, B. (2012). *Responsive web design with HTML5 and CSS3.* Birmingham: Packt Publishing Ltd.
- Gaitan, J. J. y Pruvost, A. G. (2001). *El comercio electrónico*. Santa Fe: Universidad Nacional del Litoral.
- GEOCITIES (2014a). *CPANEL.* Disponible en http://members.geocities.ws/index.php
- GEOCITIES (2014b). *Top Class Unlimited Free Web Hosting*. Disponible en http://www.geocities.ws/
- Glosario de informática e internet (junio, 2013). *Hosting*. Disponible en https://www.internetglosario.com/letra-h.html
- Harnett, M. (2014). *Guía de acceso rápido a Google Adwords.* Buenos Aires: Granica.
- ICANN Internet Corporation for Asiggned Names and Numbers (2013). *¿Qué es la ICANN?* Disponible en https://www.icann.org/es

- InterNic (2014). *The Accredited Registrar Directory.* Turkey, Los Angeles, Singapore: ICANN.
- LatinoamerICANN (2008). *mx: NIC México presenta su División Registry .MX.* Lima: NicMx-Alfa Redi
- Marcotte, E. (25 de Mayo, 2010). *Responsive web design. Alist Apart.*
- McGraw Hill (s/f). *Unidad 2. Sistemas gestores de contenidos*. Disponible en http://www.mheducation.es/bcv/guide/capitulo/8448183924.pdf
- Mercadeo en internet (1 de mayo de 2009). *Tutorial administrador de contenidos (CMS) Mi página.net. YouTube.* https://www.youtube.com/watch?v=2GZe-wuFS2U&feature=youtu.be
- MSDN Microsoft Developer Network (2014a). *Lección 2: Conceptos básicos de HTML, XHTML y CSS.* Disponible en https://docs.microsoft.com/es-es/previous-versions/msdn10/Hh749021(v=MSDN.10)
- MSDN Microsoft Developer Network (2014b). *Lección 4: Uso de las etiquetas HTML5.* Disponible en https://docs.microsoft.com/es-es/previous-versions/msdn10/hh749023(v=msdn.10)
- Peña, J. (2013). Unidades absolutas y relativas. *En: CSS 3. Unidades de medida de fuentes. Artículos técnicos comunidad MSDN.* Disponible en http://msdn.microsoft.com/es-es/library/dn602493.aspx
- Pérez, A. J. (16 de septiembre de 2010). *Cómo hacer un gestor de contenidos web.* [Archivo de video] YouTube http://www.youtube.com/watch?v=3Y4mGzPAmVI&feature=youtu.be
- Raiola Networks. (2019). *Manual de Filezilla como cliente FTP*. [En linea] Disponible en https://raiolanetworks.es/blog/manual-filezilla-cliente-ftp/
- Ramos, A., y Ramos, M. de J. (2014). *Aplicaciones Web*. Madrid: Paraninfo.
- Rigó, A. (8 de febrero, 2013). *Diseño web adaptable.* Disponible en *http://www.antoniorigo.com/diseno-web-adaptable/*
- Rodríguez, S. (2013). *Sistemas de pruebas y control de la maquetación. ARGP0110.* Málaga: IC Editorial.

- Soluciones media (17 de mayo, 2012*). Como elegir un buen Hosting para mi página web*. Disponible en http://www.solucionesmedia.com/como-elegir-un-buen-hosting-para-mi-pagina-web/

- Sancho, M. (2013). *Curso básico HTML5.* Disponible en https://www.cursosenhd.com/category/programacion/html-html5/

- Sommerville, I. (2011). *Ingeniería de software*. México: Pearson Education. Pág 13,35-36.

- Tajadura, J. (2013). *Responsive Web Design. Diseño web adaptativo ya! ¿Un tutorial para empezar?*

- Trazos net (2022). *6 aplicaciones online para maquetar tus diseños.* Disponible en https://trazos.net/desarrollo-web/

- W3C (1999). *Transitional Document Type Definition*. Disponible en https://www.w3.org/TR/html4/sgml/loosedtd.html

Bibliografía complementaria

- Mejía, J. C. (23 de enero, 2012). *Marketing Digital. Beneficios del Responsive Web Design*.

- Cocher, J. (1 de marzo de 2012). *Practical e commerce. SEO Benefits of Responsive Web Design.* Disponible en http://www.practicalecommerce.com/articles/3395-SEO-Benefits-of-Responsive-Web-Design.

- Quezada, S. (21 de noviembre de 2013). *¿Qué es el responsive web design?* The last Blog.

- Canal de aula virtualmatematic (4 de abril de 2012). *Taller Joomla. MPG*. [Archivo de video]. YouTube. https://www.youtube.com/watch?v=Cqnk_SZHJz4

- Usuna, R. (2010). Los sistemas de Gestión de Contenidos en Información y Documentación. En: *Revista General de Información y Documentación*. Vol. 20. Pág 67-100.

Fuentes de imagen

- Ajaxshake (s/f, a). *Agregar un editor HTML en tu sitio web*. Disponible en http://www.ajaxshake.com/public/usersFiles/main/html-editor-00_user_1_a99e4.jpg
- Ajaxshake (s/f, b). *Editor html WYSIWYG para sitios web*. Disponible en http://www.ajaxshake.com/public/usersFiles/main/html-editor-30_user_1_bd1da.jpg
- Desarrollo web.com (2012). *Etiquetas semánticas del HTML5.* Disponible en http://www.desarrolloweb.com/articulos/images/html5/semantica-html.png
- Universidad de valencia (s/f). *Editando un curso con MOODLE.* Disponible en https://www.uv.es/bellochc/moodle/15edicioncurso.htm
- Webconsultoría (2013). *¿Qué necesito para tener mi sitio en Internet?* Disponible en http://www.webconsultoria.com.mx/images/sitio.jpg